

CSRN 3301

(Eds.)

Vaclav Skala

University of West Bohemia, Czech Republic

Computer Science Research Notes

**31. International Conference in Central Europe on
Computer Graphics, Visualization and Computer Vision
WSCG 2023**

Plzen, Czech Republic

May 15 – 19, 2023

Proceedings

WSCG 2023

Proceedings

ISSN 2464-4617 (print)

ISSN 2464-4625 (online)

CSRN 3301

(Eds.)

Vaclav Skala

University of West Bohemia, Czech Republic

Computer Science Research Notes

**31. International Conference in Central Europe on
Computer Graphics, Visualization and Computer Vision
WSCG 2023**

Plzen, Czech Republic

May 15 – 19, 2023

Proceedings

WSCG 2023

Proceedings

This work is copyrighted.

However all the material can be freely used for educational and research purposes if publication properly cited. The publisher, the authors and the editors believe that the content is correct and accurate at the publication date. The editor, the authors and the editors cannot take any responsibility for errors and mistakes that may have been taken.

Computer Science Research Notes CSRN 3301

Editor-in-Chief: Vaclav Skala
c/o University of West Bohemia
Univerzitni 8
CZ 306 14 Plzen
Czech Republic
skala@kiv.zcu.cz <http://www.VaclavSkala.eu>

Managing Editor: Vaclav Skala

Publisher & Author Service Department & Distribution:
Vaclav Skala - UNION Agency
Na Mazinach 9
CZ 322 00 Plzen
Czech Republic
Reg.No. (ICO) 416 82 459

Published in cooperation with the University of West Bohemia
Univerzitni 8, 306 14 Pilsen, Czech Republic

CSRN 3301

International Program Committee

WSCG 2023

Baranoski,G. (Canada)
Benes,B. (United States)
Benger,W. (Austria)
Bouatouch,K. (France)
Dachsbacher,C. (Germany)
Drakopoulos,V. (Greece)
Eisemann,M. (Germany)
Galo,M. (Brazil)
Gavrilova,M. (Canada)
Gdawiec,K. (Poland)
Gudukbay,U. (Turkey)
Gunther,T. (Germany)
Hast,A. (Sweden)
Hauenstein,J. (United States)
Hitschfeld,N. (Chile)
Chaudhuri,P. (India)
Juan,M. (Spain)
Karim,S. (Malaysia)
Klosowski,J. (United States)
Kumar,S. (India)
Kurt,M. (Turkey)
Lee,J. (United States)
Liu,S. (China)
Lobachev,O. (Germany)
Manoharan,P. (India)

Marco,C. (Brazil)
Max,N. (United States)
Montrucchio,B. (Italy)
Pan,R. (China)
Pedrini,H. (Brazil)
Puig,A. (Spain)
Renaud,c. (France)
RESHETOV,A. (United States)
Ritter,M. (Austria)
Rodrigues,J. (Portugal)
Rojas-Sola,J. (Spain)
Sabharwal,C. (United States)
Savchenko,V. (Japan)
Segura,R. (Spain)
Semwal,S. (United States)
Scheuermann,G. (Germany)
Sirakov,N. (United States)
Sousa,A. (Portugal)
Thalmann,D. (Switzerland)
Tokuta,A. (United States)
Wu,S. (Brazil)
Wunsche,B. (New Zealand)
Zwettler,G. (Austria)

CSRN 3301

Board of Reviewers

WSCG 2023

Aguirre-Lopez,M. (Mexico)	Karim,S. (Malaysia)
Arora,R. (United States)	Klimaszewski,K. (Poland)
Baranoski,G. (Canada)	Klosowski,J. (United States)
Benes,B. (United States)	Komati,K. (Brazil)
Benger,W. (Austria)	Kuffner dos Anjos,R. (United Kingdom)
Bouatouch,K. (France)	Kumar,S. (India)
Cabiddu,D. (Italy)	Kurasova,O. (Lithuania)
Cline,D. (United States)	Kurt,M. (Turkey)
Czapla,Z. (Poland)	Lee,J. (United States)
Dachsbacher,C. (Germany)	Lefkovits,S. (Romania)
De Martino,J. (Brazil)	Liu,S. (China)
Drakopoulos,V. (Greece)	Lobachev,O. (Germany)
Dziembowski,A. (Poland)	Magdalena-Benedicto,R. (Spain)
Eisemann,M. (Germany)	Manoharan,P. (India)
ELLOUMI,N. (Tunisia)	Manzke,M. (Ireland)
Florez-Valencia,L. (Colombia)	Marco,C. (Brazil)
Galo,M. (Brazil)	Marques,R. (Spain)
Gavrilova,M. (Canada)	Max,N. (United States)
Gdawiec,K. (Poland)	Meyer,A. (France)
Gerrits,T. (Germany)	Miller,M. (Germany)
Goncalves,A. (Portugal)	Montrucchio,B. (Italy)
Grabska,E. (Poland)	Nawfal,S. (Iraq)
Grajek,T. (Poland)	Nguyen,S. (Vietnam)
Gudukbay,U. (Turkey)	Nikolov,I. (Denmark)
Gunther,T. (Germany)	Pagnutti,G. (Italy)
Hast,A. (Sweden)	Pan,R. (China)
Hauenstein,J. (United States)	Parakkat,A. (France)
Heil,R. (Sweden)	Pedrini,H. (Brazil)
Hitschfeld,N. (Chile)	Perez,S. (Spain)
Hu,C. (Taiwan)	Phan,A. (Viet Nam)
Hu,S. (China)	Puig,A. (Spain)
Chaudhuri,D. (India)	Quatrin Campagnolo,L. (Brazil)
Ivrissimtzis,I. (United Kingdom)	Ray,B. (India)
Juan,M. (Spain)	Renaud,C. (France)
Kaczmarek,A. (Poland)	Rershetov,A. (United States)

Ritter,M. (Austria)
Rodrigues,J. (Portugal)
Rodrigues,N. (Portugal)
Rojas-Sola,J. (Spain)
Romanengo,C. (Italy)
Sabharwal,C. (United States)
Satpute,V. (India)
Savchenko,V. (Japan)
Segura,R. (Spain)
Semwal,S. (United States)
Seracini,M. (Italy)
Shendryk,V. (Ukraine)
Scheuermann,G. (Germany)
Sirakov,N. (United States)
Skopin,I. (Russia)
Sluzek,A. (Poland)

Sousa,A. (Portugal)
Tandianus,B. (Singapore)
Tarhouni,N. (Tunisia)
Tas,F. (Turkey)
Thalmann,D. (Switzerland)
Tokuta,A. (United States)
Tourre,V. (France)
Wegen,O. (Germany)
Wu,S. (Brazil)
Wunsche,B. (New Zealand)
Yang,J. (China)
Yoshizawa,S. (Japan)
Zavala De Paz,J. (Mexico)
Zwettler,G. (Austria)

CSRN 3301

Computer Science Research Notes

WSCG 2023 Proceedings

Contents

Keynotes

Illustrating Geometric Algebra and Differential Geometry in 5D Color Space Benger,W.	1
Raytracing Renaissance: An elegant framework for modeling light at Multiple Scales Semwal,S.K.	3

FULL papers

Investigation on Encoder-Decoder Networks for Segmentation of Very Degraded X-Ray CT Tomograms Dulau,I., Beurton-Aimar,M., Hwu,Y., Recur,B.	11
Self-Checkout Product Class Verification using Center Loss approach Ciapas,B., Treigys,P.	21
Why Existing Multimodal Crowd Counting Datasets Can Lead to Unfulfilled Expectations in Real-World Applications Thissen,M., Hergenroether,E.	28
Coordinate-Unet 3D for segmentation of lung parenchyma Van Linh,L., Olivier,S.	36
Sex Classification of Face Images using Embedded Prototype Subspace Classifiers Hast,A.	43
Perceptions of Colour Pickers in Virtual Reality Art-Making Alex,M., Wünsche,B., Lottridge,D.	53
StarSRGAN: Improving Real-World Blind Super-Resolution Vo,K.D., Bui,L.T.	62
Modeling and Rendering with eXpressive B-Spline Curves Seah,H.S., Tandianus,B., Sui,Y., Wu,Z., Zhang,Z.	73
Training Image Synthesis for Shelf Item Detection reflecting Alignments of Items in Real Image Dataset Tomokazu,K., Ryosuke,S., Soma,S.	81
SAIL: Semantic Analysis of Information in Light Fields: Results from Synthetic and Real-World Data Kremer,R., Herfet,T.	90
Texture Spectral Similarity Criteria Comparison Havlicek,M., Haendl,M.	100

Designing a Lightweight Edge-Guided Convolutional Neural Network for Segmenting Mirrors and Reflective Surfaces Gonzales,M.E.M., Uy,L.C., Ilaio,J.P.	107
Monte Carlo Based Real-Time Shape Analysis in Volumes Gurijala,K.Ch., Wang,L., Kaufman,A.	117
Synthetic-Real Domain Adaptation for Probabilistic Pose Estimation Del-Tejo-Catala,O., Perez,J., Guardiola,J.L., Perez,A.J., Perez-Cortes,J.C.	127
Error-Robust Indoor Augmented Reality Navigation: Evaluation Criteria and a New Approach Scheibert,O., Möller,J., Grogorick,S., Eisemann,M.	137
Visualization of deviations between different geometries using a multi-level voxel-based representation Dietze,A., Grimm,P., Jung,Y.	147
Operational theater generation by a descriptive language Ghiotto,M., Desbenoit,B., Raffin,R.	158
Real-Time Reflection Reduction from Glasses in Videoconferences Tucholke,M-A., Christoph,M., Anders,L., Ochlich,R., Grogorick,S., Eisemann,M.	168
The Method of Mixed States for Interactive Editing of Big Point Clouds Benger,W., Voicu,A., Baran,R., Gonciulea,L., Barna,C., Steinbacher,F.	176
First Considerations in Computing and Using Hypersurface Curvature for Energy Efficiency Hauenstein,J., Newman,T.	186
MS-PS: A Multi-Scale Network for Photometric Stereo With a New Comprehensive Training Dataset Hardy,C., Queau,Y., Tschumperle,D.	194
Fast Incremental Image Reconstruction with CNN-enhanced Poisson Interpolation Erzar,B., Lesar,Z., Marolt,M.	204
Blocky Volume Package: a Web-friendly Volume Storage and Compression Solution Lesar,Z., Bohak,C., Marolt,M.	213
Generating Realistic River Patterns with Space Colonization Feng,H., Wuensche,B., Shaw,A.	222
AutogrASPing Pose of Virtual Hand Model Using the Signed Distance Field Real-time Sampling with Fine-tuning Puchalski,M., Wozna-Szczeniak,B.	232
Temporal Segmentation of Actions in Fencing Footwork Training Malawski,F., Krupa,M.	241
Accuracy of Legendre Moments for Image Representation Bustacara-Medina,C., Ruiz-Garcia,E.	249
Optimised Light Rendering through Old Glass Huan,Q., Rousselle,F., Renaud,C.	258
Versatile Input View Selection for Efficient Immersive Video Transmission Kłóska,D., Dziembowski,A., Samelak,J.	268
Massively Parallel CPU-based Virtual View Synthesis with Atomic Z-test Stankowski,J., Dziembowski,A.	277
Evolutionary-Edge Bundling with Concatenation Process of Control Points Saga,R., Beak,J.	284
Low-Rank Rational Approximation of Natural Trochoid Parameterizations Bálint,Cs., Valasek,G., Gergó,L.	292

SHORT papers

Detection of Printed Circuit Board Defects with Photometric Stereo and Convolutional Neural Networks Hable,A., Matore,M., Scherr,A., Krivec,T., Gruber,D.	300
A New Deterministic Gasket Fractal Based on Ball Sets Soto-Villalobos,R., Benavides-Bravo,F.G., Hueyotl-Zahuantitla,F., Aguirre-López,M.A.	306
The Usage of the BP-Layers Stereo Matching Algorithm with the EBCA Camera Set Kaczmarek,A.L.	315
Detail Preserving Non-rigid Shape Correspondences Bindal,M., Kamat,V.	323
Using the Adaptive HistoPyramid to Enhance Performance of Surface Extraction in 3D Medical Image Visualisation Padinjarathala,A.,Sadleir,R.	331
Semi-Supervised Learning Approach for Fine Grained Human Hand Action Recognition in Industrial Assembly Sturm,F., Sathiyababu,R., Hergenroether,E., Siegel,M.	340
Position Based Rigid Body Simulation: A comparison of physics simulators for games Seabra,M., Fernandes,F., Lopes,D., Pereira,J.	351
On Importance of Scene Structure for Hardware-Accelerated Ray Tracing Kacerik,M., Bittner,J.	361
Real-Time Visual Analytics for Remote Monitoring of Patient's Health Boumrah,M., Garbaya,S., Radgui,A.	368

Posters

On Unguided Automatic Colorization of Monochrome Images Sluzek,A.	379
Justice Expectations Related to the Use of CNNs to Identify CSAM. Technological Interview. Oronowicz-Jaskowiak, W., Wasilewski, P., Kowaluk, M.	385
The use of Artificial Intelligence for Automatic Waste Segregation in the Garbage Recycling Process Bobulski,J., Kubanek,M.	389
Detection of Dangerous Situations Near Pedestrian Crossings using In-Car Camera Kubanek,M., Karbowski,L., Bobulski,J.	393
Photogrammetry Workflow for Obtaining Low-polygon 3D Models Using Free Software Pardo,R., Remolar,I.	397
Polychromatism of all light waves: new approach to the analysis of the physical and perceptive color aspects Niewiadomska-Kaplar,J.	401

Illustrating Geometric Algebra and Differential Geometry in 5D Color Space

Werner Bengner

Airborne HydroMapping GmbH, A-6020 Innsbruck, Austria

w.bengner@ahm.co.at

Center for Computation & Technology at Louisiana State University, Baton Rouge, LA-70803

Vectors in three-dimensional Euclidean space are a fundamental concept in computer graphics and physics. Linear Algebra provides the well-known operations of adding vectors or multiplying vectors with scalars. Together with matrix algebra this framework allows for pretty much all operations that are needed for practical work. However, this set of operations is inherently incomplete such that not all operations known for scalar numbers can be applied to vectors. Particularly we can divide by numbers, but what does it mean to divide by a vector? Such an operation is not defined in Linear Algebra as there is no invertible product of vectors: There is the inner (dot) product $v \cdot u$ and the exterior (cross) product $v \wedge u$, but neither of them is invertible. It was the idea of William Kingdon Clifford to combine both products, defining the “geometric product” thereby as

$$uv := v \cdot u + v \wedge u,$$

which turns out to be invertible, though at the cost of introducing a higher dimensional space of so-called “multivectors”. This extension of Linear Algebra is known as Clifford Algebra or Geometric Algebra (GA) [Hes03, Hil13, DL03]. This formalism allows for a complete algebra on vectors same as for scalar or complex numbers. It is particularly suitable for rotations in arbitrary dimensions. In Euclidean 3D space quaternions are known to be numerically superior to rotation matrices and already widely used in computer graphics. However, their meaning beyond its numerical formalism often remains mysterious. GA allows for an intuitive interpretation in terms of planes of rotations. This algebraic framework extends easily to arbitrary dimensions and is not limited to 3D, like quaternions. However, our intuition of more than three spatial dimensions is deficient. The space of colors forms a vector space as well, though one of non-spatial nature, but spun by the primary colors red, green, blue. The GA formalism can be applied here as well, amalgamating surprisingly well with the notion of vectors and co-vectors known from differential geometry: tangential vectors on a manifold correspond to additive colors red/green/blue, whereas co-vectors from the co-tangential space correspond to subtractive primary colors magenta, yellow, cyan. GA in turn considers vectors, bi-vectors and anti-vectors as part of its generalized multi-vector zoo of algebraic objects. In

3D space vectors, anti-vectors, bi-vectors and co-vectors are all three-dimensional objects that can be identified with each other, so their distinction is concealed. In particular, in 3D all three basis vectors are given by the three primary colors red, green, blue. A bi-vector is the outer product of vectors. The bi-vector given by the \vec{x} and \vec{y} axis in Euclidean space is therefore the plane spun by the xy plane. Three such planes exist in three dimensions: xy , xz and yz (in cyclic notation). In the color space those combinations of two basis color vectors are then yellow = red \wedge green, cyan = green \wedge blue, and magenta = blue \wedge red. Same as in Euclidean space, where we can identify a vector with a plane via the notion of a “normal vector”, we can identify a color with a mixed color via its complementary color. This identification may ease some usage, but also leads to confusions, because the underlying objects - a vector versus a plane, or a pure color versus a mixed color - are inherently different.

Higher dimensional spaces exhibit the differences more clearly. In four dimensions there exist four vectors but six bi-vectors. Using space and time as the four dimension space, the four basis vectors $\vec{x}, \vec{y}, \vec{z}$ and \vec{t} result in the six possible combinations xy, yz, zx (three “spatial” bi-vectors) and xt, yt, zt (three “temporal” bivectors). Evidently, identifying every vector with every bi-vector is no longer possible in 4D as it was in 3D. The distinction between direction vectors and planes becomes unavoidable. Using colors instead of spatial dimensions we can expand our intuition by considering “transparency” as an independent, four-dimensional property of a color vector. We can thereby explore 4D GA alternatively to spacetime in special/general relativity. Here, we start with red, green, blue and transparent as the basis vectors and construct three non-transparent mixed colors yellow, cyan, magenta and three transparent pure colors transparent red, transparent green, transparent blue. Clearly, there is no way to identify those six bi-vectors with the six vectors in 4D space, not even via some complement.

However, even in 4D possibly confusing ambiguities remain between vectors, co-vectors, bi-vectors and bi-co-vectors: bi-vectors and bi-co-vectors - both six-dimensional objects - are visually equivalent. A co-vector in differential geometry is a linear, scalar valued function on vectors. These functions form their own vector space and can be seen as dual vectors. Visually

we may interpret co-vectors as the complement of a vector to form the full space: In 3D a plane complements a vector to form the full volume. Therefore a co-vector is equivalent to a bi-vector. Both have three components in 3D. In 4D a vector is complemented with a tri-vector to form a four-volume, thus in 4D co-vectors and tri-vectors are equivalent. Within the concept of color-spaces the co-vectors play the role of subtractive colors. Here, the basis co-vectors are built by the CMY system **yellow**, **cyan** and **magenta**. Their combination via light-subtractive filtering (expressed as the \wedge -product) forms the bi-co-vectors **green** = **yellow** \wedge **cyan**, **blue** = **cyan** \wedge **magenta** and **red** = **magenta** \wedge **yellow**. The equivalence of bi-co-vectors with vectors in 3D space is obvious: they are the same colors. The basis co-vectors of 4D color space are constructed by “cutting off” a basis vector from the full 4D “color-hypervolume” $\Omega := \text{red} \wedge \text{green} \wedge \text{blue} \wedge \text{transparent}$, **transparent magenta**, **transparent cyan** and **transparent yellow**. For instance, “cutting off” **red** from Ω yields **green** \wedge **blue** \wedge **transparent**, a 3D “color volume”, which is equivalent to a **transparent cyan** co-vector. Four such co-vectors exist in 4D: **transparent cyan**, **transparent magenta**, **transparent yellow** and non-transparent white. They are the set of all combinations with three properties. A bi-co-vector is constructed by cutting off two properties from the color hypervolume Ω , for instance cutting off the bi-vector **red** \wedge **transparent** yields the bi-co-vector (non-transparent) **cyan**. It is visually identical to the bi-vector (non-transparent) **cyan** because cyan is “**blue** and **green**”, but can equivalently be described in 4D color space as “not transparent and not **red**”. Both bi-vectors and bi-co-vectors provided two color properties and are thus visually indistinguishable in 4D. Higher dimensions are needed for such an unequivocal distinctions.

Envisioning five-dimensional geometry is even more challenging to the human mind than four-dimensional geometry, which we can at least associate with space-time. In color space we can add another property to the three primary colors and transparency. For instance, we can add “texture” or ~~strikethrough-text~~ to constitute a five-dimensional vector space. The five-dimensional hypervolume is then

$$\Omega_{5D} := \text{red} \wedge \text{green} \wedge \text{blue} \wedge \text{transparent} \wedge \text{strikethrough}$$

as constructed from the five base color/texture vectors. In 5D we have ten bi-vectors and ten bi-co-vectors. The bi-vectors are built from the \wedge -product of all basis vectors, there are ten possibilities to combine two properties in 5D: three mixed colors **cyan**, **magenta**, **yellow**, three transparent pure colors **transparent red**, **transparent green**, **transparent blue**, three textured pure colors **red**, **green**, **blue**, and one ~~textured-transparent~~ element. In contrast, the bi-co-vectors are built from all color elements that combine three properties in 5D, which are also 10 color space elements: one non-

transparent, non-textured element built from all three colors, i.e. “white”, three transparent, textured colors ~~textured-transparent red~~, ~~textured-transparent green~~, ~~textured-transparent blue~~, three transparent mixed colors **transparent magenta**, **transparent cyan**, **transparent yellow** and three textured mixed colors **cyan**, **magenta**, **yellow**. None of these bi-co-vectors is visually equivalent to any of the bi-vectors in 5D. The three-property color elements are distinct from the two-property color elements. Thus, in this five-dimensional color space we can see immediately that bi-co-vectors are distinct from bi-vectors, a distinction that is not obvious in 4D or 3D. While envisioning the same geometrically via five spatial dimensions is hard, but using color space it is easy to comprehend. This impression serves to demonstrate that vectors, bi-vectors, co-vectors and bi-co-vectors are actually different kinds of vectors, and they should be treated as objects with different properties before identifying them in special situations. An explicit distinction clarifies the meanings of algebraic objects in 3D Euclidean space such as “tangential vectors”, “axial vectors” or “normal vectors”, which are just 3D names of these vector quantities: a “tangential vector” is basis vector in 3D; an “axial vector” is a bi-vector in 3D; a “normal vector” is a co-vector in 3D. Confusing these different algebraic objects in 3D unavoidably leads to programming errors, such as applying a wrong coordinate transformation (normal vectors transform inversely to tangential vectors). A type-safe implementation that honors the mathematical differences therefor allows for better, clearer formulations of algorithms in 3D that are less prone to implementation errors.

The ideas presented here are meant to inspire using colors and beyond as alternative to spatial geometry. We did not make use of the inner product which may find its use in vision research to describe perceptual intensity, for instance. Also, we did not make use of the anti-symmetric property of the \wedge product such that $x \wedge y = -y \wedge x$ which introduces an orientation (this is why the highest dimensional \wedge -product was called “ Ω ” in this text) to multivectors: with **red** \wedge **green** being a “left-polarized” yellow versus **green** \wedge **red** yielding a “right-polarized” yellow may open an approach to include more properties of light into a mathematical framework. This is left for future work and / or an inspired audience.

REFERENCES

- [DL03] Chris Doran and Anthony Lasenby. *Geometric Algebra for Physicists*. Cambridge University Press, 2003.
- [Hes03] David Hestenes. Oersted medal lecture 2002: Reforming the mathematical language of physics. *American Journal of Physics*, 71(2):104–121, 2003.
- [Hil13] Dietmar Hildenbrand. *Foundations of Geometric Algebra Computing*. Springer, 2013.

Raytracing Renaissance: An elegant framework for modeling light at Multiple Scale

Sudhanshu Kumar Semwal

Department of Computer Science

University of Colorado

Colorado Springs, CO

USA 80906

ssemwal@uccs.edu

ABSTRACT

Ray tracing remains of interest to Computer Graphics community with its elegant framing of how light interacts with virtual 3D objects, being able to easily support multiple light sources during rendering and using sampling of estimates of intensity values at multiple surfaces in a recursive manner using light as ray. Ray tracing can also provide a simple framework of merging synthetic and real cameras. Recent trends to provide implementations at the chip-level means raytracing's constant quest of realism would propel its usage in real-time applications. AR/VR, Animations, 3DGames Industry, 3D-large scale simulations, and future social computing platforms are just a few examples of possible major impact. Raytracing is also appealing to HCI community because raytracing extends well along the 3D-space and time, seamlessly blending both synthetic and real cameras at multiple scales to support storytelling. This presentation will include a few milestones from my work such as the Slicing Extent technique and Directed Safe Zones. Our recent applications of applying Scan&Track with machine learning techniques creating novel synthetic views, which could also provide a future doorway to handle dynamic scenes with more compute power as needed, will also be presented. It is once again renaissance for ray tracing which for last 50+ years has remained the most elegant technique for modeling light phenomena in virtual worlds at whatever scale compute power could support.

Keywords

Ray tracing, Slicing Extent Technique, Directed Safe Zones, Active Space Indexing Method, AR.

1. INTRODUCTION

In Augmented Reality applications as the synthetic camera images are merged with the reality all around us, the merging is usually not smooth due to lighting conditions and mismatch of conditions as two disparate events are joined together with spatial mismatch. The main thinking of this paper is that raytracing with Active Space Index Method could propel a renaissance of using raytracing techniques towards an effective solution of merging of such disparate events by merging real and synthetic scenes into one physical space captured in front of a set of cameras. We call this 3D-space an active-space as it allows projection of a point onto a set of cameras to be seen. Also, an active-space indexing method is developed so that given the projection of the same 3D point in active-space in the set of cameras can be used to estimate the 3D point's coordinates. This paper presents basic ideas in this paper so that real and imaginary objects could be part of raytracing

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

techniques. Summary of our previous work [Dau90, Kva97, Sem92, Sem93, Sem98a, Sem98b, Sem01] is first presented. Using some of these ideas we make a case towards using raytracing as a unifying concept towards merging synthetically generated scenes with camera-based sequences in the hopes of creating a process towards resolving subtle light mismatch seen in recent work in AR applications [Li20, Har23] and movies where synthetic and natural objects are merges to create a rendered image.

2. Spatial subdivision algorithms for Raytracing

A ray starting at some point C and passing through a point on the image-plane (IP) intersects with object A and generates two new rays. R1 and T1. These two rays recursively traverse the scene. For example, ray R1 is shown to intersect with object B generating, in turn, R2 and T2. Both the intersection points on objects A and B are in line-of-sight of light source as shown in the Figure 1, which contributes light intensity as L1 and L2 at points A and B respectively. When the rays start from point C the process is called *backward* ray tracing as opposed to when the rays start at the light source and then are tracked in forward raytracing. Since the idea is to generate the

scene for the image-plane, backward raytracing is considered much efficient as only those rays are tracked which originate from C and pass through every pixel on the image plane and are needed to create the scene-render. For example, if we are generating a 60 by 40 image, there are 2400 initial rays which are tracked through the scene starting at point C through 2400 pixels on the image plane. The intensity of each such ray starting at point C and passing through some pixel-point IP is estimated using a tree which keeps track of the secondary rays as shown in Figure 2.

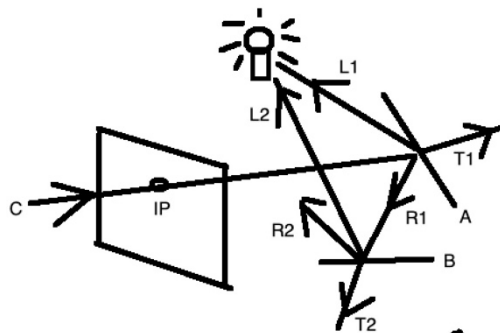


Figure1: Basic ray tracing starting from C.

To estimate the intensity at point IP on the image plane using the concept of recursive raytracing, we follow the path of the rays which are generated as reflective rays, R1 and R2, and transmitted rays, T1, and T2 (Figure 2) are followed generating their own intersection points with other objects in the scene in turn providing sample intensities which can then be combined as these intensities are summed upward through the tree from leaf nodes. Effect of lights for intensity values being returned from all visible intersection points can also be added as shown in Figure 3. Ambient intensities approximate the intensity returned by a ray when a ray travels out of the scene. Usually this happens when a ray travel outward away from all objects as it intersects the bounding box containing the scene [Woo90]. The direct line of sight from light sources to the intersection points means that the light source effects can also be incorporated into the intensities, as shown in Figure 3. Aggregate of all these effects can be summed incorporating the distance of light source, reflectivity and transmissivity of the objects mathematically. All these effects can be combine to return the estimated intensity for the pixel IP as shown in Figures 1 and 3. Subpixel samples can be incorporated when multiple stochastic rays generate effects based on bi-directional reflectance distribution function (BRDF) based on material properties of the objects in the scene. This leads to the idea of path tracing which have been used to create stunning realist images in many movies and animation sequences. Path tracing has also been

implemented in several industry leading special effects and movies recently as well.

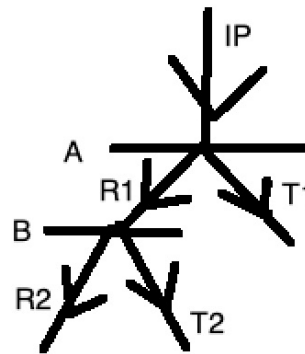


Figure 2: Estimating Intensity IP for a primary ray at point IP using secondary rays T1, R1, T2, R2 etc.

Path tracing has become the industry standard creating photo-realistic images by judiciously spawning several rays stochastically and applying BRDF functions and its variations judiciously. Our methods and new directions proposed can also be extended to those industry leading implementations. This includes path tracing renderers in Maya [Geo18], Sony's Arnold [Kul18], Weta's Manuka [Fas18], Disney's Hyperion [Bur18] and Pixar's Ruderman [Chr18]. Cloud implementations of raytracing are also working towards a goal of real time ray tracing [Xie2021] with 8 frames per second being reported, and NVidia is reporting GPU enhanced ray tracers for some years now. In recent work [Har21], the graphics pipeline is improved as deep learning techniques generate frames in between two graphics pipeline rendered frames. If the scenes are not changing, then generated rendered frames can be used as examples of images based on camera angle when scene is invariant. In game playing, as large number of frames are rendered for invariant scenes and light sources, they can be used for training and synthetic frame rendering and can sometime be used to generate acceptable frames as explained in [Har21]. The trained network is used to create an in between to offload the rendering pipeline and works for the case reported in [Har21]. The idea is to offload graphics pipeline and use deep learning, and supposedly faster frames in-between the rendered frames [Har21]. Ofcourse, when the scene or light source change then offloading can be suspended in favor of graphics rendering again. When the scene stabilizes then we can revert back the load to using deep learning learned images. As we will discuss later, most of the spatial data structures do not handle change in scenes due to explosions or when objects intersects the data structure updates are necessary (i.e. preprocessing) needs to occur adding delays in rendering. Once idea which we propose

later in this paper is to let the user know that the scene is under construction, especially when massively multiplayer games interactions are so critical to happen in real-time.

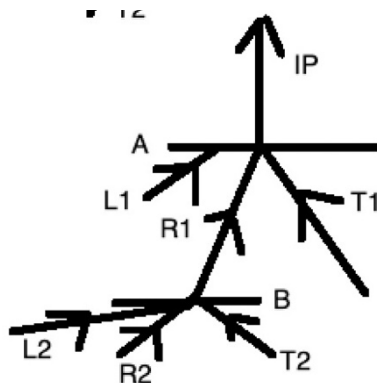


Figure 3: Merging of samples of intensities using a raytracing tree which us generated and their intensities combined.

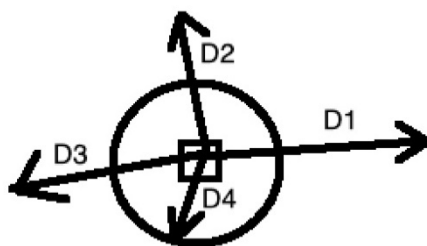


Figure 4: Proximity Cloud (PC) vs Directed Safe Zones (DSZ).

3. SET and DSZ ray tracing techniques.

The Slicing Extent Technique (SET) [Sem87, Sem92] uses projection of the objects in the scene of 2-D planes surrounding the object. In 1990, it was implemented using a 3D-grid interpretation, and was called the modified slicing extent technique (MSET) [Sem93]. The benefit was to allow the fast grid-traversal to be incorporated while ray traverses through the scene mimicking the SEADS implementation [Fuj86]. In addition, we also incorporated an octree [Gla84] to isolated isolate dense and sparse area in the scene so that a finer grid could be used for dense areas, and sparse areas can be skipped quickly in comparison to octree [Gla84], thus allowing multiple hierarchies to be managed using MSET. Later, work on the proximity cloud [Coh94a, 94b] further improved the grid-implementations for ray tracing. A method was developed during preprocessing so that a safe-distance value (D) was determined for every grid-voxel so that a ray passing through a voxel with

distance D can skip distance D without missing an intersection as during pre-processing it was determined that there we no objects withing a D distance from this voxel, hence the name safe-distance. This D distance allowed isolated areas to be bypassed in much more efficient way in cormarison to distance $D=1$ which will mimic the grid traversal itself. It is much faster to skip areas of the scene with no objects in it. During preprocessing, the value of D was determined using transformation [Bor86] to isolate areas of no-object efficiently by using a 3 by 3 by 3 filter on 3D-grid voxels. This is illustrated in Figure 4 where a 2D-grid voxel, or cell, is shown and nearest object for any ray through the edge of the 2D-voxel (cell) are $D1$, $D2$, $D3$, and $D4$ away. So minimum radius is $D4$ which is the safe distance a ray could travel in any direction from this 2D-cell without finding an object to intersect. Every cell thus could have its own 2D-circle, or extending this idea to 3D, its own 3D-sphere. Each voxel of the 3D-grid could have such sphere. One can now imagine every voxel to have different (yet close) values creating many spheres with no objects in them. One could imagine spheres of different radii, hence the term, proximity clouds (PC) used in [Coh94b] to describe a Proximity Cloud. Proximity Cloud was a major improvement very as it was more efficient way where ray tracing image generation times were shown to improve over the previously known grid implementations. Normal grid-traversal, moving from one voxel to next voxel could be suspended in favor of jumping D distance away without missing any intersections. Figure 4 shows this concept in 2D with four safe values in four directions. In PC, we choose the minimum of these 4 values, and conclude the safe distance of $D4$ for that voxel.

Usually city-block distance transformations are used to implement Proximity Clouds so instead of radius we could imagine city-block distances [Coh94b]. This faster method was further improved by a variation of the slicing extent technique called Directed Safe Zones (DSZ) in [Kva97] where the six safe-distances in six direction are calculated as the ray emerges out of a voxel through one of the six-faces of a 3D-voxel (grid-cell). All such distances are calculated during preprocessing by modifying PC's distance transformations filter to suit the DSZ implementation [Kva97]. After preprocessing in DSZ method each of the six faces of 3D-voxel of would have a distance associated with it moving outward from the cell towards left, right, bottom, top, up and down directions. In 2D, this is shown as $D1$, $D2$, $D3$, $D4$ values in Figure 4. In DSZ, the ray has the capacity to skip six different distances based on its direction of traversal as the ray passes through any of these six faces. This meant that Directed Safe Zones, which extends the Slicing Extent techniques, is more efficient. As shown in Figure 4, DZS has the

flexibility to choose either of D1, D2, D3 or D4 as safe-distances guaranteeing that image generation time will always be better or same in DSZ in comparison to PC implementations. DSZ uses larger distance based on the traversal direction of the ray showing in theory and is an improvement over PC. Using scenes from [Hai98] called random, lanes, snowflake, and cars, a comparative analysis in [Kva97] showed that DSZ outperformed the SEADS/grid and PC implementations for all four scenes. As expected, because of the potential of more empty areas in random and snowflakes scenes major improvements in rendering times were obtained for random and snowflake scenes using DSZ in comparison to Proximity Clouds and SEADS/Grid implementations. Typical performance speedup of 2 for DSZ were seen when compared with grid implementation. For the PC, speedup was 1.5 with respect to grid implementation. In all cases, as expected, DSZ outperformed the PC method and grid (SEADS) [Fuj86] method.

Additional benefits of DSZ method is that, in addition to the outgoing rays emanating from a voxel, DSZ method can treat incoming rays because each face of the voxel can maintain two distances, as was also explained in [Sem87, Sem93]. A ray passing through the left face of a 3D-voxel to the right, or from right to left, upwards-to-downwards or downwards-to-upwards, and front-to-back and back-to-front can be recognized, allowing two directions per face so that 12 different classifications instead of six are possible as a ray passed through a face of a voxel. This is a useful benefit allowing us to manage 3D scenes better, as we plan to embed synthetic scenes in active spaces as explained in next sections.

4. Active Space Indexing Method Review Setup and Data Capture

Active-Space Indexing Method [Sem01] uses ideas of triangulation, including closest distance between two rays to find 3 D (x,y,z) position of a point P. Given image-imprint Im1, Im2, and Im3 on the camera-images for a point P, Active Space Indexing method preprocesses projections of several 3D grid points on each camera images to determine the 2D-indices for each camera images. If we assume that 2D grid points can be indexed between 1 to n and 1 to m then Im1, Im2, and Im3 must fall on some index (x,y) using the projections of grid-patterns in p such planes. When three such indices on for each Im1, Im2 and Im3 points identified in all three camera images as corresponding to the point P, then these indices define an area, and that area will decrease first and then increase as we process p of these plane from front to back. This allows us to find a voxel which contains point P. Active-space indexing method connects the projection-space to the real 3D

space. More details of how Im1, Im2, Im3, called imprint-set can be used to determine position P is further explained in [Sem01] in more detail.

In summary, Active Space Indexing Method is a study of 2D-projections of a set of 3D-points as seen by three cameras. These set of 3D-points are arranged in real 3D-grid in physical space in front of the three cameras. This space in front of the camera is called an active space [Sem98]. The active space indexing method is created by projecting set of n by m planer points inside a rectangle R which contains n by m in an equal distance grid pattern. We used a whiteboard for this purpose. The points on the whiteboard can be shifted some distance away from the previous placement of the whiteboard. In this way, the whiteboard it has an effect of moving same points inside the rectangle R will also move parallel to previous plane positions. We repeat this process several times to obtain images for a set of p whiteboard positions. As the whiteboard moves, it has an effect that 3D grid points inside the active space are projected and preprocesses during preprocessing to create an active space indexing method. During preprocess, the exact pixel locations of all grid points for all p whiteboard positions are determined and stored during preprocessing. This information is sufficient to estimate point P's x,y,z location in n by m by p space using point P's imprint-set as explained in [Sem01]. Assuming the process repeats p times, using same distance. This will have an effect of creating a set of 3D-grid points in the real physical space which we call active-space. For example, n=m=4 and p=8 in Figure 5 so that a 4 by 4 by 8 grid of points are in the active space. Each time image-plane moves we record the projections of set of 4 by 4 image-plane points on 3 cameras which are called Left (L), Center (C) and Right (R) points as shown in Figure 5 and then the as shown in Figure 6. The three cameras are related in a way that all p planes are visible from all the three camera and their projections are therefore highly correlated. We also have shown the projected shape of the rectangles in Figure 5. As explained earlier, rectangle R was created on a whiteboard which was available in the lab [Sem98], and already had a 2D-grid-pattern and the intersection of this points were highlighted with a blue/black pen which were then easily picked up during pre-processing. In this way, we were able to correlated n by m by p points in active space to their projections by preprocessing p of these projections for every camera. All of these points were clearly visible in the three cameras because we had planned the arrangement. Figure 6 shows approximately the front-plane, marked F, and back-plane, marked B, and corresponding projections of rectangular B and F planes and their projected shapes in all three cameras. As the white-board moves from Back to Front p planes create p projections of the grid-pattern on the

on three cameras. Sets of these projections are created and processed during preprocessing and are the basis of active space indexing method. Vertical and horizontal lines and their projected lines on image planes are used to first find the index in 2D planes in all three cameras, as explained earlier, and a triangulation algorithm is used to find the voxel which contained the given 3D-point P using the image print (Im1, Im2, and Im3) of the point P. The image-print, i.e. Im1, Im2, Im3, is identified manually in our implementation. Image-imprint could be detected automatically as well. Finding image prints is called correspondence problem and there are a variety of techniques including finding significant points first and then correlating these points as image prints using the projections of these points on three cameras. We used significant points extraction and correlation of these points as mentioned in [Sem98a, Sem98b, Sem01].

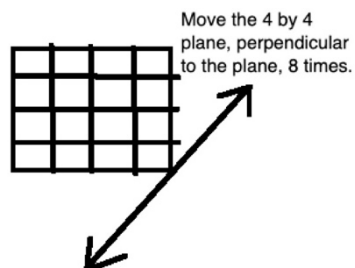


Figure 5: Scan&Track data collection – 4 by 4 planer grid moves 8 times perpendicularly to create a 4 by 4 by 8 active space, 128 grid points in 3D active-space. A set of voxels in 3D-real space is created.

Image imprint and corresponding 3D points in practice

Basic idea of Scan&Track [Sem98] system was implemented project by a 10 by 10 by 10 set of 3D grid points onto three highly correlated cameras [Sem98a, Sem98b, Sem01]. Highly correlated camera would mostly maintain the geometric relationship between any two grid points in the same plane. Data collection step uses a planer whiteboard with, say fixed 10 by 10 points clearly marked and visible from each of the camera. Next whiteboard is moved by fixed distance perpendicular to the present location of the whiteboard 10 times to capture projection of 100 points spaced as 10 by 10 by 10 grid of 1000 points all visible in the three cameras. Here the idea was to create active space, a 10 by 10 by 10 grid in physical area) with over constrained systems of 1000 points in 3D space. For example, a 3D point P and their associated projections image-imprint (Im1, Im2, and Im3) are known by processing each of the projections. Image imprint Im1, Im2, and Im3 are the pixel locations in the images. Now if the person is

inside the active-space and we identify same point, e.g. the tip of the nose in all three camera images as Im1, Im2, and Im3, then active space can be used to find P the location of the tip of the nose of the participant. Scan&Track system can now be used to identify set of image imprints. As explained, Active-space indexing method uses a triangulation process to find the 3D point given I1, I2, and I3 pixel location.

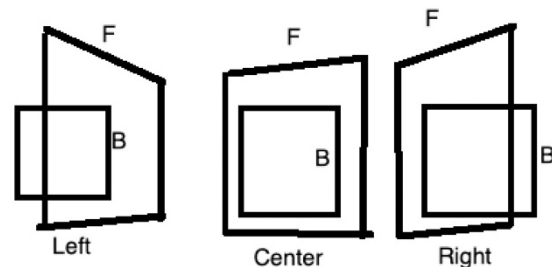


Figure 6: Scan and Track projections on left, center and right. Distortions of the planes are exaggerated to show the effect of projections of front and back planes of active space.

Generating Image imprints using Deep Learning

One of the tasks in the Scan&Track implementation was to generate image imprints (Im1, Im2, Im3). Identification of the imprint was done by a simple filter in our [Sem01] implementation. Today we can use deep learning algorithms to find distinct image imprints specifically for the human participant. The end points for hands and feet could be identified across the three cameras, creating image imprints I1, I2, I3 for hands and feet. This will identify 3D positions in the active-spaces for hands and feet and many such points, as explained in [Sem98b].

5. Future Proposed Work

Active-Spaces for both real and virtual worlds

Both Scan&Track and DSZ methods are based on 3D grid data structures. In DSZ, a voxel's six faces helped us define 12 different directional distance measures which we can safely skip along the direction of the ray. In the Scan&Track system active spaces are the 3D-grids in real-space which we can effectively use to embed both real and synthetically generate worlds.

Our main idea is that virtual worlds consisting of C, IP, A, B and L which can be used to generate intensity values at point IP. Real world consisting of human participant seen by three cameras. Active-space allows us to place A, B, and L relative to human participant while also isolating human participants in the real-work inside an active space. Now virtual objects (such as A, B, and L) can be

placed in the active space in front of the human participant by using active-space Indexing method to find the approximate location of the human participant and then placing A, B and L relative to H. Environment E could also be wrapped around all of these, as shown in Figure 7. E could be another projected image of some other active space, or another active space could be defined in that place behind the human participant as E. The idea is that world of many active spaces can be populated by synthetic scenes and synthetic objects as active spaces provides us one way to define virtual and realobject in the same 3D space. Once these objects are placed, raytracing can be used to combined overall scene. In Figure 7, we have tried to show the mixed-reality scene in active-space merging both synthetic and real objects together so that idea of raytracing can be applied to render images with real participant enclosed inside their active space, and E, represented by active-space of its own in turn. Extending the idea of multiple active spaces we can now expand the 3D space to larger areas as well as define the 3D-space recursively, e.g. one active space can contain several other active spaces. The faces of active space containing human participant H could be samples by multiple cameras in outside-in manner so that approximate intensities on the surface on the active-space can be used during ray tracing as was done in ASET [Dau90].

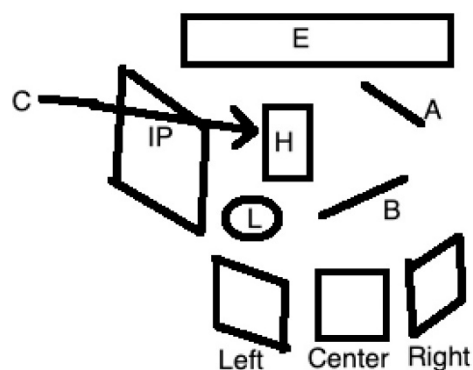


Figure 7: Mixed Reality Setup. Light source L has been placed in front of human participant H. Ray tracing camera is shown with C and IP (See also Figures 1 and 2). A and B are synthetic objects. Note E is environment which could be another Active space Index Mixed Reality Setup or can be green/blue screeded real camera-captured natural scene.

Proposal to merge moving Active-Spaces and multiple scales of grids

Both Scan&Track and DSZ methods are based on 3D grid data structures. In DSZ, a voxel's six faces helped us define 12 different directional distance

measures which we can safely skip along the direction of the ray. In the Scan&Track system active spaces are the 3D-grids or real world where we can embed other active-spaces or even synthetically generate virtual world objects. As the objects move the grid-spaces can be managed such that light-effects can be collected in the voxels as an approximation method as we have explored this idea in [Dau90] where primary rays, rays starting from point C in Figure 1 and secondary rays at first level called T1 and R1 or even second level T2 and R2 are checked for intersection with real objects to create a some level or correctness. However, at some level, say level 3 onwards, secondary rays may start to consider that the object occupies the whole voxel which it passes through, thus discarding any intersection checks for level 3 or more secondary rays. Our motivation in [Dau90] was to avoid the actual intersection where such approximation could suffice. Here we are proposing that external facing active spaces faces can be approximated by camera images which capture the human participants actions, and the image's r,g,b values can suffice as intensity value to combine with other effects during ray tracing process. This needs to be further investigation in future.

Scene changes and updates

One of the challenges for ray tracing has been that any movement of objects or light source, even if minor, can severely impact the final image. Spatial data structures need to be completely updated if such events occur. Also, any changes in the shape or the objects such as explosions can severely affect the whole data structure during ray tracing forcing us to first manage those updates to objects before rendering can occur. We think that such updates in the grid method can be managed by hierarchical embedding of one active-space in another or extending the active-spaces of multiple voxel size. As the objects move, some voxels may be vacated by the object and other will be occupied based on the movement of the object itself. By mapping both synthetic object and real-objects in active-spaces and their grids allows us one way to manage as both the real and imaginary worlds can be combined using number of active-spaces. Active spaces with high activity, handling major explosions, can be isolated and can be label as "under construction" where the walls of that active space would be considered "approximate" while under construction thus giving time to synchronize itself to "available again" while that active space completes whatever it was "under construction" for.

6. Summary

Main idea here is that both virtual worlds and real worlds are in appropriate scale merged using a variety of active-space grids holding both synthetic and real object at an appropriate scale.

Labelling a particular active space under construction is similar to real-life, for example when we see “under construction” sign—we know that experience will usually improve in future. More processor resources could be allocated to fix the restructuring of the active space to manage fragments of data in case of explosion, or movement due to scaling, translation and rotation which object may go through. Rendered frames which are labeled “under-construction” may also appear to notify the user that their experience will improve later.

7. CONCLUSION AND FUTURE RESEARCH

Using distance transformation, we developed a faster method for ray tracing called the directed safe zones (DSZ) where the direction of incoming and outgoing ray from a 3D voxel can be classified based on which face of the voxel the ray emerges out of or goes into. We also used rays to develop a triangulation method to first find the voxel which contains a 3D point P when point P's image imprint (Im1, Im2, Im3) is known. This led to calculating the location of point P. In this paper, we proposed that active spaces could be distributed in real-world of human participants and these worlds can be places with synthetic objects by adding them to the active spaces so that synthetic objects and active-spaces can raytraced, hopefully avoiding the mismatch of scale when synthetic and real-worlds are combined in Mixed Reality 3D applications. In our case, we are proposing that such scenes can be raytraced.

8. ACKNOWLEDGMENTS

My deepest thanks to my colleagues Dr. Jun Ohya, Department Head, and Dr. Ryohei Nakatsu, Director of ATR Media Integration and Communication Lab. hosting my Summer Research visits at ATR, Kyoto, Japan during 1997-99 where Scan&Track systems was developed. My deepest thanks to Dr. Vaclav Skala for providing me this opportunity to present this keynote at one of the finest graphics conference in the world – WSCG 2023! Thank you.

9. REFERENCES

[Bor86] Borgefors G. Distance transformation in digital images, *Computer Vision, Graphics and Image Processing*, 34, pp. 344-371 (1986).

[Bur18] Brent Burley, David Adler, Matt Jen-Yuan Chiang, Hank Driskill, Ralf Habel, Patrick Kelly, Peter Kutz, Yining Karl Li, and Daniel Teece. 2018. The Design and Evolution of Disney's Hyperion Renderer. *ACM Trans. Graph.* 37, 3,

Article 33 (July 2018), 22 pages.
<https://doi.org/10.1145/3182159>

- [Chr18] Per Christensen, Julian Fong, Jonathan Shade, Wayne Wooten, Brenden Schubert, Andrew Kensler, Stephen Friedman, Charlie Kilpatrick, Cliff Ramshaw, Marc Bannister, Brenton Rayner, Jonathan Brouillat, and Max Liani. 2018. RenderMan: An Advanced Path-Tracing Architecture for Movie Rendering. *ACM Trans. Graph.* 37, 3, Article 30 (Aug. 2018), 21 pages. <https://doi.org/10.1145/3182162>.
- [Cle88] Cleary J.G. and Wyvill G. Analysis of an algorithm for fast ray tracing using uniform space subdivision, *The Visual Computer* 4, pp. 65-83 (1988).
- [Coh94a] Cohen D. Voxel Traversal along a 3D Line, *Graphics Gems, IV*, pp. 366-368 (1994)..
- [Coh94b] Cohen D. and Sheffer Z. Proximity clouds - an acceleration technique for 3D grid traversal, *The Visual Computer*, 11, pp. 27-38 (1994).
- [Dau90] David Dauenhauer and Sudhanshu Kumar Semwal, Approximate Raytracing, *Graphics Interface*, Halifax, Nova Scotia, Canada, pp. 75-82. Canadian Information Processing Society, International Association for Computing Machinery's Special Interest Group on Computer Graphics and Interactive Techniques (ACM SIGGRAPH) (ACM SIGGRAPH), and Canadian Man-Computer Communication Society (1990).
- [Fas18] Luca Fascione, Johannes Hanika, Mark Leone, Marc Droske, Jorge Schwarzhaupt, Tomáš Davidovič, Andrea Weidlich, and Johannes Meng. 2018. Manuka: A Batch- Shading Architecture for Spectral Path Tracing in Movie Production. *ACM Trans. Graph.* 37, 3, Article 31 (Aug. 2018), 18 pages.
<https://doi.org/10.1145/3182161>.
- [Fuj86] Fujimoto A, Tanaka T. and Iwata K. ARTS: Accelerated ray tracing system, *IEEE Computer Graphics And Applications*, 6(4), pp. 16 26 (1986).
- [Geo18] Iliyan Georgiev, Thiago Ize, Mike Farnsworth, Ramón Montoya-Vozmediano, Alan King, Brecht Van Lommel, Angel Jimenez, Oscar Anson, Shinji Ogaki, Eric Johnston, Adrien Herubel, Declan Russell, Frédéric Servant, and Marcos Fajardo. 2018. Arnold: A Brute-Force Production Path Tracer. *ACM Trans. Graph.* 37, 3, Article 32 (Aug. 2018), 12 pages.
<https://doi.org/10.1145/3182160>
- [Gla84] Glassner A.S. Space subdivision for fast ray tracing, *IEEE Computer Graphics And Applications*, 4(10), pp. 15-22 (1984).

- [Gla89] Glassner, A.S. An Introduction to Ray Tracing edited by A.S. Glassner, Academic Press (1989).
- [Gla21] Andrew Glassner, Deep Learning: A visual approach, Penguin Random House, 776 pages, 2021.
- [Hai87] Haines E.A., A Proposal for Standard Graphics Environments, *IEEE CG&A*, 7(11), pp. 3-5 (Nov 1987).
- [Har21] Mark W Harris and Sudhanshu Kumar Semwal, A multi-stage advanced deep learning Graphics Pipeline, SA'21 Technical Communications: SigGraph Asia 2021 technical communications, Article No.: 7, pp. 1-4. <https://doi.org/10.1145/3478512.3488609> (2021).
- [Har23] Hartholt, Arno; Mozgai, Sharon Creating Virtual Worlds with the Virtual Human Toolkit and the Rapid Integration & Development Environment, In: 2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG), pp. 1–6, 2023.
- [Kul18] Christopher Kulla, Alejandro Conty, Clifford Stein, and Larry Gritz. 2018. Sony Pictures Imageworks Arnold. *ACM Trans. Graph.* 37, 3, Article 29 (Aug. 2018), 18 pages. <https://doi.org/10.1145/3180495>
- [Kva97] Kvanstrom H. The Dual extent and Directed Safe Zones techniques for ray tracing, *Graphics Interface*, 1-72 (1997).
- [Li20] Li, Jiaman; Kuang, Zhengfei; Zhao, Yajie; He, Mingming; Bladin, Karl; Li, Hao, Dynamic Facial Asset and Rig Generation from a Single Scan, In: *ACM Transactions on Graphics*, vol. 39, no. 6, 2020.
- [Sem87] Sudhanshu Kumar Semwal, The Slicing Extent Technique for Ray Tracing, Ph.D. dissertation supervised by Dr. Mike Moshell, Department of Computer Science, University of Central Florida, Orlando, Summer 1987, pp. 1-227 (1987). <https://stars.library.ucf.edu/rtd/5062/>
- [Sem92] Sudhanshu Kumar Semwal, Ray Tracing using the Slicing Extent Technique, Institute of Electronics, Information and Communication Engineering (IEICE) Spring Conference, Tokyo, Japan, pp. 7-367 (1992).
- [Sem93] Semwal S.K., Kearney C.K., and Moshell J.M. The Slicing Extent Technique for Ray Tracing: Isolating Sparse and Dense Areas, *IFIP Transactions*, vol. B-9, pp. 115-122 (1993).
- [Sem98a] Semwal SK, Ohya J. The scan&track virtual environment, *Virtual Worlds* 98, LNAI 1434, pp.63-80, 1998.
- [Sem98b] Sudhanshu Kumar Semwal and Jun Ohya, Geometric-Imprints: A Significant Points Extraction Method for the Scan&Track Virtual Environment, *Proceedings of the IEEE Third International Conference on Automatic Face and Gesture Recognition (F&G98) Conference*, April 14-16, 1998, Nara, Japan, pp. 480-485, IEEE Computer Society.
- [Sem01] Sudhanshu Kumar Semwal and Jun Ohya. Spatial Filtering using the Active-Space Indexing Method, in the *Graphical Models and Image Processing*, Academic Press journal, vol 63, pp 135-150 (2001).
- [Wes17] West Geoffrey, *Scale: The universal Laws of life and death in organisms, cities and companies*, Weidenfeld & Nicolson, Great Briton, pp. 1-455 (2017)
- [Whi80] Whitted T., An improved illumination model for shaded display, *CACM*, 23(6), 343-349 (June 1980).
- [Woo90] Woo A. Fast Ray-Box Intersection, *Graphics Gems, I*, pp. 395-396 (1990).
- [Xie21] Fen Xie, P. Mishchuk, W. Hunt, Real-time cluster path tracing, SA'21 Technical Communications: SigGraph Asia 2021 technical communications, Article No.: 17, pp. 1-4.

Investigation on Encoder-Decoder Networks for Segmentation of Very Degraded X-Ray CT Tomograms

Idris Dulau
LaBRI UMR 5800
Bordeaux University
Talence
FRANCE
idris.dulau@labri.fr

Marie
Beurton-Aimar
LaBRI UMR 5800
Bordeaux University
Talence
FRANCE
beurton@labri.fr

Yeykuang Hwu
Institute of Physics
Academia Sinica
Taipei
TAIWAN
phhwu@sinica.edu.tw

Benoit Recur
Institute of Physics
Academia Sinica
Taipei
TAIWAN
benoit.recur@gmail.com

ABSTRACT

Field of View (FOV) Nano-CT X-Ray synchrotron imaging is used for acquiring brain neuronal features from Golgi-stained bio-samples. It theoretically requires a large number of acquired data for compensating CT reconstruction noise and artefacts (both reinforced by the sparsity of brain features). However reducing the number of radiographs is essential in routine applications but it results to degraded tomograms. In such a case, traditional segmentation techniques are no longer able to distinguish neuronal structures from surrounding noise. Thus, we investigate several deep-learning networks to segment brain features from very degraded tomograms. We focus on encoder-decoder networks and define new ones addressing specifically our application. We demonstrate that some networks wildly outperform traditional segmentation and discuss the superiority of the proposed networks.

Keywords

X-Ray nano-tomography, segmentation, deep-learning, brain imaging.

1 INTRODUCTION

Field-Of-View (FOV) Nano-CT X-Ray synchrotron imaging provides 3D images of biological samples at about 300nm by computed tomography (CT). In this study, bio-samples are mouse brains stained with a Golgi solution targeting neuronal connectome (neuron cells, axons, dendrites, ...). Since the whole organ is much larger than the scanner FOV, each brain is cut in several blocks (sized $\approx 3 \times 3 \times 5 \text{ mm}^3$). Each block is introduced in a rod transparent to X-Rays and is positioned on a 3-Axis translational + Z-Axis rotational sample-holder. Numerous FOV CT acquisitions are measured in a sequence (cf. Fig. 1(a-b)) for imaging the whole sample rod. Each FOV acquisition is a set of N_θ radiographs (2560×2160 pixels). According to rod dimensions, scanning resolution and the overlap required between adjacent FOV 3D tiles, the overall rod scan is composed of ≈ 360 CT acquisitions.

Each FOV CT acquisition is processed by a CT algorithm to reconstruct a 3D volume (i.e. tomogram sized

$2560^2 \times 2160$ voxels) imaging the inner features of the sample [Tof96, NW02]. For instance, Fig. 1 (c) shows acquired radiographs (strong absorption reveals Golgi-stained neuronal features), and Fig. 1 (d) is a 3D visualisation of a reconstructed tomogram. Once all tomograms have been reconstructed, the next data processing step consists of segmenting the brain features from the background (empty regions, non-stained parts of the brain) in order to perform further 3D analysis and visualisation of the mouse neuronal connectome. In that context, it is obvious that tomographic reconstruction and segmentation steps are of great significance since both high-quality and accuracy are required in tomograms to segment for achieving such a whole brain analysis.

However overall data acquisition of a whole brain needs to be performed as a routine application [SLH⁺23], thus requiring to drastically reduce both acquisition and CT reconstruction time. This limitation is practically addressed by reducing the number of acquired radiographs in CT acquisition, resulting to a degraded 3D tomogram. Such tomogram can no longer be segmented using traditional methods. Thus in this paper we investigate deep-learning based on encoder-decoder networks since it has been already demonstrated that they are well adapted to segmentation and / or denoising problems. We first introduce our overall data processing sequence and the positioning of our investigation in section 2 and we preface encoder-decoder based seg-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

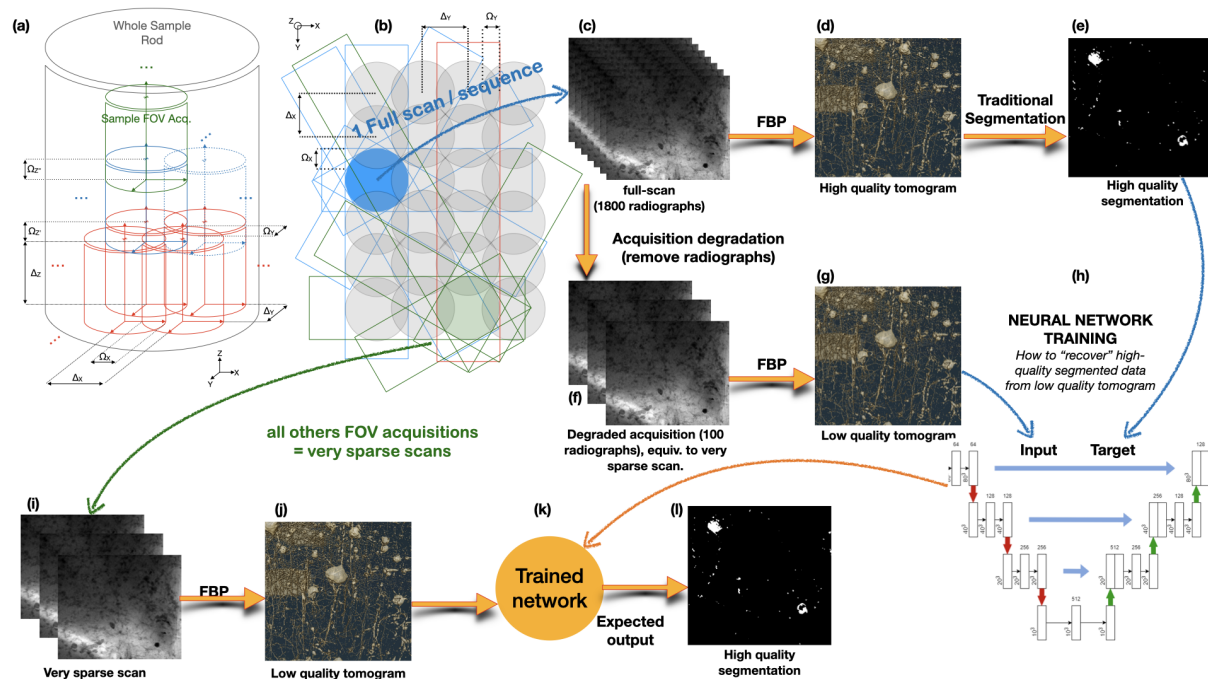


Figure 1: (a) Sample rod with several FOV areas. Translation steps (Δ) calibrated to optimize overlaps (Ω). (b) Acquisitions by X/Y-Axis translations: illustration of two FOV scans in blue and green. A unique full-scan (c) per sequence provides high-quality tomogram (d) and binarisation (e) by traditional segmentation. Degraded full-scan (f) provides degraded tomogram (g). The pair {(g), (e)} = (input, target) training dataset (h). All other sequence scans done in very sparse configuration (i), providing 3D volumes (j), that are processed by the trained network (k). Expected output (l) is a high-quality segmentation from degraded tomograms.

mentation methods in section 3. In section 4 we design two networks more suitable to our use-case and explain our design choices according to our data. Before concluding, all networks are experimented in section 5 and we discuss why our networks design specificities lead to better results compared to the state-of-the-art techniques.

2 EXPERIMENTAL DATA PROCESSING

We present in this section the overall data processing sequence developed in this study and the positioning of investigated deep-learning methods (cf. Fig. 1).

Traditional tomographic reconstruction is the Filtered Back Projection (FBP), widely used in both medical and industrial CT scanners and appreciated for its easy implementation and fast computation [Han81]. However, FBP is very sensitive to the noise in the acquisitions and the tomogram quality depends on the number of radiographs (cf. image degradation with the number of radiographs on Fig. 2). Indeed, according to Shannon-Nyquist theory, FBP requires at least $N_{opt} \approx 2000$ radiographs to optimally reconstruct slices sized $N^2 = 2560^2$ pixels [PGF⁺05]. In our case, $N_\theta = 2000$ leads to an excessive irradiation damaging bio-sample, but we can safely measure 1800 radiographs (denoted

full-scan in the following) without noticeable accuracy losses on the tomograms. However, due to the huge number of acquired data (≈ 360 FOV acquisition / rod, and ≈ 40 rods / mouse brain), both acquisition and data processing times have to be drastically decreased. The most practical solution consists of reducing the number of radiographs (fast-scan).

Such a fast-scan - i.e when $N_\theta \ll N_{opt}$ - leads to angular sampling problem which can be addressed thanks to iterative CT reconstructions (IR-CT). This domain of research has been amazingly fruitful and addressed for decades [HNY⁺13, HL89, GG96, And89, JW03, KS01]. Numerous IR-CT methods have been proposed [DMND⁺00, ZWZ⁺18, RFK⁺14], including radiograph ordering optimizations [Kol05, KB98, EF99], multi-scale / multi-grid methods [MKL⁺16] or GPU-based implementations [SKKH07, RXT07, ZHZ09].

Furthermore, FOV tomography (also denoted interior / region-of-interest / local tomography) can be addressed by dedicated IR-CT methods [ZNG08, HGD⁺10, LHW⁺15, PM17, SKR⁺14]. Despite their efficiency, IR-CT techniques are sparsely deployed in routine applications because of both their complexity and their computation time ($\times 10$ to $\times 100$ FBP requirements). This latter limitation makes them ultimately unrealistic to be used in our project.

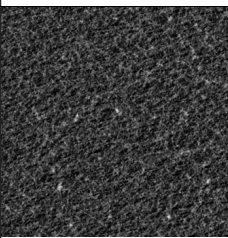
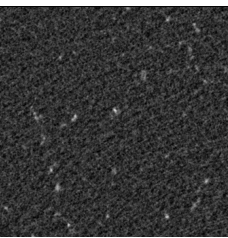
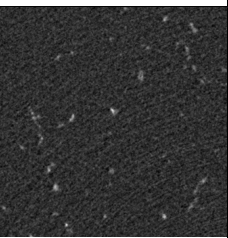
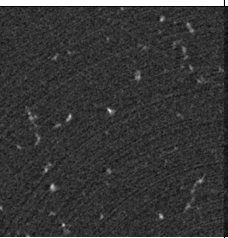
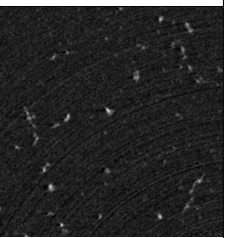
N_θ	100	300	600	900	1800
					
Dice	0.02	0.84	0.93	0.96	1.00 (ref.)

Figure 2: Reconstructed slices from acquisitions with 900 to 100 radiographs, compared to reference (ref.) full-scan image ($N_\theta = 1800$). Noise and artefacts increase when scanning becomes faster. Dice is obtained by adaptive Renyi entropy based segmentation (compared to segmentation result from full-scan).

Golgi-staining leads to high contrasted features (foreground) on full-scan reconstructed slices (cf. bright features on Fig. 1(d)). Thanks to such a sample preparation, empty regions of the brain are very low intensity and almost confounded with empty space (i.e. both can be considered as background). In that condition, a traditional (binary) segmentation such that an adaptive Renyi entropy method [San04, KR14, SRR08, FZL⁺17] conveniently extracts neuronal features even if they are widely unbalanced (less than 3% of a tomogram are neuronal connectome in our case). Unfortunately, traditional segmentation accuracy directly depends on the tomogram quality, and thus significantly decreases when a fast-scan is processed. As illustrated on Fig. 2, we have for instance experimented that about 30% (*resp.* 50%) of the segmented features are lost when the segmentation is processed from a tomogram reconstructed with 300 (*resp.* 100) radiographs (compared to the segmentation result obtained from full-scan tomogram).

Alternatively to traditional approaches, we investigate deep-learning networks for accurately segmenting degraded tomograms. Our bio-sample acquisition produces massive CT data from a sequence in which all FOV CT scans are exactly measured in the same conditions and on the same sample. This high repeatability is particularly suitable for deep learning. We thus apply the following data processing protocol (cf. Fig. 1): i) only one full-scan FOV CT (c) is acquired per sequence to obtain a high-quality tomogram (d) and its corresponding high-quality segmentation (e) ; ii) the full-scan is degraded (remove radiographs) to reach very sparse scanning configuration (f) ; iii) the resulting degraded tomogram (g) is combined to the segmentation (e) to provide the pair (input, target) feeding the investigated neural network training ; then, iv) all the other FOV regions are acquired in fast-scan conditions (j), providing degraded tomograms (j) which are segmented using the trained network. One may note that the training dataset is automatically obtained thanks to the full-scan combined with a traditional data process-

ing pipeline, leading to an overall processing sequence which is fully automated.

3 RELATED WORKS

Several efficient Convolutional Neural Networks (CNN) have been proposed to address segmentation task. The first network architecture has been proposed by Long *et al.* [LSD15]. It is a Fully Convolutional Network (FCCN) able to achieve a semantic segmentation of natural images. However in these primary works, pixel classification is considered inadequate to achieve accurate segmentation since pixel localisation, which is essential, is ignored. Thus Ronneberg *et al.* have proposed a new architecture called U-Net [RFB15] to address this limitation. U-Net is a CNN based on encoder-decoder operations: i) the encoder compresses the input data into a latent-space representation, and, ii) the decoder aims at predicting the output from the latent-space representation [BKC17].

The U-Net design is one of the most well-known architecture for segmentation. A lot of variants have been proposed to address several applications, such as segmentation of *Drosophila* cells in microscopy images [RFB15] or road recognition in natural images [ZLW18], for instance. U-Net network also address 3D segmentation thanks to 3D convolutional layers. For example, Cicek *et al.* [ÇAL⁺16] have designed a 3D U-Net with a ReLU activation function to segment kidney in confocal microscopic images.

The reason why U-Net and its variants are suitable for medical image segmentation is that its structure can simultaneously combine low and high level information. The low-level information aims at improving accuracy while the high-level information helps to extract complex features [LSLZ21]. Classical U-Net architecture is based on a binary cross-entropy (BCE) [RFB15] loss function suitable for balanced classes. In our study case, class distribution is very imbalanced so that using such a loss function could lead to roughly classifying background only. A DICE loss function is more dedi-

	U-Net [RFB15]	V-Net [MNA16]	RED-CNN [HZRS16]	RED-Net [MSY16]
Original Application	Segmentation	Segmentation	Denoising	Restoration
Original Dimension	2D	3D	2D	2D
3D Capabilities	Yes	Yes	Yes	Yes
Activation function	ReLU	PReLU	ReLU	ReLU
Sampling function (encoder path)	Maximum Pooling	Convolution	Convolution	Convolution
Sampling function (decoder path)	Transposed Convolution	Transposed Convolution	Transposed Convolution	Transposed Convolution
Skip connectors between encoder & decoder	Vector concatenation	Vector concatenation	Scalar sum	Scalar sum
Feature map size variations	Yes	Yes	No	No
Number of floors	4 + Bottleneck	4 + Bottleneck	—	—
Total number of Convolutional layers	28	23	10	10 to 30
Loss function	(C) BCE	(G) DICE	(A) MSE	(B) MSE
Other experimented loss	(F) DICE	-	(D) DICE	(E) DICE

Table 1: Encoder-decoder characteristics. Notations (A) to (G) are refers to discussion Table 5.

cated to our use-case since it only scores estimated foreground pixels. This U-Net variant, denoted V-Net, has been first proposed by Milletari *et al.*[MNA16]. Another major design difference between U-Net and V-Net consists of using convolution layers instead of pooling layers to down/up-sample between each level of the network.

Residual Networks[HZRS16] are deep learning networks mainly made of convolutions and skip connections. Their architecture is based on a deep succession of convolutional layers which have empirically showed an accuracy gain over shallower networks when coupled with skip connections. These skipped connections take places between two not successive layers to address the vanishing/exploding gradients problem by adding a shallow residual mapping to a deeper layer input. Our investigation also focuses on RED-CNN and RED-Net which are residual networks, but also encoder-decoder such as the previously introduced U-Net and V-Net. RED-CNN (Residual Encoder-Decoder Convolutional Neural Network) has been developed for low-dose CT imaging. It has been demonstrated in [HZRS16] that such a network is particularly efficient for preserving structural information while reducing noise. Despite RED-CNN has been proposed for segmenting tomograms reconstructed from a large amount of low-dose X-Ray radiographs, we investigate this network since the noise / artefacts it deals with are quite similar to those observed on our degraded tomograms. We also focus on the capabilities of the RED-Net network which is a very deep fully convolutional encoder-decoder network for image restoration. Since denoising is part of the very de-

graded CT tomogram segmentation problem, RED-Net architecture could provide some ideas to address our task. Usually, RED-Net performs ten to thirty successive convolutions (depending on the version), and is also coupled with skip connections. A summary of the characteristics of all the state-of-the-art networks (U-Net, V-Net, RED-CNN, RED-Net) explored in this study is provided in the Table. 1.

4 OPTIMAL ENCODER-DECODER NETWORKS

In addition to the encoder-decoder networks detailed in section 3, we introduce in this section two new encoder-decoder networks specifically addressing the segmentation of very degraded brain tomograms (Fig. 3). We justify our architectural choices in accordance with our data and the other networks architectures. Each level of the encoder path is composed of a succession of convolutional layers + batch normalization + PReLU activation function ; maximum pooling (network I) or convolutions with strides (network H) are applied between each down-sampling step. Symmetrically, the decoder part is composed of a succession of transposed convolutional layers with concatenations of symmetric encoder feature maps. The final up-sampling step of the decoder is finalized by a sigmoid function coupled with a binary threshold in order to achieve the segmentation.

On top of that, batch normalization is added to speed up the training process enabling a higher learning rate [IS15], while PReLU activation function generalizes the traditional rectified unit and improves network fitting with nearly zero extra computational cost [HZRS15]. We investigated on two variations

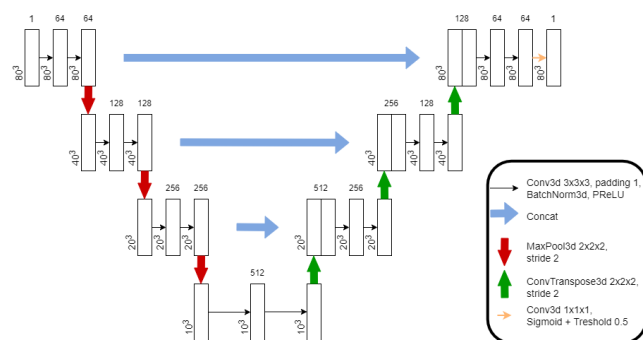


Figure 3: New investigated networks architecture and summary of their principal characteristics.

Our Application	Segmentation of widely unbalanced data in very noisy tomograms
Act. function	PReLU
Sampling function (encoder path)	Batch normalization + (H) Convolution (I) Maximum Pooling
Sampling function (decoder path)	Transposed Convolution
Skip connectors	Vector concatenation
Number of floors	4
Total number of Convolutional layers	18
Loss function	Sigmoid + DICE + Threshold

of this architecture assuming: i) network (I) with maximum pooling will maximize Recall (Table. 4); ii) network (H) with convolutions will maximize Precision (Table. 4). Finally, the network minimizes a DICE loss function at the end since it has been already demonstrated it is suitable for accounting on very imbalanced foreground data. However, this minimization is completed with a sigmoid + threshold in order to achieve the binary segmentation. Finally, the overall networks can be designed to work in 2D to perform a segmentation slice by slice, or directly in 3D. Thus both designs have been implemented and results are discussed in the following section.

Second we experiment several level of pre-filtering over the training data to verify the consideration of imbalanced background during the training. All networks investigated in this paper are trained with: i) no filtering (i.e. all patches are considered), ii) > 0 filter (only the patches containing at least one foreground pixel are used), and, iii) $> 5\%$ filter (only the patches containing at least 5% of foreground pixels are used).

Prediction results are given as an illustration on Table 2 for our network (I), but similar behavior is observed for all networks. It leads to a performance superiority for trained networks excluding patches without any data of interest, slightly better than the classic not filtering process, but highly better than the 5% filtering. Considering the reduced amount of training data from a higher filtering rate, we investigated on this potential bias. Prediction results of the filtering process with a same number of training data are given Table 3. The ranking between the processing methods remains the same with still a huge gap between > 0 and $> 5\%$ filtering methods and a superiority of the > 0 filtering over the no filtering method. Training on too much background (no filtering) lowers the features of interest over background ratio while eliminating too much foreground ($> 5\%$ filtering) lowers the absolute features of interest information to train on. The > 0 filtering method take the fullest information while increasing the features of

interest over background ratio leading to higher performances.

5 DISCUSSION

In this section, we first compare our proposed networks to the state-of-the-art encoder-decoders through their segmentation performances from very degraded brain CT tomograms. Notice that all networks are trained using the optimal pre-filtering explained in previous section. Then, since network design can be considered in 2D or in 3D, we compare both design to estimate the quality gain of 3D (which is computationally widely more gloutonous, thus requiring HPC or GPU-box computers) over to 2D (rapidly achievable on a general purpose computer).

Metric comparisons are based on the standard confusion matrix composed of true positive (tp), true negative (tn), false positive (fp) and false negative (fn) ratio compared to ground truth. We also discuss the results using the Precision, Recall, Dice and Jaccard metrics, reminded in the Table 4. Precision (ratio of correctly segmented voxels among all predicted foreground) and Recall (ratio of correctly segmented foreground voxels compared to ground truth) are complementary: a high Precision coupled with a high Recall means a high-quality prediction of the network. Conversely, if one of them is not high enough, the prediction quality downgrades. Dice and Jaccard values are mentioned since they state the similarity and the diversity between predictions and ground truth.

As already mentioned, in all our experiments: i) full-scan are processed with 1800 radiographs while fast-scans only contain 100 radiographs ; ii) FBP is used for all CT reconstructions, and, iii) an adaptive Renyi entropy based segmentation is used from high-quality tomograms to obtain the training target and ground truths to be compared with network predictions.

5.1 Networks comparison

Prediction results are given on Table 5 for networks trained on fast-scans containing only 100 radiographs.

Filters	nb patches	TP	FP	FN	TN	\mathcal{P}	\mathcal{R}	\mathcal{D}	\mathcal{J}
no filter	9680	18020	2280	10361	3066939	0.887	0.634	0.740	0.587
> 0	7294	21261	3703	7120	3065516	0.852	0.749	0.797	0.663
> 5%	257	25118	222378	3262	2846842	0.101	0.885	0.182	0.100

Table 2: Processing filtered data (maximum data)

Filters	nb patches	TP	FP	FN	TN	\mathcal{P}	\mathcal{R}	\mathcal{D}	\mathcal{J}
no filter	257	17979	14646	10402	3054573	0.551	0.633	0.589	0.418
> 0	257	17764	10697	10617	3058522	0.624	0.626	0.625	0.455
> 5%	257	25118	222378	3262	2846842	0.101	0.885	0.182	0.100

Table 3: Processing filtered data (reduced data)

Precision \uparrow	Recall \uparrow	Dice \uparrow	Jaccard \uparrow
$\mathcal{P} = \frac{tp}{tp+fp}$	$\mathcal{R} = \frac{tp}{tp+fn}$	$\mathcal{D} = \frac{2*tp}{2*tp+fp+fn}$	$\mathcal{J} = \frac{tp}{tp+fp+fn}$

Table 4: Definition of Prediction, Recall, Dice and Jaccard metrics.

Methods	TP	FP	FN	TN	\mathcal{P}	\mathcal{R}	\mathcal{D}	\mathcal{J}
Trad	28381	3067457	0	1762	0.009	1.0	0.018	0.009
(A) RED-CNN (MSE)	0	0	28381	3069219	0	0	0	0
(B) RED-Net (MSE)	0	0	28381	3069219	0	0	0	0
(C) U-Net (BCE)	0	0	28381	3069219	0	0	0	0
(D) RED-CNN (DICE)	0	0	28381	3069219	0	0	0	0
(E) RED-Net (DICE)	0	0	28381	3069219	0	0	0	0
(F) U-Net (DICE)	19194	2451	9186	3066769	0.887	0.676	0.767	0.623
(G) V-Net (DICE)	20132	3380	8249	3065839	0.856	0.709	0.776	0.634
(H) Our Network (conv)	18715	2212	9665	3067008	0.894	0.659	0.759	0.612
(I) Our Network (maxP)	21261	3703	7120	3065516	0.852	0.749	0.797	0.663

Table 5: Encoder-decoder performances overview (tomogram reconstructed with 100 radiographs)

In that case, the traditional segmentation performances lead to a nearly white-only pixel prediction. Among the nine encoder-decoder networks, five are not efficient and perform a black pixel only prediction: i) Networks (A) with BCE loss and (B) & (C) using MSE loss which are weak against hugely imbalanced classification tasks; ii) (D) RED-CNN and (E) RED-Net which are customized with a DICE loss but are primarily designed for denoising and restoration thus justifying their inefficiency to our use-case. The four working encoder-decoders are (F) U-Net customized with DICE, (G) V-Net, and our networks, (H) and (I), which lead to a DICE score of 0.759 to 0.797 (*resp.*) compared to the 0.018 for the traditional method. Each of the four networks is built using vector concatenation as skip connectors between encoder and decoder, coupled with feature map size variations and a DICE loss function. Note that it is not the case for any of the networks which does not work (i.e. (A), (B), (C), (D) and (E)).

Investigating on the design specificities of the four networks linked to their performance ranking, we were not able to highlight some evidence. Especially we ex-

pected: i) (F) and (I) networks with maximum pooling to maximize Recall; ii) (G) and (H) networks with convolutions to maximize Precision; iii) (H) and (I) networks with fewer floors to be ranked better as they managed to process more information in the network.

5.2 2D versus 3D DL segmentation

We investigate now our deep learning segmentation model using independent slices (2D segmentation of slices) and 3D (tomogram segmentation). As highlighted in Figure 4 and on the Table 6 metric comparison between 2D and 3D designs, there is a huge performance gap between 2D and 3D processing for our particular task. Processing 3D volumes brings shapes continuity, which should explain the performance superiority. Indeed, considering the tomogram as a whole results in 3D patches decomposition in the network model. Such a volumetric representation propagates the 3D morphological context of the features to segment through the latent space of the network, which would not be the case if the tomogram were considered slice by slice.

Dimension	TP	FP	FN	TN	\mathcal{P}	\mathcal{R}	\mathcal{D}	\mathcal{J}
2D	25577	15043	31457	3025523	0.629	0.448	0.523	0.354
3D	39630	4679	17404	3035887	0.894	0.694	0.782	0.642

Table 6: Tomogram processing (3D model) against independent slice processing (2D slice-by-slice segmentation)

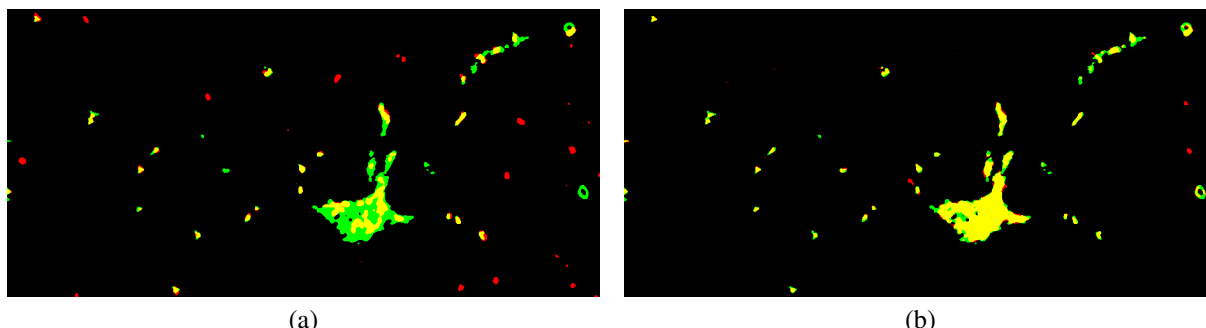


Figure 4: Comparison of 2D model result (a) and 3D model result (b) compared to ground truth (Yellow=TP, Red=FP, Green=FN, Black=TN).

6 CONCLUSION

Being able to reconstruct whole brain network from degraded tomograms remains a challenge as the traditional algorithms are not able to produce useful results. We have investigated several models of deep learning methods and proposed new approaches. From this whole set of experiments we can observe that encoder-decoder performances highly overcome traditional segmentation for 100 radiographs fast-scan processing and such models are good candidates to achieve our addressed task: providing a good quality segmentation from degraded tomograms reconstructed with a limited number of radiographs. More with a small number of layers the best networks are able to manage 3D images and allow to extract information coming from the whole brain conformation. Future works will investigate post processing methods to reconstruct the brain neuronal structures, for example by introducing graph concept into the deep-learning network designs in order to connect positive pixels, i.e. to directly recover neuronal structures instead of the subset of voxels composing it.

7 REFERENCES

- [And89] Anders H Andersen. Algebraic Reconstruction in CT from Limited Views, 1989.
- [BKC17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.
- [ÇAL⁺16] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.
- [DMND⁺00] B. De Man, J. Nuyts, P. Dupont, G. Marchal, and P. Suetens. Reduction of metal streak artifacts in x-ray computed tomography using a transmission maximum a posteriori algorithm. *IEEE Transactions on Nuclear Science*, 47(3):977–981, 2000.
- [EF99] Hakan Erdogan and Jeffrey A Fessler. Ordered subsets algorithms for transmission tomography. *Physics in Medicine & Biology*, 44(11):2835, 1999.
- [FZL⁺17] Yuncong Feng, Haiying Zhao, Xiongfei Li, Xiaoli Zhang, and Hongpeng Li. A multi-scale 3d otsu thresholding algorithm for medical image segmentation. *Digital Signal Processing*, 60:186–199, 2017.
- [GG96] Huaiqun Guan and Richard Gordon. Computed tomography using algebraic reconstruction techniques (ARTs) with different projection access schemes: a comparison study under practical situations. *Physics in Medicine and Biology*, 41(9):1727–1743, sep 1996.
- [Han81] Kenneth M Hanson. Technical aspects of computed tomography. *Radiology of the Skull and Brain*, 5(1):3941–3955, 1981.
- [HGD⁺10] Benoit Hamelin, Yves Goussard, Jean-Pierre Dussault, Guy Cloutier, Gilles

- Beaudoin, and Gilles Soulez. Design of iterative roi transmission tomography reconstruction procedures and image quality analysis. *Medical physics*, 37(9):4577–4589, 2010.
- [HL89] Tom Hebert and Richard Leahy. A generalized em algorithm for 3-d bayesian reconstruction from poisson data using gibbs priors. *IEEE transactions on medical imaging*, 8(2):194–202, 1989.
- [HNY⁺13] Jiang Hsieh, Brian Nett, Zhou Yu, Ken Sauer, Jean Baptiste Thibault, and Charles A Bouman. Recent Advances in CT Image Reconstruction. *Current Radiology Reports*, 1(1):39–51, 2013.
- [HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1024–1034, December 2015.
- [HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.
- [IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 448–456. PMLR, 2015.
- [JW03] Ming Jiang and Ge Wang. Convergence of the simultaneous algebraic reconstruction technique (sart). *IEEE Transactions on Image Processing*, 12(8):957–961, 2003.
- [KB98] Chris Kamphuis and Freek J. Beekman. Accelerated iterative transmission CT reconstruction using an ordered subsets convex algorithm. *IEEE Transactions on Medical Imaging*, 17(6):1101–1105, 1998.
- [Kol05] J. S. Kole. Statistical image reconstruction for transmission tomography using relaxed ordered subset algorithms. *Physics in Medicine and Biology*, 50(7):1533–1545, 2005.
- [KR14] Satish Kumar and Gurdas Ram. A generalization of the Havrda-Charvat and Tsallis entropy and its axiomatic characterization. *Abstract and Applied Analysis*, 2014(5), 2014.
- [KS01] Avinash C. Kak and Malcolm Slaney. 7. Algebraic Reconstruction Algorithms. In *Principles of Computerized Tomographic Imaging*, pages 275–296. SIAM, 2001.
- [LHW⁺15] Minji Lee, Yoseob Han, John Paul Ward, Michael Unser, and Jong Chul Ye. Interior tomography using 1d generalized total variation. part ii: Multiscale implementation. *SIAM Journal on Imaging Sciences*, 8(4):2452–2486, 2015.
- [LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR Computer Vision and Pattern Recognition)*, pages 3431–3440, june 2015.
- [LSLZ21] Xiangbin Liu, Liping Song, Shuai Liu, and Yudong Zhang. A review of deep-learning-based medical image segmentation methods. *Sustainability*, 13(3):1224, 2021.
- [MKL⁺16] Glenn R Myers, Andrew M Kingston, Shane J Latham, Benoît Recur, Thomas Li, Michael L Turner, Levi Beeching, and Adrian P Sheppard. Rapidly converging multigrid reconstruction of cone-beam tomographic data. In *Developments in X-Ray tomography X*, volume 9967, page 99671M. International Society for Optics and Photonics, 2016.
- [MNA16] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016.
- [MSY16] Xiaojiao Mao, Chunhua Shen, and Yubin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.
- [NW02] F. Natterer and Ge Wang. The Mathematics of Computerized Tomography. *Medical Physics*, 29(1):107–108, jan 2002.

- [PGF⁺05] Thammanit Pipatsrisawat, Aca Gačić, Franz Franchetti, Markus Püschel, and José M.F. Moura. Performance analysis of the filtered backprojection image reconstruction algorithms. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume V, 2005.
- [PM17] Pierre Paleo and Alessandro Mirone. Efficient implementation of a local tomography reconstruction algorithm. *Advanced structural and chemical imaging*, 3(1):1–15, 2017.
- [RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [RFK⁺14] Benoit Recur, Mathias Fauconneau, Andrew Kingston, Glenn Myers, and Adrian Sheppard. Iterative reconstruction optimisations for high angle cone-beam micro-ct. In *Developments in X-Ray Tomography IX*, volume 9212, pages 288–299. International Society for Optics and Photonics, 2014.
- [RXT07] Dmitri Riabkov, Xinwei Xue, and Dave Tubbs. Accelerated cone-beam backprojection using GPU-CPU hardware. *Proceedings of the 9th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pages 68–71, 2007.
- [San04] Bulent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146, 2004.
- [SKKH07] Holger Scherl, Benjamin Keck, Markus Kowarschik, and Joachim Hornegger. Fast GPU-based CT reconstruction using the Common Unified Device Architecture (CUDA). In *IEEE Nuclear Science Symposium Conference Record*, volume 6, pages 4464–4466, 2007.
- [SKR⁺14] Emil Y Sidky, David N Kraemer, Erin G Roth, Christer Ullberg, Ingrid S Reiser, and Xiaochuan Pan. Analysis of iterative region-of-interest image reconstruction for x-ray computed tomography. *Journal of Medical Imaging*, 1(3):031007, 2014.
- [SLH⁺23] Anton PJ Stampfl, Zhongdong Liu, Jun Hu, Kei Sawada, H Takano, Yoshiki Kohmura, Tetsuya Ishikawa, Jae-Hong Lim, Jung-Ho Je, Chian-Ming Low, et al. Synapse: An international roadmap to large brain imaging. *Physics Reports*, 999:1–60, 2023.
- [SRR08] Ahmad Adel Abu Shareha, Mandava Rajeswari, and Dhanesh Ramachandram. Textured renyi entropy for image thresholding. In *Proceedings - Computer Graphics, Imaging and Visualisation, Modern Techniques and Applications, CGIV*, pages 185–192, 2008.
- [Tof96] Peter Toft. The radon transform. *Theory and Implementation (Ph. D. Dissertation)(Copenhagen: Technical University of Denmark)*, 1996.
- [ZHZ09] Xing Zhao, Jing-jing Hu, and Peng Zhang. Gpu-based 3d cone-beam ct image reconstruction for large data volume. *International Journal of Biomedical Imaging*, 2009:1–8, 2009.
- [ZLW18] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.
- [ZNG08] Andy Ziegler, Tim Nielsen, and Michael Grass. Iterative reconstruction of a region of interest for transmission tomography. *Medical physics*, 35(4):1317–1327, 2008.
- [ZWZ⁺18] Hao Zhang, Jing Wang, Dong Zeng, Xi Tao, and Jianhua Ma. Regularization strategies in statistical image reconstruction of low-dose x-ray CT: A review, 2018.

Self-Checkout Product Class Verification using Center Loss approach

Bernardas Ciapas
Institute of Data Science
and Digital Technologies
Vilnius University
Akademijos str. 4
Vilnius, LT-08663,
Lithuania
bernardas.ciapas@mif.vu.lt

Povilas Treigys, Ph.D.
Institute of Data Science
and Digital Technologies
Vilnius University
Akademijos str. 4
Vilnius, LT-08663,
Lithuania
povilas.treigys@mif.vu.lt

ABSTRACT

The traditional image classifiers are not capable to verify if samples belong to specified classes due to several reasons: classifiers do not provide boundaries between in-class and out-of-class samples; although classifiers provide separation boundaries between known classes, classifiers' latent features tend to have high intra-class variance; classifiers often predict high probabilities for out-of-distribution samples; training classifiers on unbalanced data results in bias towards over-represented classes. The nature of the class verification problem requires a different loss function than the ubiquitous cross entropy loss in traditional classifiers: input to a class verification function includes a suggested class in addition to an image. As opposed to outlier detection, space is transformed to be not only separable, but discriminative between in-class and out-of-class inputs. In this paper, class verification based on a euclidean distance from the class centre is proposed and implemented. Class centres are learnt by training on a centre loss function. The method's effectiveness is shown on a self-checkout image dataset of 194 food retail products. The results show that a two-fold loss function is not only useful to verify class, but does not degrade classification performance - thus, the same neural network is usable both for classification and verification.

Keywords

Self-checkout images, class verification, centre loss, outlier detection.

1 INTRODUCTION

Real-world computer vision tasks include a need to verify claims that an image contains a claimed type of object. A popular research area of class verification is face verification ([1], [15], [18], [14], [21], [5], [13]), where a class represents a person. In face verification, the computer vision task is to verify if a person in an image is the same person he/she claims to be. The negative samples are usually ID of another person than in the image. Another popular research area in class verification is predicting image authenticity, given an image and a class in that image ([8], [11]). The negative examples are usually images generated by conditional generative adversarial networks (GAN).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

In food retail self-checkouts, a computer vision task attempts to check if a product in an image is the same product as a customer's chosen one. Negative samples are images of products other than the customer's choice.

The class verification task is a binary classification task that takes two inputs: an image and a label ([21], [2]). A label is one of the trained classes. An image contains one of the following: a) an object of the same class as the label b) an object of a different trained class than the label or c) out-of-distribution input (OOD - any object of the unknown class or no object at all). The goal of the class verification task is to separate a) ("Correct") from the rest ("Incorrect") - whether an object in an image belongs to the claimed class or not. Class verification does not need to distinguish between b) and c) - whether an object in an image is of any other known class, or an unknown class, or does not contain an object at all.

The class verification task differs from other classic computer vision tasks - classification and detection. Multi-class classification algorithms predict the dis-

tribution of probabilities only among a list of known classes. An image containing an object of an unknown class passed to a classification neural network leads to unpredictable results, whereas the class verification task requires it to be rejected as "Incorrect" no matter what label is passed as input. Passing an image containing a known class' object to a classifier is likely to yield a higher probability for the correct class. Still, the probability boundary that separates correct class from incorrect is unknown. Thus, classification networks cannot be used for verifying class identities directly. Detection networks usually consist of two steps: 1) predicting object patches with the highest objectness probabilities (whether an object of a known class exists) and 2) predicting probability distributions of the patches between the known classes (classifying). Thus, detection algorithm errors in predicting objectness are penalized differently from errors classifying known objects. Nevertheless, class verification tasks are indifferent to the existence of objects other than the claimed class. Training detection networks require labels with bounding boxes around the objects, but class verification tasks are indifferent to object location within images - both during training and inference. Detection networks usually classify image crops having the highest objectness scores in a similar way as classification networks: they distribute class probabilities among the list of known classes. Therefore, a similar lack of boundary between correct and incorrect classes in detection networks makes them directly unusable to verify classes.

Despite the lack of proper loss function for verifying classes in image classification and detection neural networks, their ability to extract image features has been widely demonstrated [25], [19], [10]. Knowledge transfer is widely used between different tasks. Therefore, the backbones of neural classification or detection neural networks are likely useful in class verification if the loss function is changed.

Class verification task relates to conditional outlier detection task ([16], [17]). Outlier detection algorithms draw a boundary between in-distribution and out-of-distribution samples, and judge new samples based their relation to that boundary. However, outlier detection tasks do not make an explicit attempt to transform space in such a way that all samples (of a single class) are placed nearby.

Class verification task has a wide variety of applications. In the context of face verification, the suggested class comes from a presented ID, which must be confirmed by class verification. In the context of self-checkouts, the suggested class is a customer's

chosen product from a picklist menu, which must be confirmed by class verification.

2 RELATED WORKS

Images are multi-dimensional data points that must be reduced in dimensionality prior to applying machine learning techniques. The most common by far are convolutional neural networks. [24], [4] use a fully connected layer of a classifier.

Outlier detection estimators judge samples by learnt boundary between in-distribution and out-of-distribution samples. Boundary shapes differ by method: robust covariance [17] learns ellipsoid-, one-class SVM [7] learns hyperplane-, isolation forest [6] learns any-shaped boundaries. This research uses a simple hypersphere-shaped boundary. However, our loss function pushes latent space variables of any class to the same point ("class centre"), thus a centre-enclosing hypersphere-shaped boundary suffices.

The class prototype is a generalization of multiple data samples of a single class. Multiple research attempts have been made to derive a class prototype given a set of data samples. [9] uses the term "class prototype in a semantic space", which is category vectors (one per category). They construct the category vectors by using auxiliary textual information about the classes of interest. We do not use any textual or other information about the classes to train category vectors - mostly because discriminative textual information is not easily obtainable for the country-, chain-, or store-specific classes of self-checkout products.

A typical task of class verification is a well-researched face verification. Siamese network in [18] effectively learns a distinction function - whether two images belong to the same class (person) or not. It consists of two identical networks with shared weights and a distinction layer that measures the euclidean distance between embeddings of a fully connected layer. A similar concept is employed in Triplet Loss [14], except it uses three images to calculate a loss function: an anchor, a positive (same class/persons') and a negative (another person's). The Anchor+Positive pair is trained to output an opposite value than the Anchor+Negative pair. Both distinction function-based methods - Siamese and Triplet - require reference images (or their embeddings) during inference. Although that's usually satisfiable when a number of images per class is small, using big training datasets faces several challenges: first, different reference images lead to different verification results; second, inferring against multitude of reference images is rarely feasible due to performance and storage reasons.

Research in artificial intelligence safety attempts to verify if the input is consistent with known (in-distribution) samples. Deep Verifier Networks (DVN) [2] use an autoencoder's latent layer's activations to estimate the density of known samples. Samples with latent activations inconsistent with the density model are rejected as adversarial. DVN does not attempt to model latent space where intra-class samples are clustered together. Thus DVN does not derive class prototypes. Since our "adversarial" samples are images of other than the declared class, we suggest that deriving a class prototype is meaningful.

Another way to verify class is to derive a class prototype during training and then compare an input sample against the prototype during inference. The Discriminative Feature Learning [21] derives a class centre using a neural net's latent layer's activations. They use a two-fold loss function: one member is a standard classifier's cross-entropy loss; another is the euclidean distance from a class prototype's centre. Class centres are updated in every iteration, thus "learned". Training such a two-fold loss function pushes the latent layer activations closer together for samples of the same classes. The first member of such a loss function - cross-entropy - ensures that different class centres are separable, i.e. do not regress to the same point. They perform extensive experiments to pick the best weight between the summands of the loss function. In this article, authors use the same two-fold loss function (Eq. 1) in the experiments. In addition, authors perform experiments on the best size of the latent layer. Discriminative Feature Learning focuses on verification and only uses cross-entropy loss to separate class centres but does not provide classification results. We recognize that classification results are as important as verification results, thus provide both and compare classification-only versus classifier-plus-verifier results.

The loss function of class-prototype-based methods measure the distance between class prototype's and a sample's embeddings. SphereFace [13], ArcFace [5] measure the angular distance between class prototype's and sample's embeddings, then modify cross-entropy loss function to use angular distances. They show better discrimination of inter-class features than regular cross-entropy. Their unfold loss function does not allow to adjust classification vs. verification relative importance, but this research' loss function (Eq. 1) allows it using λ hyperparameter. Since this research' primary focus is verification, and authors only include cross-entropy loss in order to preserve class separability, i.e. not to regress all class centres to the same point, it is important to adjust this relative importance.

3 METHODS

3.1 Dataset

Authors used the same dataset of retail products in the self-checkout environment as in their other research articles [3] and [4]. The dataset contains 194 different food retail products that do not usually carry barcodes. Thus, they need to be identified using different methods at the time of checkout. The training dataset was balanced by data augmentation and contained roughly 10K different images per class, most being augmented variations of original images. Neither the testing nor training set included out-of-distribution samples - samples of unknown classes. All the negative samples (i.e. "Incorrect" selections) were generated by labelling an image with one of the available classes other than the correct class. Out-of-distribution samples were not included due to difficulties in collecting them. Authors recognize that including out-of-distribution samples might be helpful in further research.

3.2 Architecture Details

Authors started architecture experiments with their own individual class classifier's backbone that is explained in detail in [3]. The classifier's backbone contains 7 convolutional and 2 dense blocks. Each block contains a Convolutional or a Dense layer, followed by a Batch-Norm layer, followed by a ReLU activation. Each convolutional layer is followed by a MaxPool layer. It was shown to perform in other research papers [3] and [4] on the same self-checkout dataset. Presumably, this implies that the architecture is fit to carry enough information through the network layers about the classness of sample images. In addition, the architecture contains little parameters (3.2mln) compared to leading architectures on big sets like ImageNet - CoCa [22] (2100mln), ViT-G/14 [23] (1843mln), EfficientNet [20] (11mln and up).

In addition to the original classifier, the authors added a Center Loss layer (explained below) and an Extra Dense layer. The final model architecture is shown in Fig.1. Experiments were performed, and results were reported using different sizes of the Extra Dense layers. Training without the Extra Dense layer did not saturate the loss function.

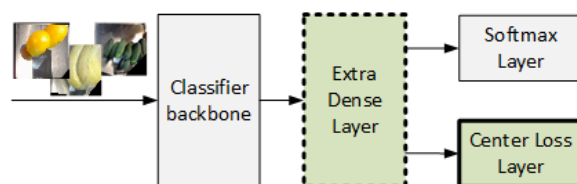


Figure 1: Model architecture

3.3 Center Loss layer

The Center Loss (CL) layer takes two inputs: activations $\in \mathbb{R}^{m \times cnt}$ (m - number of minibatch samples; cnt - count of neurons in the extra dense layer) and image labels (m one-hot vectors). It has an internal parameter of class centres $\in \mathbb{R}^{n \times cnt}$ (n - number of classes). The output of the CL layer is the difference vector $\in \mathbb{R}^{m \times cnt}$ between samples' corresponding class centres and samples' activations of the extra dense layer.

3.4 Loss function

The class verification task's loss function should return a high value when the image does not contain an object of the claimed class and a low value when it does. Authors suggest a concept of a class centre (or a class prototype) - virtual data points in latent space, one per class. The design of the loss function was twofold: first, the intra-class proximity had to be developed; second, the inter-class difference had to be preserved. Such a loss function allows verifying if the sample belongs to the class based on its distance from other samples of that class.

The entire loss function (Eq. 1) is a weighted sum of Cross Entropy loss and Center loss, having a relative weight hyperparameter λ .

$$L = L_S + \lambda * L_C \quad (1)$$

where:

- L - Total Loss
- L_S - Cross Entropy Loss of Softmax
- λ - Center Loss weight (hyperparameter)
- L_C - Center Loss

Cross Entropy loss (Eq. 2) preserves differences between classes. Without Cross Entropy loss, the centres of all classes are likely to regress to one point. Cross Entropy loss does not minimize differences between various samples of the same class.

$$L_S = - \sum_{i=1}^m \log \frac{e^{W_{y_i}^T * x_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T * x_i + b_j}} \quad (2)$$

where:

- L_S - Cross Entropy Loss of Softmax
- m - Number of samples
- n - Number of classes
- x_i - i -th sample's activations extra dense layer
- y_i - i -th sample's label
- $W \in \mathbb{R}^{cnt \times n}$ - Weights, last dense layer
- $b \in \mathbb{R}^n$ - Biases, last dense layer
- cnt - Count neurons, extra dense layer (hyper-p)

Center Loss (Eq. 3) aims to minimize the distance between various intra-class samples. A concept of the class centre is introduced: it is an average vector of all samples in that class of the extra dense layer's activations. The sample's distance from the class centre is calculated as L_2 norm of the difference between the sample's and the class centre's vector. Although other distance types than Euclidean are available, authors limited this research to L_2 only.

$$L_C = \frac{1}{2} \sum_{i=1}^m \|x_i - c_{y_i}\|_2^2 \quad (3)$$

where:

- L_C - Center Loss
- m - Number of samples
- x_i - i -th sample's activations of the extra dense layer
- y_i - i -th sample's label
- c_{y_i} - Center of the y_i -th class

Whereas centre loss attempts to minimize the distance between a "class centre" and samples of that class, it does not attempt to maximize inter-class distances. Although it is possible extending the loss function to punish low inter-class distances would result in an even better separation of classes, authors left it out of this research.

3.5 Training Details

The Center Loss layer did not have any trainable parameters updateable by gradient descent. Yet, the internal parameter of class centres was updated in each iteration: only centres of the classes represented by the samples in a minibatch were updated, whereas unrepresented class centres were left untouched. Class centres were updated as shown in Eq. 4.

$$Center = Center + \alpha * (Activations - Center) \quad (4)$$

where:

- $Center$ - center of a sample's class $\in \mathbb{R}^{cnt}$
- $Activations$ - extra dense layer's activations $\in \mathbb{R}^{cnt}$
- α - center's learning rate (hyperparameter)
- cnt - count of neurons extra dense layer (hyper-p)

Authors trained for up to ten epochs with a patience criteria of five epochs (i.e. training was stopped and best weights restored if the five last epochs of training did not improve the validation loss function value). Training on a relatively big training set of two million images (about 10 thousand per class) usually saturated in the

first 1-2 epochs. Therefore a maximum of 10 epochs was never reached. The criteria for the best weights selection and early stopping was total validation loss, which is the sum of Softmax layer loss and Centre loss. Adam [12] optimizer with a default learning rate of 0.001 was used.

Authors had to choose a proper λ value (relative weight of Centre loss vs Softmax' loss). The value was chosen 0.1 from previous experiments [21]. Finding the best value of λ was left out of scope of this paper and needs to be investigated in further research.

The training was executed on a GPU Nvidia Tesla V100-SXM2-32GB. The training duration of 1 epoch on about two million images varied between 45 and 55 minutes.

3.6 Testing Details

For every test image, authors measured the distance from every class centre. That gave $m \times n$ measurements (m - dataset size; n - number of classes), of which m were positive ("Correct" selections) and $m \times (n - 1)$ were negative ("Incorrect" selections). The results were calculated by giving weight $(n - 1)$ to the positive measurements so that the "Correct" and the "Incorrect" classes were balanced. The test set did not contain any out-of-distribution (OOD) samples, i.e. samples outside the known list of classes.

4 RESULTS

The main result in Table 1 shows class verification Equal Error Rate (EER, or Error Rate@False Positive Rate=False Negative Rate) and Receiver Operating Characteristic's Area Under Curve (ROC AUC). Authors exclude neuron counts before near-saturation was reached.

Neuron Count	EER	ROC AUC
2048	0.073	0.978
1536	0.073	0.978
1024	0.076	0.976
768	0.073	0.979
512	0.076	0.974
256	0.110	0.956

Table 1: Equal Error Rate (EER) and ROC Area Under Curve (AUC) for various neuron counts in Extra Dense layer

Figure 2 displays ROC curves for various distances from centre thresholds and for various number of neurons in the extra dense layer. ROC Area Under Curve saturates when the Center Loss layer reaches approximately 512-768 neurons.

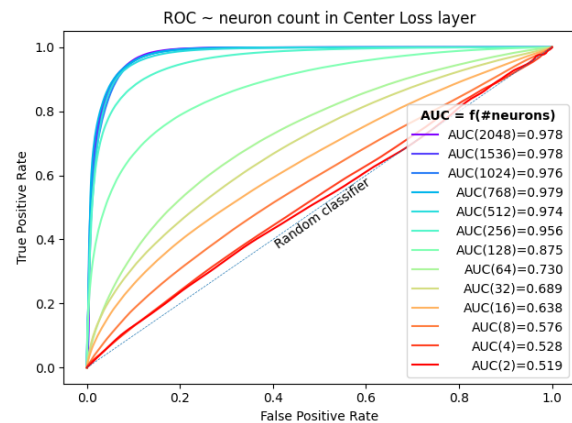


Figure 2: Receiver Operating Characteristic (ROC) curves for various numbers of neurons in the extra dense layer

Figure 3 shows ROC AUC's experimentally found dependency on the number of neurons in Extra Dense layer. AUC climbs steeply until it saturates at about 512 neurons. An increase in neuron count above 512 does not improve AUC.

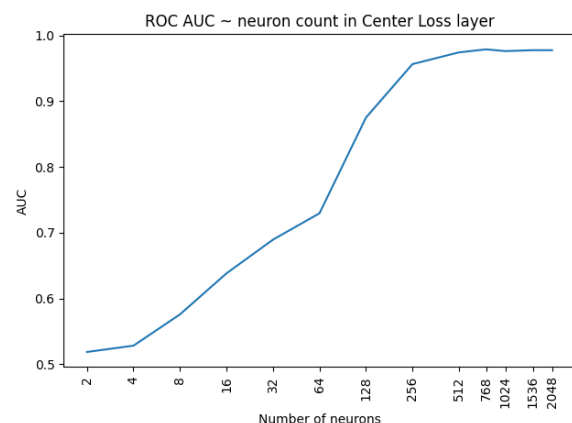


Figure 3: ROC Area Under Curve (AUC) dependency on the number of neurons in the extra dense layer

Figure 4 depicts the accuracy of the individual class classification. The original classifier performed at 73.2% accuracy on the validation set. Authors expected that an additional component of Center Loss in the Loss function would decrease the accuracy of individual class classification. However, the graph shows approximately the same individual classification accuracy when the Extra Dense layer contains at least 8 neurons: $73.2\% \pm 0.8\%$.

Figure 5 shows the relative importance of the Loss function summands: L_S (Softmax' cross-entropy loss)

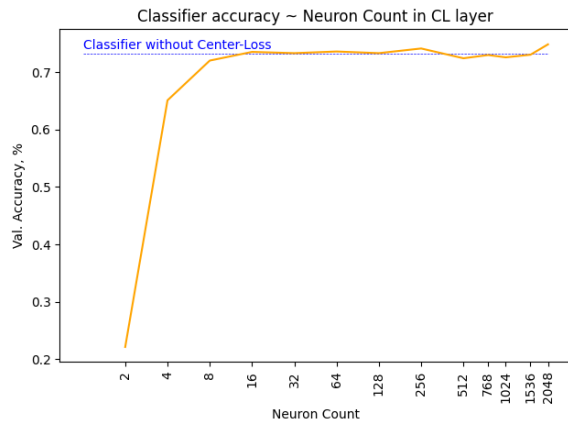


Figure 4: Classification accuracy dependency on the number of neurons in the extra dense layer

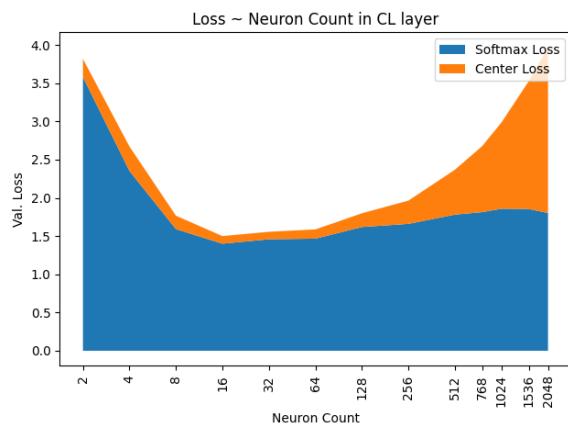


Figure 5: Loss (Softmax and Center) relative values and dependency on number of neurons in the extra dense layer

and L_C (CL layer loss). The Softmax's cross-entropy loss gains minimum value at approximately 8-16 neurons in the Extra Dense layer, then stays at about the same rate upon adding more neurons. This relates to the fact that classification accuracy also saturates at about the same 8 to 16 neurons. Center Loss obtains its minimum values between 8 and 128 neurons, then

risers steeply outside this range, mainly due to rising dimensionality of space where distances are calculated.

Figure 6 depicts sample distance distributions for the selected number of neurons in the Extra Dense layer. The separation between distances from samples' own class centres ("Correct" classes) versus from other class centres ("Incorrect" classes) increases with the increase of neurons in the Extra Dense Layer until saturation is reached. The mean distance of both - Correct and Incorrect - increases with the number of neurons.

Figure 7 shows sample images and their distances from selected class centers. Threshold at Equal Error Rate (Thr.@EER) mark separating line between positive (below the line) and negative (above the line) verification result, when False Positive Rate equals False Negative Rate.

4.1 Architecture experiments

Authors performed experiments without the Extra Dense layer shown in Figure 1. With Extra Dense excluded, training did not saturate the loss function and did not achieve satisfiable separation in distances between "Correct" and "Incorrect" classes.

5 ACKNOWLEDGMENTS

Authors express gratitude to Vilnius University Mathematics and Informatics department Information Technologies Open Access Center for providing high-performance computing (HPC) resources used in this research.

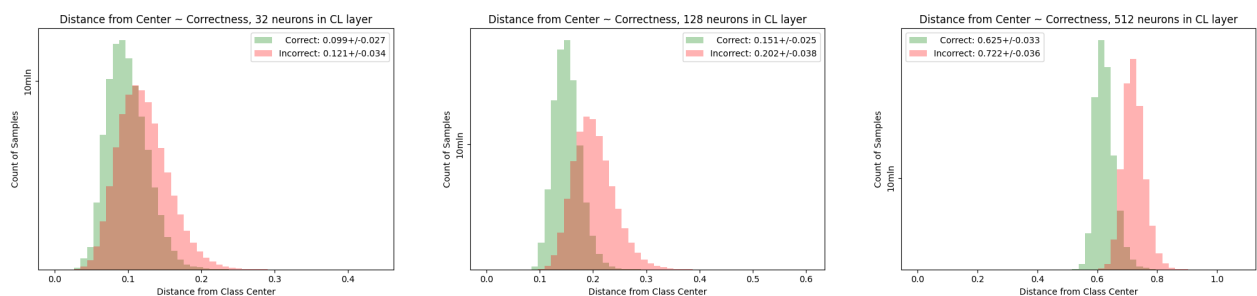


Figure 6: Distance distribution of CL layer activations from the same (Correct) vs other (Incorrect) class centres



Figure 7: Sample images and their distances from selected class centers

6 CONCLUSIONS

Our work reinforces the value of using the Center Loss function to verify sample's belonging to a class. In a self-checkout products dataset we received 92.7% verification accuracy (at Equal Error Rate). Authors discovered that adding a centre loss function to discriminate class features did not negatively affect classification accuracy: $73.2\% \pm 0.8\%$ (vs 73.2% without Center Loss). Thus the same neural network can be used both for classification and verification without sacrificing accuracy. We showed a minimum number of neurons is necessary for both classification accuracy and class verification accuracy to saturate. Once saturated, adding more neurons does not improve classification or verification accuracy.

7 REFERENCES

- [1] Ghaliya Alfarsi, Jasiya Jabbar, Ragad M Tawafak, Abir Alsidiri, and Maryam Alsinani. Techniques for face verification: Literature review. In *2019 International Arab Conference on Information Technology (ACIT)*, pages 107–112. IEEE, 2019.
- [2] Tong Che, Xiaofeng Liu, Site Li, Yubin Ge, Ruixiang Zhang, Caiming Xiong, and Yoshua Bengio. Deep verifier networks: Verification of deep discriminative models with deep generative models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7002–7010, 2021.
- [3] Bernardas Čiapas and Povilas Treigys. High F-score model for recognizing object visibility in images with occluded objects of interest. *Baltic journal of modern computing*, 9(1):35–48, 2021.
- [4] Bernardas Ciapas and Povilas Treigys. Retail Self-checkout Image Classification Performance: Similar Class Grouping or Individual Class Classification Approach. In *International Baltic Conference on Digital Business and Intelligent Systems*, pages 167–182. Springer, 2022.
- [5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.
- [6] Zhiguo Ding and Minrui Fei. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes*, 46(20):12–17, 2013.
- [7] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, 58:121–134, 2016.
- [8] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *International conference on computer aided verification*, pages 3–29. Springer, 2017.
- [9] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Learning Class Prototypes via Structure Alignment for Zero-Shot Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, sep 2018.
- [10] G Jignesh Chowdary, Narinder Singh Pun, Sanjay Kumar Sonbhadra, and Sonali Agarwal. Face mask detection using transfer learning of inceptionv3. In *International Conference on Big Data Analytics*, pages 81–90. Springer, 2020.
- [11] Sydney M Katz, Anthony L Corso, Christopher A Strong, and Mykel J Kochenderfer. Verification of image-based neural network controllers using generative models. *Journal of Aerospace Information Systems*, 19(9):574–584, 2022.
- [12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [13] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphreface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.
- [14] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

- [15] Azzam Sleit, R Abu-Hurra, and Wesam Al-mobaideen. Lower-quarter-based face verification using correlation filter. *The Imaging Science Journal*, 59(1):41–48, 2011. 2020.
- [16] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Conditional anomaly detection. *IEEE Transactions on knowledge and Data Engineering*, 19(5):631–645, 2007.
- [17] Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. Conditional Gaussian Distribution Learning for Open Set Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2020.
- [18] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lior Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, jun 2014.
- [19] Srikanth Tammina. Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP)*, 9(10):143–150, 2019.
- [20] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.
- [21] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.
- [22] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models, 2022. URL <https://arxiv.org/abs/2205.01917>, 2022.
- [23] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.
- [24] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover’s distance and structured classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12203–12213, 2020.
- [25] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76,

Why Existing Multimodal Crowd Counting Datasets Can Lead to Unfulfilled Expectations in Real-World Applications

Martin Thißen
Hochschule Darmstadt
Schöfferstraße 3
64295, Darmstadt, Germany

Prof. Dr. Elke Hergenröther
Hochschule Darmstadt
Schöfferstraße 3
64295, Darmstadt, Germany

ABSTRACT

More information leads to better decisions and predictions, right? Confirming this hypothesis, several studies concluded that the simultaneous use of optical and thermal images leads to better predictions in crowd counting. However, the way multimodal models extract enriched features from both modalities is not yet fully understood. Since the use of multimodal data usually increases the complexity, inference time, and memory requirements of the models, it is relevant to examine the differences and advantages of multimodal compared to monomodal models. In this work, all available multimodal datasets for crowd counting are used to investigate the differences between monomodal and multimodal models. To do so, we designed a monomodal architecture that considers the current state of research on monomodal crowd counting. In addition, several multimodal architectures have been developed using different multimodal learning strategies. The key components of the monomodal architecture are also used in the multimodal architectures to be able to answer whether multimodal models perform better in crowd counting in general. Surprisingly, no general answer to this question can be derived from the existing datasets. We found that the existing datasets hold a bias toward thermal images. This was determined by analyzing the relationship between the brightness of optical images and crowd count as well as examining the annotations made for each dataset. Since answering this question is important for future real-world applications of crowd counting, this paper establishes criteria for a potential dataset suitable for answering whether multimodal models perform better in crowd counting in general.

Keywords

Crowd Counting, Multimodal Learning, RGB-T, Transformer

1 INTRODUCTION

One of the biggest challenges of crowd counting in real-world applications is dealing with varying lighting conditions. Since crowd counting can be very important for event security and crowd monitoring, good performance independent of lighting conditions is essential for real-world applications. Especially at night, lighting is often poor, resulting in less contrast and information in optical images and thus reducing the accuracy of prediction models. In this case, thermal images are more suitable because they do not rely on visible light. On the other hand, optical images can contain more information during the daytime compared to monochrome thermal images due to their color information. In addition, the environment may heat up during the day, resulting in lower contrast in thermal images, as human body temperature is almost constant. Overall, the use of both modalities seems to be symbiotic and to lead to better results compared to the use of a single modality. Using multiple modalities to train a model has led to state-of-the-art results in many cases. In particular, with the rise of transformers [Vas17], where inputs are transformed into homogeneous tokens, using multiple modalities such as text or images in a model has be-

come easier. In the area of monomodal crowd counting, the use of transformers has not been fully explored. To our best knowledge, with the exception of one work [Tia21], previous research has focused only on convolutional networks. The use of transformers has tremendous potential, as previous work [Zha16] [Li18] has often achieved better results when improving the extraction of multi-scale features. Although existing work [Liu21] [Pen20] concludes that the use of optical and thermal imagery leads to better crowd counting predictions, it is not yet fully understood how such models internally work and how they extract enriched features from both modalities.

Apart from the lack of understanding of how multimodal models work internally, it is not entirely understood whether the multimodal approach leads to better crowd counting results in general or only under certain conditions. Further research with potential influencing factors such as illumination, distance to the crowd, or number of people per image is needed to gain more certainty about whether multimodal crowd counting leads to better predictions in general. For this reason, in this paper we investigate the impact of using optical and thermal images simultaneously in crowd counting.

To investigate the impact of using optical and thermal images simultaneously in crowd counting, we designed a monomodal and several multimodal architectures consisting of the same key components. When we designed the monomodal model, we took into account the latest developments in the field of monomodal crowd counting. In addition, we have developed three multimodal models that incorporate different strategies of multimodal learning. To allow a comparison between the monomodal and the multimodal architectures, all key components of the monomodal architecture are also part of the multimodal architectures. The goal of this comparison is to find out whether multimodal models lead to better crowd counting results in general or only under certain conditions. Since this comparison led to interesting findings, we further analyzed all the datasets used to compare the models. To this end, we examined the relationship between the brightness of optical images and the number of individuals in the image. We also randomly selected a subset of each dataset and examined how individuals were labeled in the images from both modalities.

In examining the differences between the monomodal and the multimodal architectures, we found that existing datasets have a bias toward thermal images. This does not allow us to determine whether multimodal crowd counting leads to better results in general or only under certain conditions. For this reason, we have described criteria for a dataset suitable for investigating the research question.

2 RELATED WORK

Monomodal Crowd Counting: Crowd counting has been studied for decades. While a few works have used thermal images for crowd counting, most works have used optical images to examine crowd counting. As in other areas, the use of deep learning models [Wan15] [Fu15] has led to more accurate predictions in crowd counting. In recent years, the use of a density map-based approach for crowd counting has become prevalent. Many recent works have addressed the question of how to deal with scale variations in images. In particular, techniques such as multi-column models [Zha16] or dilated convolutions [Li18] have been used to extract multi-scale features from the image. Since such techniques aim to increase the receptive field of a network, it was no surprise that state-of-the-art results could be achieved by using a transformer encoder [Vas17] to extract features [Tia21].

Multimodal Crowd Counting: Multimodal learning is becoming increasingly relevant in the field of crowd counting. So far, the use of optical and thermal images [Liu21] [Pen20] [Tan22] [Gu22] as well as the use of optical and depth images [Lia22] [Lia19] has been investigated. However, depth images provide only a lim-

ited depth range (0 ~ 20 meters), making them unsuitable for many real-world crowd counting applications [Liu21]. Also, when using depth images, there is still the problem that less information is available in poorly illuminated scenes. For this reason, we will focus on the use of optical and thermal images in this paper. While all work concludes that the additional use of thermal images leads to better predictions in crowd counting, it is not fully understood under what circumstances it is beneficial to complement optical images with thermal images to obtain better predictions. Previous work has focused primarily on constructing a novel model architecture that outperforms the state-of-the-art in multimodal crowd counting. While this approach proves the effectiveness of the models created, it does not allow us to fully understand how complementary information is extracted from both modalities.

Multimodal Crowd Counting Datasets: Similar to different multimodal models, two different datasets [Liu21] [Pen20] consisting of optical and thermal image pairs have been published in recent years. The dataset published by Peng et al. [Pen20] was acquired with a drone and contains 3,600 image pairs. Furthermore, this dataset contains information about distance (scale of individuals), illumination and crowd count per image pair. The other dataset, which was published by Liu et al. [Liu21], contains 2,030 image pairs. The image pairs of this dataset were taken from a normal perspective. Information on the number of individuals and lighting is available for each image pair.

3 EFFECTIVENESS OF MULTIMODAL CROWD COUNTING

To allow a comparison between monomodal and multimodal architectures, we first developed a monomodal architecture. This monomodal model takes into account recent advances in the field of monomodal crowd counting and its main components are reused in subsequent multimodal architectures to allow a fair comparison. Since the constructed monomodal architecture is heavily inspired by recent advances in monomodal crowd counting and does not incorporate any new strategies, we only used one monomodal model for comparison.

3.1 Monomodal Architecture

The monomodal architecture designed in this work is inspired by the work of Tian et al. [Tia21] as well as the implementation of the work realized in [Wan21a]. The CCTrans model designed by Tian et al. [Tia21] achieves state-of-the-art results on multiple monomodal crowd counting benchmarks [Zha16] [Wan21b] [Idr18]. Our monomodal architecture is shown in Fig. 1. Instead of Twins [Chu21], which was used by Tian et al.

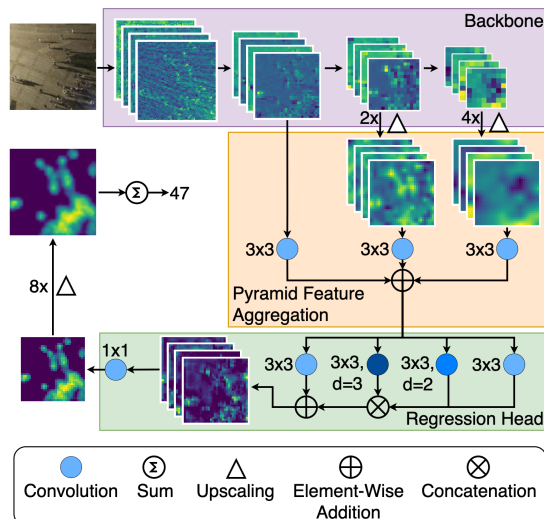


Figure 1: The architecture of our monomodal model, which is inspired by the work of Tian et al. [Tia21] as well as the implementation of the work realized in [Wan21a]. The input image is first transformed into tokens. From these tokens, features are extracted by a hierarchical transformer-based backbone. The hierarchical feature maps are then aggregated and finally used by the regression head to predict the crowd count. The parameter d indicates the dilation rate used.

[Tia21], we used PVTv2 [Wan21c] as the transformer-based backbone in our architecture. By empirical analysis, we found that the PVTv2 architecture leads to better results for us. More specifically, for our monomodal architecture, we used the PVTv2 B0 variant, which allows shorter training time and requires less computational resources. However, this leads to slightly worse results compared to other PVTv2 variants with more parameters. This was acceptable to us, as our primary goal was not to construct a novel architecture with state-of-the-art results. Furthermore, we adopted the pyramid feature aggregation and regression head of Tian et al. [Tia21], but used the convolution kernel sizes used in [Wan21a]. Again, through empirical analysis, we found that these kernel sizes led to slightly better results for us.

3.2 Multimodal Architectures

After the monomodal architecture was designed, three different multimodal architectures were designed that incorporate different strategies of multimodal learning. As mentioned before, the key characteristics of the monomodal architecture are also incorporated in the three different multimodal architectures. The idea behind this is that when using the same weight initialization (prior) and the same model properties (which constrain the hypothesis space), better results can only be explained by more information provided by the additional modality (data). In this work, we chose to use

early and late fusion as two simple multimodal strategies. These have also been used in previous work on multimodal crowd counting [Liu21] [Pen20]. In addition, we apply a more advanced deep fusion strategy using the Information Aggregation and Distribution Module (IADM) of Liu et al. [Liu21] which has been shown to be effective for multimodal crowd counting.

Early Fusion Model: With the early fusion strategy, modalities are fused at the beginning of the model. For this purpose, the constructed monomodal model was adapted to support 6-channel inputs by changing the amount of filters in the first layer. Thus, the multimodal early fusion model has the same number of parameters as the monomodal model.

Late Fusion Model: In contrast to the early fusion strategy, the fusion of modalities takes place at the end of the model with the late fusion strategy. The idea here is that features of both modalities are first extracted individually. Thus, except for the final layer (1×1 convolution), both modalities are investigated with the constructed monomodal model individually. Then, the extracted feature maps from both individual columns are concatenated. Based on the concatenated feature maps, a density map is then finally predicted by a 1×1 convolution. Hereby, the late fusion model requires around twice as many parameters as the monomodal model and the early fusion model.

Deep Fusion Model: In contrast to the early fusion and late fusion architectures, the multimodal information exchange in the deep fusion architecture takes place during feature extraction. For this purpose, a third column is added to the architecture, which extracts the complementary information of both modalities. In particular, this is done by using the IADM of Liu et al. [Liu21]. Through the IADM, information is exchanged between the modality-specific columns and the cross-modality column. However, this only takes place during feature extraction in the backbone, as shown in Fig. 2. Of all the models used in this work, this architecture requires the most parameters.

3.3 Evaluation

To evaluate the performance of the monomodal model and the three multimodal models, we used the mean absolute error (MAE) and the root mean squared error (RMSE). Both of these measures are widely used in crowd counting. The use of these measures allows comparison of our results with the results of other work. The mean absolute error and root mean squared error are defined as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (1)$$

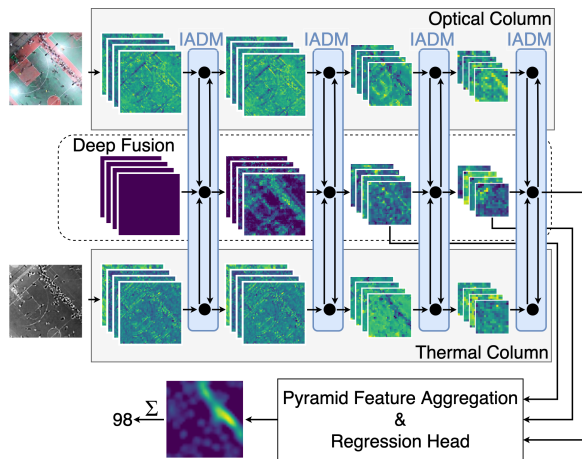


Figure 2: The architecture of our deep fusion model. To extract complementary information and enable exchange between modality-specific and modality-shared columns, we use the IADM of Liu et al. [Liu21]. In their work, it was shown that the use of the IADM is effective for multimodal data.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}, \quad (2)$$

where N is the number of image pairs, y_i is the ground-truth number of individuals in image pair i , and \hat{y}_i is the predicted number of individuals for image pair i .

3.4 Training

Overall, our training approach is heavily inspired by the training approach chosen by Tian et al. [Tia21]. The B0 variant of the PVTv2 architecture was initialized with pre-trained weights in all experiments. We used random cropping with a cropping size of 256 for both dimensions and horizontal flipping with a probability of 50% as augmentation strategies. In addition, AdamW [Los19] was used as the optimizer and a batch size of 8 was chosen for training. The learning rate was $1e-5$ in all experiments, but was increasingly regulated by a weight decay of $1e-4$. Bayesian loss [Ma19] with a sigma value of 8 was used as a loss function. The models were trained for 60 epochs in all experiments.

3.5 Results

The results for all constructed models on both datasets are shown in Tab. 1 and Tab. 2. Three aspects in particular caught our attention, which we describe in more detail below.

Discrepancy between optical and thermal images in both datasets. One of the first things we noticed is that the monomodal model performs much better on thermal images than on optical images, as can be seen in Tab. 1 and Tab. 2. This holds true for both datasets. Nevertheless, the discrepancy is larger for the RGBT-CC dataset

Modality	Architecture	MAE	RMSE
RGB	Monomodal	26.48	55.28
T	Monomodal	15.19	28.27
RGB-T	Early Fusion	14.92	25.86
RGB-T	Late Fusion	13.83	25.16
RGB-T	Deep Fusion	14.32	24.64
RGB-T	BL + IADM [Liu21]	15.61	28.18
RGB-T	TAFNet [Tan22]	12.38	22.45

Table 1: Performance of the different architectures on the RGBT-CC [Liu21] dataset. The use of thermal images leads to dramatically better results compared to optical images. Moreover, the multimodal approach leads to better results than the monomodal approach.

Modality	Architecture	MAE	RMSE
RGB	Monomodal	10.40	16.44
T	Monomodal	6.70	10.20
RGB-T	Early Fusion	7.41	11.43
RGB-T	Late Fusion	7.01	11.18
RGB-T	Deep Fusion	7.20	11.45
RGB-T	MMCCN [Pen20]	7.27	11.45
RGB-T	MFCC [Gu22]	7.96	12.50

Table 2: Performance of the different architectures on the Drone-RGBT [Pen20] dataset. Surprisingly, using thermal images solely with the monomodal architecture led to the best result for the Drone-RGBT dataset. In contrast, using optical images solely with the monomodal architecture leads to considerably worse results.

than for the Drone-RGBT dataset. Since we used the exact same model and training approach, these results raise the question of whether thermal images are more suitable for crowd counting in general. Before investigating this question, we first wanted to gain a better understanding of both data sets. The investigation is described in more detail in Section 4.

The monomodal model performs better than the multimodal models for the Drone-RGBT [Pen20] dataset. Contrary to our assumption that the multimodal approach of using optical and thermal images would lead to better crowd counting predictions, using thermal images solely led to the best result for the Drone-RGBT dataset. This result further affirmed our motivation to gain a better understanding of both datasets. To our best knowledge, we have achieved state-of-the-art results for the Drone-RGBT dataset using the monomodal architecture.

IADM [Liu21] seems to be less effective with transformer encoders. Comparing the three multimodal models, the late fusion model achieves the best results on both datasets. The deep fusion model, although more complex and shown to be effective by Liu et al. [Liu21], performs worse in our study than the late fusion model. Since Liu et al. also compared the IADM to a late fusion model, the most obvious explanation for

this is the use of a transformer encoder in our work. Liu et al. did not use a transformer encoder in their work. Nevertheless, a more detailed investigation beyond this work is needed to better understand why the IADM is less effective when used with transformer encoders.

4 ANALYSIS OF EXISTING MULTIMODAL CROWD COUNTING DATASETS

To understand more profoundly whether thermal images are better for crowd counting in general, or whether the characteristics of the datasets used lead to better results on thermal images, we used two different approaches.

4.1 Relationship Between Brightness and Crowd Count

First, we investigated the relationship between the brightness of optical images and the number of individuals. We suspected that many optical images in both datasets were taken in poorly illuminated environments, which could be the reason for the discrepancy between thermal and optical images. This would also be in line with our main motivation to use multimodal data. Since the two metrics we used consider the counting error and are sensitive to outliers, we thought it is relevant to investigate the relationship between brightness and crowd count. To measure the brightness of an optical image, we used the following equation:

$$\text{Brightness} = \frac{\sum_{i=1}^{W*H} R_i + G_i + B_i}{3 * W * H}, \quad (3)$$

where W is the width and H is the height of the optical image. R_i , G_i and B_i represent the three color values of pixel i . The relationship between brightness and crowd count for both datasets are shown in Fig. 3 and Fig. 4.

The RGBT-CC [Liu21] dataset is unbalanced regarding brightness and crowd count. The RGBT-CC dataset contains many images with very low brightness and high crowd count, as can be seen in Fig. 3. In comparison, the images in the Drone-RGBT dataset are much brighter on average and the overall distribution between brightness and number of individuals is much more balanced, as shown in Fig. 4. Since both metrics are sensitive to outliers and many optical images with very low brightness (low information) have a high crowd count in the RGBT-CC dataset, we assume that this explains the bigger discrepancy for the RGBT-CC dataset between optical and thermal images.

This finding has serious implications on our research question. Since this imbalance of the dataset likely affects all trained models and results in higher activations for thermal input, we believe that the research question cannot be thoroughly investigated with the RGBT-CC

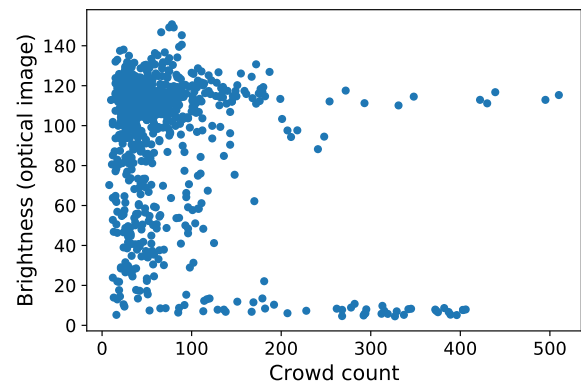


Figure 3: Scatter plot showing the relationship between the brightness of optical images and crowd count in the RGBT-CC dataset. It can be seen that the RGBT-CC dataset is unbalanced in terms of brightness and crowd count. Many images with very low brightness have a high crowd count.

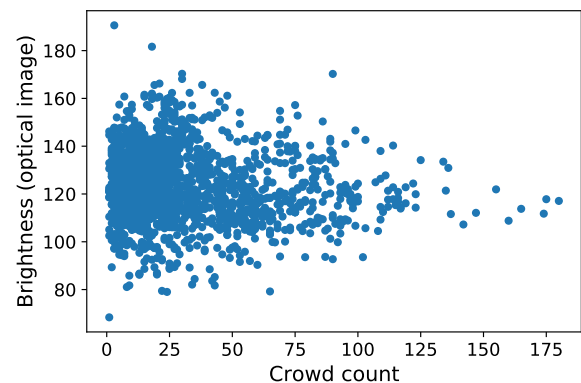


Figure 4: Scatter plot showing the relationship between the brightness of optical images and crowd count in the Drone-RGBT dataset. The relationship between brightness and crowd count appears very uniform compared to the distribution of the RGBT-CC dataset.

dataset. In particular, we assume that many optical images with low brightness (low optical information) and a high crowd count (high error) will cause the model to pay more attention to thermal images as the counting error is propagated back into the network during training. In this way, it is difficult to verify whether multimodal crowd counting leads to better results in general, since a certain condition (low brightness, high crowd count) has a great impact on the training of the model as well as the metrics. Nevertheless it is important to say that images with low brightness and high crowd count are not a problem, but are important and desirable for training a robust crowd counting model. However, we are concerned about whether our research question can be fairly investigated due to an inherent correlation between the number of people and brightness in the RGBT-CC dataset.

4.2 Annotation Sample Analysis

Our second approach to better understand both datasets was to perform a sample analysis of how the annotations were made. Since both datasets contain two different modalities recorded with two different cameras, we wanted to understand if images of both modalities were synchronized and how perspective changes were handled (because the cameras were probably next to each other during the recording). We decided to perform the sample analysis of how the annotations were made since both datasets provide shared annotations for both modalities. To this purpose, we randomly selected 10% of the image pairs per dataset and visualized the annotations in the images of both modalities to verify how the individuals were labeled in each image.

Only thermal images were used to label individuals in both datasets. By randomly selecting 10% of all image pairs per dataset and visualizing the annotations for both modalities, we found that both datasets used only the thermal image to label individuals. Examples of both datasets showing that only thermal images were used to label individuals are provided in the Appendix in Fig. 5 and Fig. 6.

All image pairs of the Drone-RGBT dataset were taken at night. We have seen that the optical images in the Drone-RGBT dataset are on average brighter than the optical images in the RGBT-CC dataset. However, by looking at the annotations for each image pair in the Drone-RGBT dataset, we perceived that all images were taken at night. To gain more confidence in this perception, we looked at all the optical images in the Drone-RGBT dataset. In this way, we found that all image pairs in the Drone-RGBT dataset were taken at night. Nevertheless, many images were taken in environments with much artificial light, therefore the optical images are on average brighter than those of the RGBT-CC dataset. The fact that all images were taken at night adds a new perspective to the results obtained with the Drone-RGBT dataset. This leads to the assumption that optical images do not provide additional information at night and that a monomodal approach with thermal images leads to better results. Further research beyond this paper is needed to validate this assumption.

For image pairs in the RGBT-CC dataset, individuals were sometimes visible in one modality but not the other. Liu et al. [Liu21] have already stated in their work that optical and thermal images in the RGBT-CC dataset are not strictly aligned because they were captured with different sensors. However, when examining the annotations, we found that not only were the image pairs not strictly aligned, but sometimes individuals were visible in one modality but not the other. Examples for this are provided in the Appendix in Fig. 7.

5 CRITERIA FOR A MULTIMODAL CROWD COUNTING DATASET

Because both datasets have some weaknesses that make it difficult to draw general conclusions about the effectiveness of multimodal crowd counting, we decided to set criteria for a suitable dataset. Overall, the image pairs should be taken evenly throughout the day. In this way, the variability of the two modalities gets extensively covered. Ideally, this would even take into account different seasons and climate zones. Also, the crowd count per image pair should be independent of when the image was taken. This allows for an equal influence of both modalities on the multimodal model during training, as no modality receives more attention due to a higher counting error. When labeling individuals, both modalities should be considered so that later models can learn to extract the information from both modalities and incorporate it into the prediction (even when one modality contains little information and the other contains much). Furthermore, the images for both modalities should be taken simultaneously. In this way, the images of both modalities are aligned as precisely as possible, which allows the use of the same annotations for both modalities.

6 IS MULTIMODAL CROWD COUNTING BETTER IN GENERAL?

The goal of this work was to find out if the simultaneous use of optical and thermal images leads to better predictions in crowd counting in general. We found that existing datasets have a bias toward thermal images, making it difficult to draw general conclusions about the effectiveness of multimodal crowd counting. The results on the Drone-RGBT dataset indicate that solely using thermal images at night results in better predictions than a multimodal approach. Since the RGBT-CC dataset contains both daytime and nighttime images, the better predictions with multimodal data seem to indicate that the multimodal approach leads to better results during the daytime. However, these assumptions are by no means proven, but could serve as hypotheses for future research. Furthermore, we encourage the creation of a multimodal dataset in order to be able to investigate such hypotheses. We have provided criteria for the creation of such a dataset in the previous Section 5. However, it remains an open question whether multimodal crowd counting (including technical challenges like perspective distortion and synchronization between modalities) is the perfect approach. It could also be the case that two monomodal models produce better results than one multimodal model. For example, one monomodal model could be used with optical images during the day and another with thermal images at night.

7 CONCLUSION

In this work, we found that existing multimodal crowd counting datasets have a bias toward thermal images. For this reason, we outlined criteria for a balanced dataset. To our best knowledge, we also obtained state-of-the-art results on the multimodal Drone-RGBT dataset. Interestingly, for this we used solely thermal images and the monomodal model constructed in this work. Considering the results of this work, we encourage the creation of a multimodal dataset that meets the criteria outlined in this paper. In this way, we can understand more profoundly whether the simultaneous use of optical images and thermal images leads to better predictions in crowd counting in general.

8 REFERENCES

- [Vas17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser and I. Polosukhin, “Attention is All you Need” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., vol. 30, pp. 6000–6010, 2017.
- [Tia21] Y. Tian, X. Chu and H. Wang (2021), “CCTrans: Simplifying and Improving Crowd Counting with Transformer” unpublished.
- [Zha16] Y. Zhang, D. Zhou, S. Chen, S. Gao and Y. Ma, “Single-Image Crowd Counting via Multi-Column Convolutional Neural Network” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 589-597.
- [Li18] Y. Li, X. Zhang and D. Chen, “CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1091-1100.
- [Liu21] L. Liu, J. Chen, H. Wu, G. Li, C. Li and L. Lin, “Cross-Modal Collaborative Representation Learning and a Large-Scale RGBT Benchmark for Crowd Counting” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4821-4831.
- [Pen20] T. Peng, Q. Li and P. Zhu, “RGB-T Crowd Counting from Drone: A Benchmark and MMCCN Network” in *Computer Vision - ACCV 2020: 15th Asian Conference on Computer Vision*, 2020, pp. 497–513.
- [Tan22] H. Tang, Y. Wang and L. Chau (2022), “TAFNet: A Three-Stream Adaptive Fusion Network for RGB-T Crowd Counting” unpublished.
- [Lia22] D. Lian, X. Chen, J. Li, W. Luo and S. Gao, “Locating and Counting Heads in Crowds With a Depth Prior” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, 2022, pp. 9056–9072.
- [Lia19] D. Lian, J. Li, J. Zheng, W. Luo and S. Gao, “Density Map Regression Guided Detection Network for RGB-D Crowd Counting and Localization” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1821–1830.
- [Wan21a] F. Wang and F. Taioli (2021) CCTrans: Simplifying and Improving Crowd Counting with Transformer (Code reproduction) [Source code]. <https://github.com/wfs123456/CCTrans>
- [Wan21b] Q. Wang, J. Gao, W. Lin and X. Li, “NWPU-Crowd: A Large-Scale Benchmark for Crowd Counting and Localization” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, 2021, pp. 2141–2149.
- [Idr18] H. Idrees et al., “Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds” in *European Conference on Computer Vision*, 2018, pp. 544–559.
- [Chu21] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia and C. Shen H., “Twins: Revisiting the Design of Spatial Attention in Vision Transformers” in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 9355–9366.
- [Wan21c] W. Wang, E. Xie, X. Li, D. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao (2021), “PVT v2: Improved Baselines with Pyramid Vision Transformer” unpublished.
- [Los19] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization” in *7th International Conference on Learning Representations*, 2019.
- [Ma19] Z. Ma, X. Wei, X. Hong and Y. Gong, “Bayesian Loss for Crowd Count Estimation With Point Supervision” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6141–6150.
- [Gu22] S. Gu and Z. Lian (2022), “A Unified Multi-Task Learning Framework of Real-Time Drone Supervision for Crowd Counting” unpublished.
- [Wan15] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, “Deep People Counting in Extremely Dense Crowds” in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1299–1302.
- [Fu15] M. Fu, P. Xu, X. Li, Q. Liu, M. Ye, and C. Zhu, “Fast crowd density estimation with convolutional neural networks” in *Engineering Applications of Artificial Intelligence*, vol. 43, 2015, pp. 81–88.

APPENDIX

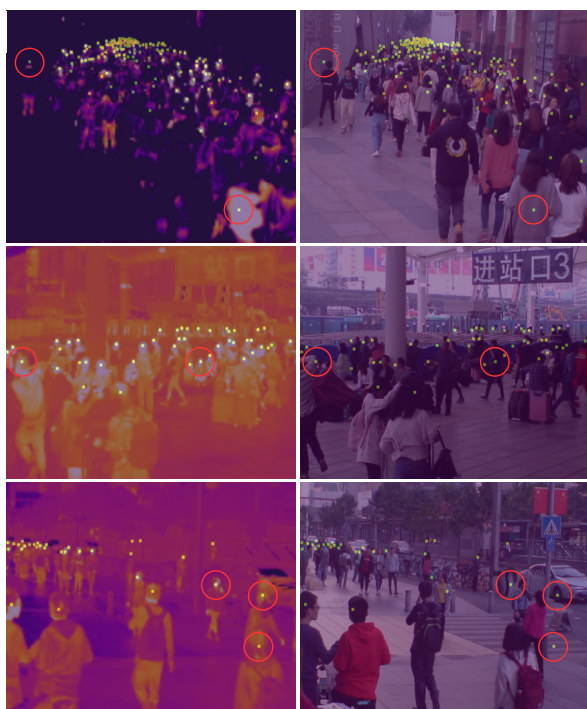


Figure 5: Illustration of the annotations (yellow squares) of the RGBT-CC [Liu21] dataset for both modalities. It can be seen that the annotations were created based on the thermal images. For better understanding, examples of inaccuracies in the annotation of the optical images have been encircled. The thermal images are also encircled in the corresponding places, but the annotations are correct there.



Figure 6: The annotations of the Drone-RGBT [Pen20] dataset are shown in yellow squares. It can be seen that the annotations were made on the basis of the thermal images. For better understanding, individuals have been encircled who are easily recognizable in the optical image, but have not been annotated. In the thermal image, on the other hand, it can be seen that precisely these individuals are difficult to recognize, which is probably why they were not annotated.



Figure 7: Two examples from the RGBT-CC dataset where a person is visible in one modality but not the other. It can be seen that this is due to the time-delayed capture of the images from both modalities.

Coordinate-Unet 3D for segmentation of lung parenchyma

Van-Linh Le^{1,2,3}
van-linh.le@u-bordeaux.fr

Olivier Saut^{1,2}
olivier.saut@inria.fr

¹ MONC team - INRIA Bordeaux Sud-Ouest, Talence-33400, France

² University of Bordeaux (IMB), CNRS and Bordeaux INP, UMR 5251, Talence-33400, France

³ BRIC Unit, University of Bordeaux, Talence-33400, France

ABSTRACT

Lung segmentation is an initial step to provide accurate lung parenchyma in many studies on lung diseases based on analyzing the Computed Tomography (CT) scan, especially in Non-Small Cell Lung Cancer (NSCLC) detection. In this work, Coordinate-UNet 3D, a model inspired by UNet, is proposed to improve the accuracy of lung segmentation in the CT scan. Like UNet, the proposed model consists of a contracting/encoder path to extract the high-level information and an expansive/decoder path to recover the features to provide the segmentation. However, we have considered modifying the structure inside each level of the model and using the Coordinate Convolutional layer as the final layer to provide the segmentation. This network was trained end-to-end by using a small set of CT scans of NSCLC patients. The experimental results show the proposed network can provide a highly accurate segmentation for the validation set with a Dice Coefficient index of 0.991, an F1 score of 0.976, and a Jaccard index (IOU) of 0.9535.

Keywords

Lung segmentation, NSCLC, Unet, Coordinate Convolutional, Deep learning

1 INTRODUCTION

Lung cancer is a major disease that accounts for more than one million deaths [33], and Non-Small Cell Lung Cancer (NSCLC) accounts for 85% of all lung cancers[11]. Early detection of lung cancer could reduce the mortality rate and increase the patient's survival rate during treatment operations. In most techniques to capture the image of the cancer patient, Computed Tomography (CT) scan is an effective medical screening that can be used for the diagnosis and detection of lung cancer. As CT acquisition represent a large volume of CT scans with millions of voxels, manual diagnosis and analysis of lung diseases is a difficult task and time-consuming even for experienced radiologists. Thus, automatic computer-aided diagnosis (CAD) for lung CT is a powerful solution to help radiologists. As usual, CAD involves several steps, of which accurate segmentation of the lung is the first step towards the success of the entire CAD system.

In order to response to clinician demands, lung segmentation algorithms have been proposed and have continued to emerge. However, obtaining accurate lung segmentation remains a challenge. In general, lung segmentation algorithms are grouped into two categories: (1) traditional image processing algorithms and (2) artificial intelligence (AI) based algorithms. In traditional image processing, algorithms consider image value, shape, and spatial information to provide lung segmentation. However, most of these methods are computationally expensive, difficult to create representative training features. In recent years, artificial intelligence-based approaches, for example, machine learning and deep learning, have become popular. These methods offer better accuracy under certain ill-defined conditions.

Over the past decade, deep learning methods have been at the forefront of computer vision [12, 7], showing significant improvements over previous methodologies on visual understanding. They are generally becoming a universal solution for image processing applications, with fewer pre-processing steps, but provide better results than other algorithms. Besides classification and object detection tasks [18, 27], segmentation is another success story of deep learning methods in computer vision. These methods (e.g., the FCN model [21] or Unet model [5]) mention labeling the pixels in the image into several categories. Generally, these models consist of two paths: an encoder path to extract and capture the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

image's features from a low to a high level and; a decoder path to expand and reconstruct the image features to provide the segmentation map.

In this work, we proposed another 3D architecture that was inspired by UNet model [5], named Coordinate-UNet, for accurate lung segmentation from a 3D volume of CT scan of a NSCLC patient. Our model had the same structure as the original Unet model [5]. However, we have modified the core inside each level of the encoder and the decoder paths. Our model achieved an accurate segmentation (a Dice score of 0.99) based on a small set of 3D images of NSCLC patients. The highlighted points of our method are summarized as follows:

- The proposed model inputs lung CT volume (3D) and outputs the segmentation of lung parenchyma without any post-processing operation.
- The proposed model can provide the results in a short time.
- The proposed model is effective for lung segmentation. Its performance is very stable. It can also be applied to detect nodules in the CT scan of the lung.

This paper is organized as follows: The following section discusses the related works on lung segmentation approaches. Section 3 presents the methodology which details the Coordinate-UNet 3D architecture and the studied datasets. The experimental results will be shown in section 4, and some concluding remarks will be stated in section 5.

2 RELATED WORKS

In normal lung CT scans, it is easy to distinguish the lung parenchyma and the non-lung region because the intensities of these regions are very different. However, this task becomes a difficult task when considering abnormal CT images such as the CT scans of NSCLC patients. Therefore, lung segmentation could be a challenge in many studies about lung diseases [8, 35, 26].

As discussed, initial approaches for lung segmentation consider the characteristics of the image or based on the shape knowledge by using the traditional image processing technique such as thresholding based on the grey-level, adaptive thresholding technique, region growing, image registration [4] or the others [34]. However, these methods are computationally expensive, and it is difficult to generalize the learning features.

In medical imaging applications, Convolutional Neural Networks (CNNs) models can be used to analyze lung CT scans [2]. Their applications vary from detecting the lung pathology segmentation [2], or classifying the lung region [2, 37] to segment the lung volume [31].

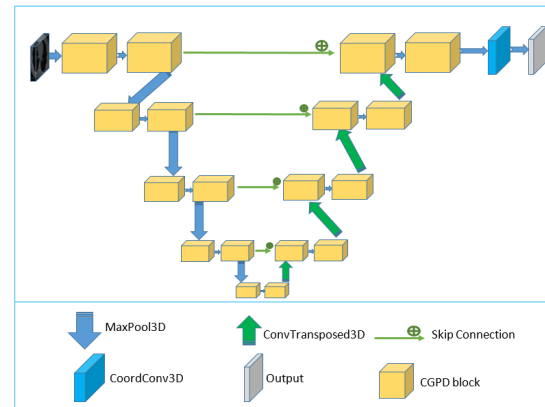


Figure 1: The architecture of the Coordinate-UNet 3D

In the approaches to provide whole volume of lung segmentation, most of applications have chosen to consider independently each slice of CT volume: Ravindra Patil et al. [28] and Brahim A. S. et al. [31] proposed different modifications on UNet model to provide the segmentation of slices in CT volume. Lei Geng et al. [10] presented a combination of VGG-16 [30] and dilated convolution to provide the lung segmentation. Swati P. Pawar et al. [29] have proposed a conditional generative adversarial network to encode the features from the slices of CT image. Then, the encoder and decoder were used to extract the multi-scales features and to give the lung segmentation, respectively. In whole 3D approach, Negahdar et al. [25] have proposed a volumetric segmentation network based on V-net [24] for 3D volumetric medical segmentation.

3 METHODOLOGY

In this section, we describe the architecture of the Coordinate-UNet 3D model and the hyper-parameters related to the training process. Then, the studied dataset used to train and validate Coordinate-UNet will be presented in detail.

3.1 Network architecture

In recent years, the UNet architecture [5] has been promoted as a good model for medical image segmentation. Generally, the UNet model [5] consists of two paths: a contraction path and an expansion path. The contraction path is like a classical CNN to extract low to high level features from an input representation. On the other hand, the expansive path consists of bottom-up sampling of the feature map followed by bottom-up convolution layers to reconstruct the features at different scales of the input. A concatenation is done between the convolution layer in the contraction path and the up-convolution of the expansion path to obtain more accurate labeling. At the end of the model, a convolution layer is used to map the features to the desired number of classes.

Depending on the requirements of the applications, the depth and layers at each level of the models are different. Figure 1 shows the structure of the Coordinate-UNet 3D model. Our model is inherited from the Unet model [5], but we have modified the UNet structure to adapt to our problem: (1) each level of contracting and expansive path contains two CGPD blocks (described in the next section); (2) the number of channels of the input is doubled before reducing the space to the next level; (3) a coordinate convolutional layer (CoordConv) [20] (instead of the classical convolution layer) is added at the end of the model to provide the segmentation map. Table 1 details the input/output images at each level of Coordinate-Unet 3D model.

Figure 2 illustrates the layers in a CGPD block. Each CGPD block consists of one Convolutional layer, one Group normalization (GN) layer [36], one PreLU activation function [13], and one Dropout layer [32]. Firstly, the GN is used instead of Batch normalization (BN) [14] because BN increases the error rapidly caused by inaccurate batch statistics estimation when the batch size becomes smaller; while GN divides the channels into groups and computes within each group the mean and variance for normalization. GN's computation is independent of batch sizes, and its accuracy is stable in a wide range of batch sizes. Second, the PreLU activation function [13] is used instead of ReLU activation function [23] to make the leakage coefficient a parameter that is learned along with the other network parameters. Finally, the Dropout layer is added to avoid over-fitting.

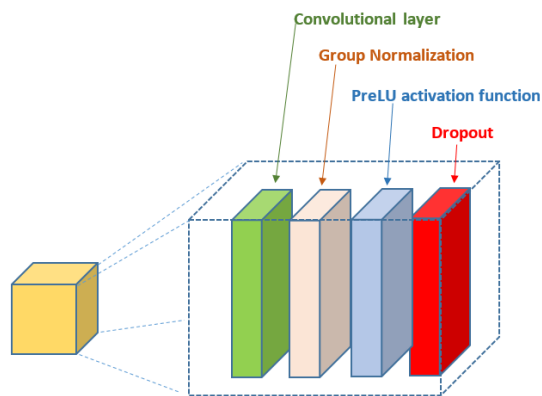


Figure 2: Layers in a CGPD block

As shown in the figure 1, each level of the contracting path includes two CGPD blocks, and the number of input channels is doubled at the second block before applying a max pooling layer to reduce the spatial size of the input and sending it to the next level of the model. At the expansive path, every level consists of an up-sampling layer that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, followed by two CGPD blocks to provide the information context of the

feature map. At the end of the expansive path, a Coord-Conv layer (figure 3) is used to map the features to the desired number of classes.

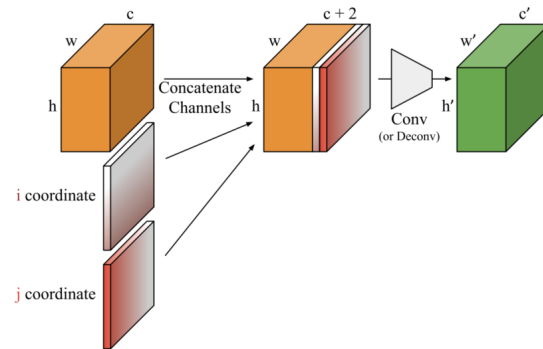


Figure 3: CoordConv layer [20]

As mentioned in [20], CoordConv is an extension of the standard convolution layer. It has the same functional signature as a convolution layer but accomplishes the mapping by first concatenating extra channels to the incoming representation. These channels contain hard-coded coordinates. The CoordConv layer keeps the properties of few parameters and efficient computation from convolutions but allows the network to learn to keep or to discard translation in variance as is needed for the task being learned. This is useful for coordinating transform based tasks where regular convolutions can fail. Because of the presentation of this layer, we give the name Coordinate-UNet 3D for our architecture.

3.2 Evaluation metrics

The evaluation of 3D lung parenchyma segmentation is based on the comparison between ground-truth and outputted segmentation from the model. For quantitative analysis of the predicted segmentation, several performance metrics are considered, including Dice score coefficient (DSC), F1-score, Jaccard similarity (IOU) and Matthews correlation coefficient (MCC) [3].

DSC is expressed as in Eq. 1 according to [6]. Here, GT and SP refer to the ground truth and predicted segmentation, respectively.

$$DSC = \frac{2 * (|GT \cap SP|)}{|GT| + |SP|} \quad (1)$$

The IOU score is represented using Eq. 2 according to [15].

$$IOU = \frac{(|GT \cap SP|)}{|GT \cup SP|} \quad (2)$$

The MCC score is computed according to Eq. 3.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (3)$$

where TP, FP, TN, and FN are True Positive, False Positive, True Negative, and False Negative rates, respectively.

Level	Input ($c \times w \times h \times d$)	Output ($c \times w \times h \times d$)	Level	Input ($c \times w \times h \times d$)	Output ($c \times w \times h \times d$)
Encoder path			Decoder path		
Input	-	$1 \times 256 \times 256 \times 48$	Output	$32 \times 256 \times 256 \times 48$	$1 \times 256 \times 256 \times 48$
Level 0	$2 \times 256 \times 256 \times 48$	$32 \times 256 \times 256 \times 48$	Level 0	$96 \times 256 \times 256 \times 48$	$32 \times 256 \times 256 \times 48$
Level 1	$32 \times 128 \times 128 \times 24$	$64 \times 128 \times 128 \times 24$	Level 1	$192 \times 256 \times 256 \times 48$	$64 \times 128 \times 128 \times 24$
Level 2	$64 \times 64 \times 64 \times 12$	$128 \times 64 \times 64 \times 12$	Level 2	$384 \times 64 \times 64 \times 12$	$128 \times 64 \times 64 \times 12$
Level 3	$128 \times 32 \times 32 \times 6$	$256 \times 32 \times 32 \times 6$	Level 3	$768 \times 32 \times 32 \times 6$	$256 \times 32 \times 32 \times 6$
Level 4	$256 \times 16 \times 16 \times 3$	$512 \times 16 \times 16 \times 3$	Level 4	$512 \times 16 \times 16 \times 3$	$512 \times 16 \times 16 \times 3$

Table 1: The dimensions of input/output features at each level of proposed model.

3.3 Experimental data

Coordinate-UNet 3D was trained and validated on 3D CT images of NSCLC patients. Images were obtained from 2 sources: NSCLC-Radiomics-Interobserver1 dataset [16] consisting of 21 CT scans, and a local dataset consisting of 66 CT scans. Each image has a variable size from $(512 \times 512 \times 60)$ to $(512 \times 512 \times 600)$ pixels. All images were reformatted from standard DICOM to Neuroimaging Informatics Technology Initiative (NIfTI) format created by the National Institutes of Health [19]. The NIfTI file held the 3D image matrix and diverse metadata. Figure 4 shows a 3D image for volumetric measurements of lung parenchyma across three axes.

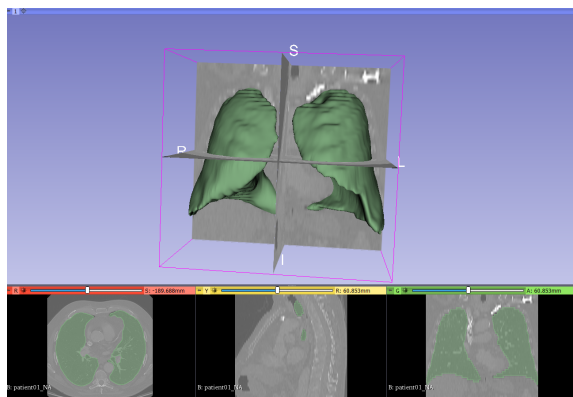


Figure 4: 3D volume of lung with XYZ axes.

Data splitting and pre-processing:

The images from 2 datasets were divided into two sets: train/validation and testing set. The train/validation set consists of 21 images from Interobserver1 dataset [16] and 41 images from the local dataset. The testing set consists of remaining 25 images of the local dataset. During the training process, the 62 images were divided into two sets corresponding to the training and validation process with a ratio of 0.8 : 0.2.

According the guideline of CT imaging, a CT scan consists of pixel spacing, axial slice thickness and view in the z axis with various scans. Therefore, the input images are uniformly pre-processed to minimize the vari-

ability within the database. In this work, the input images go through the three following steps for preparing the images:

1. The image intensity of each slice was first truncated in the range of $[-1200, 600]$.
2. Z-normalization was performed on each slice of 3D image.
3. The CT scans were cropped to focus on lung region and converted to $256 \times 256 \times 48$ pixels.

Data augmentation

Due to the limitation of the number of samples in the training set, we used online and offline augmentation operations to increase the number of images in the training set. Offline augmentation means that we generate multiple augmented versions from an original image and add them to the dataset, while online augmentation operations are performed during the training process. In our work, we applied warping and flipping operations to generate two new versions from one image. Figure 5 shows an augmented example in our dataset. After the offline augmentation, we obtained $62 \times 3 = 186$ images for the training and validation processes. In addition to offline augmentation, online augmentation was performed like other segmentation approaches such as spatial flipping and image shearing.

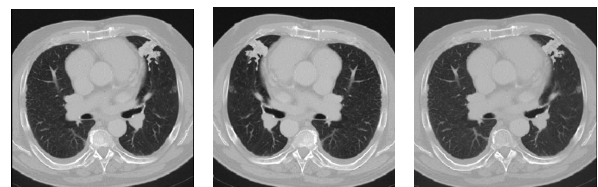


Figure 5: A slice in a 3D studied image with its augmentation. From left to right: original slice, flipped slice, and deformed slice.

4 EXPERIMENTS AND RESULTS

Coordinate-UNet 3D was implemented on Pytorch-lightning [9]. The model has been trained in 500 epochs with Dice loss ($\text{Dice loss} = 1 - \text{DSC}$). The

Adam optimization [17] has been used with an initial learning rate of 10^{-4} and reduce to 10^{-6} by using cosine annealing schedule [22]. An early stopping has been employed to monitor the validation loss to avoid overfitting.

The proposed model was firstly trained in one fold with 80% of data and validated on 20% of data. The objective was to find the best hyperparameters for our model. After adjusting the hyperparameters, we trained the model on 5-fold cross-validation to ensure the stability of the model. Then, we computed the average of the scores outputted from 5 folds. Table 2 shows the average scores obtained on the validation set by using different kinds of final layer: CoordConv (the second row) and the traditional Convolutional layer (the third row). We see that the model can provide a very accurate lung parenchyma segmentation with both two kinds of Convolutional layers. Moreover, the outputted scores of CoordConv are higher than the traditional Convolutional layer. It provides a high Dice score, the other scores (IOU, F1, MCC) are smaller than Dice but remain in a high accuracy rank.

Score	Dice index	IOU	F1	MCC
CoordConv	0.9907	0.9535	0.9761	0.9705
Convolution	0.9202	0.9383	0.9643	0.9236

Table 2: The performance of Coordinate-UNet model on validation set

These scores showed that the Coordinate-UNet 3D model has a good performance on the validation set. Then, the trained model was used to predict the segmentation of the testing images. The outputted segmentation was evaluated by calculating the dice score between prediction and the ground truth. Then, the average Dice score of 25 testing images was considered. We have obtained an average Dice score of 0.776 for all 25 images. Figure 6 and 7 illustrate the two examples in the testing set: one good and one worst predictions. From left to right, each one represents the ground truth and the segmentation generated by our network. The top row shows the segmentation of a slice in the image, the bottom row illustrates the 3D-segmentation.

Figure 8 shows the number of CT scans in several considered ranges of C-index scores. We see that our model worked effectively to provide good segmentation for most of the images in the test set (17 out of 25 CT scans obtain Dice scores greater than 0.75). Even in the image with the worst Dice score (figure 7), we see the segmentation of the lung parenchyma is correct, with the predicted segmentation producing only a minor error outside the lung region. This area really affected the dice score, but it was not that significant. Indeed, we want to see if the model can provide the segmentation of the lung parenchyma.

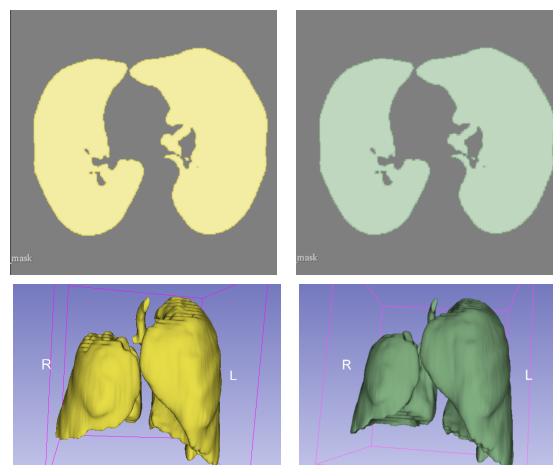


Figure 6: A good prediction in the test set (yellow = ground truth, green = prediction). Top: Segmentation of a slice in 3D image. Bottom: Lung parenchyma segmentation in 3D

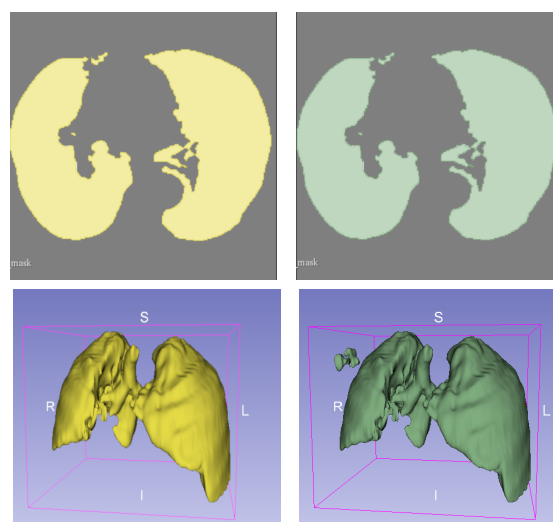


Figure 7: A worst prediction in the test set (yellow = ground truth, green = prediction). Top: Segmentation of a slice in 3D image. Bottom: Lung parenchyma segmentation in 3D

The performance of the model on the testing dataset is good, but this is far from the results on the validation set. The hypothesis is the difference between the training and testing datasets. It is worth noting that the model was trained on a combination of the CT scans from two datasets. After checking the data, we see that 21 images from the Interobserver1 dataset [16] do not have the bronchi, while the bronchi are present in the local dataset.

It is not fair to compare our results with other results because we are on different approaches [28, 31, 10]. However, Coordinate-UNet 3D model can segment lung parenchyma with very satisfactory performance and have the potential to locate and analyze lung lesions. It was employed to segment the lung parenchyma in other

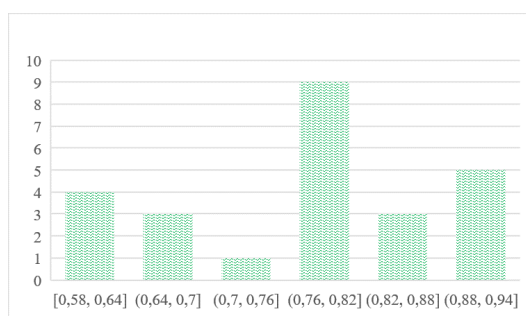


Figure 8: The number of predictions in each range of Dice score

datasets: NSCLC-Radiomics [1], which were used in another work to provide the segmentation of the tumor inside the lung.

5 CONCLUSION

In this work, we presented Coordinate-Unet 3D model to provide the lung segmentation. We have obtained an accurate segmentation with 0.99 Dice coefficient index for the validation set. However, the average Dice score on the test set has decreased a little bit. The problem has been found as a difference between the training and testing datasets. Basically, the prediction error is not so high, it can be solved by applying an algorithm to remove the small object in the segmentation map. The advantage of the method is the fact that can work with a whole 3D volume of the CT image and it can be applied to a wide area of different medical image segmentation task. Our objective is to generalize the approach to apply it to another task, for example, to perform lung tumor segmentation. This work has opened some questions that we can address in future work. (1) Concerning the data, we tried to reduce the bias between the images in the datasets, as well as limit the loss of data during the normalization process. (2) Analyze the effect of the presence of bronchi in the CT images by considering two groups of input images with and without the presence of bronchi. (3) This is an intermediate stage in our pipeline which provides the segmentation of NSCLC tumors, it is interesting to compare the results of the whole pipeline with other methods.

6 ACKNOWLEDGMENTS

This work was supported by the Fondation MSDAvenir and Fondation Inria for the Pimiento project. The experiments presented in this paper were carried out using the PlaFRIM platform¹.

7 REFERENCES

- [1] Hugo JWL Aerts, Emmanuel Rios Velazquez, Ralph TH Leijenaar, Chintan Parmar, Patrick

Grossmann, Sara Carvalho, Johan Bussink, René Monshouwer, Benjamin Haibe-Kains, Derek Ritveld, et al. Decoding tumour phenotype by non-invasive imaging using a quantitative radiomics approach. *Nature communications*, 5(1):4006, 2014.

- [2] A Asuntha and Andy Srinivasan. Deep learning for lung cancer detection and classification. *Multimedia Tools and Applications*, 79:7731–7762, 2020.
- [3] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.
- [4] Kenneth R Castleman. *Digital image processing*. Prentice Hall Press, 1996.
- [5] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2016: 19th International Conference, Athens, Greece, October 17–21, 2016, Proceedings, Part II 19*, pages 424–432. Springer, 2016.
- [6] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
- [7] Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric Topol, Jeff Dean, and Richard Socher. Deep learning-enabled medical computer vision. *NPJ digital medicine*, 4(1):5, 2021.
- [8] David S Ettinger, Wallace Akerley, Gerold Bepler, Matthew G Blum, Andrew Chang, Richard T Cheney, Lucian R Chirieac, Thomas A D’Amico, Todd L Demmy, Apar Kishor P Ganti, et al. Non-small cell lung cancer. *Journal of the national comprehensive cancer network*, 8(7):740–801, 2010.
- [9] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019.
- [10] Lei Geng, Siqi Zhang, Jun Tong, and Zhitao Xiao. Lung segmentation method with dilated convolution based on vgg-16 network. *Computer Assisted Surgery*, 24(sup2):27–33, 2019.
- [11] Peter Goldstraw, David Ball, James R Jett, Thierry Le Chevalier, Eric Lim, Andrew G Nicholson, and Frances A Shepherd. Non-small-cell lung cancer. *The Lancet*, 378(9804):1727–1740, 2011.
- [12] Mahmoud Hassaballah and Ali Ismail Awad. *Deep learning in computer vision: principles and applications*. CRC Press, 2020.

¹ <https://www.plafrim.fr>

- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.
- [15] Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.
- [16] Petros Kalendralis et al. Fair-compliant clinical, radiomics and dicom metadata of rider, interobserver, lung1 and head-neck1 tcia collections. *Medical Physics*, 47(11):5931–5940, 2020.
- [17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.
- [19] Michele Larobina and Loredana Murino. Medical image file formats. *Journal of digital imaging*, 27:200–206, 2014.
- [20] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *Advances in neural information processing systems*, 31, 2018.
- [21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [22] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
- [23] Andreas Maier, Christopher Syben, Tobias Lasser, and Christian Riess. A gentle introduction to deep learning in medical image processing. *Zeitschrift für Medizinische Physik*, 29(2):86–101, 2019.
- [24] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571, 2016.
- [25] Mohammadreza Negahdar, David Beymer, and Tanveer Syeda-Mahmood. Automated volumetric lung segmentation of thoracic ct images using fully convolutional neural network. In *Medical Imaging 2018: Computer-Aided Diagnosis*, volume 10575, pages 356–361. SPIE, 2018.
- [26] Elise Noel-Savina and Renaud Descourt. Focus on treatment of lung carcinoid tumor. *OncoTargets and therapy*, pages 1533–1537, 2013.
- [27] Ajeet Ram Pathak, Manjusha Pandey, and Siddharth Rautaray. Application of deep learning for object detection. *Procedia computer science*, 132:1706–1717, 2018.
- [28] Ravindra Patil, Leonard Wee, and Andre Dekker. Auto segmentation of lung in non-small cell lung cancer using deep convolution neural network. In *Advances in Computing and Data Sciences: 4th International Conference, ICACDS 2020, Valletta, Malta, April 24–25, 2020, Revised Selected Papers 4*, pages 340–351. Springer, 2020.
- [29] Swati P Pawar and Sanjay N Talbar. Lungseg-net: Lung field segmentation using generative adversarial network. *Biomedical Signal Processing and Control*, 64:102296, 2021.
- [30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [31] Brahim Ait Skourt, Abdelhamid El Hassani, and Aicha Majda. Lung ct image segmentation using deep neural networks. *Procedia Computer Science*, 127:109–113, 2018.
- [32] Nitish Srivastava et al. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [33] Hyuna Sung et al. Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 71(3):209–249, 2021.
- [34] Selin Uzelaltinbulat and Buse Ugur. Lung tumor segmentation algorithm. *Procedia computer science*, 120:140–147, 2017.
- [35] Jan P Van Meerbeeck, Dean A Fennell, and Dirk KM De Ruysscher. Small-cell lung cancer. *The Lancet*, 378(9804):1741–1755, 2011.
- [36] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
- [37] Aleksandr Zotin, Yousif Hamad, Konstantin Simonov, and Mikhail Kurako. Lung boundary detection for chest x-ray images classification based on glcm and probabilistic neural networks. *Procedia Computer Science*, 159:1439–1448, 2019.

Sex Classification of Face Images using Embedded Prototype Subspace Classifiers

Anders Hast¹

¹ Department of Information
Technology
Uppsala University
Centre for Image Analysis
SE-751 05 Uppsala, Sweden
anders.hast@it.uu.se

ABSTRACT

In recent academic literature Sex and Gender have both become synonyms, even though distinct definitions do exist. This give rise to the question, which of those two are actually face image classifiers identifying? It will be argued and explained why CNN based classifiers will generally identify gender, while feeding face recognition feature vectors into a neural network, will tend to verify sex rather than gender. It is shown for the first time how state of the art Sex Classification can be performed using Embedded Prototype Subspace Classifiers (EPSC) and also how the projection depth can be learned efficiently. The automatic Gender classification, which is produced by the *InsightFace* project, is used as a baseline and compared to the results given by the EPSC, which takes the feature vectors produced by *InsightFace* as input. It turns out that the depth of projection needed is much larger for these face feature vectors than for an example classifying on MNIST or similar. Therefore, one important contribution is a simple method to determine the optimal depth for any kind of data. Furthermore, it is shown how the weights in the final layer can be set in order to make the choice of depth stable and independent of the kind of learning data. The resulting EPSC is extremely light weight and yet very accurate, reaching over 98% accuracy for several datasets.

Keywords

Sex and Gender Classification, Subspaces, Embedded Prototype Subspace Classification, Face Recognition.

1 INTRODUCTION

Sex or Gender classification is one important task in the field of face recognition (FR) and it will be shown how this can be done efficiently using *Embedded Prototype Subspace Classification* (EPSC) [Has22, HV21, HLV19, HL20, HV21], which has already been proven to be able to classify datasets of various kinds, such as digits, words and objects.

Another important contribution to EPSC in this paper, is to show how the projection depth can be learned and how the weights in the final layer can be set in order to make sure that the algorithm is stable and that the results are always reliable, regardless of how the depth parameter is set for unknown data to be classified.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

1.1 Sex or Gender?

Sex and Gender has increasingly become synonyms and the recent research papers about how to use FR to determine whether there is a male or a female in the image, generally use the word *Gender*, just as many other academic papers tend to do in their titles [Hai04] nowadays. However, there is a distinct difference between the words *Sex* and *Gender*, and the Council of Europe gives several definitions on one of their websites [oE23]. To make a long story short, *Sex* generally refers to "biological differences between males and females", while *Gender* is presented as a "social, psychological and cultural construct".

In this paper the question about sex and gender will be handled in a very simple way, by just using the labels "Male" and "Female" provided in the datasets. Hence, the classification has already been done in some way, and then the task is for any computer algorithm to determine the class based on cues such as face geometry, facial hair etc, as explained in the following sections. The main question is which cues to use and whether humans use other cues than Machine Learning (ML) algorithms for sex and gender classification. It will be argued that when using face feature vectors (FFV) from

FR, they are predominantly based on the geometry of the face, regardless of facial expression and outer attributes such as facial hair, makeup or hair-cuts, which otherwise could reveal the gender of the person at hand.

1.2 Cues

Hoss et al. [HRGL05] found that high masculinity in male faces, but not high femininity in female faces, facilitated sex classification when showing face images to both adults and children. And they also found that, independently of masculinity/femininity, attractiveness affected not only the accuracy of the sex classification, but also the speed.

Interestingly, Bruce et al. [BBH*93] found that cues from features such as eyebrows, and skin texture, play a more important role when humans are deciding whether faces are male or female, than cues from such things as hairstyle, makeup, and facial hair. Moreover, O'Toole et al. [OPD96] found that female observers were more accurate at classification of sex than were male observers, on both Caucasian and oriental faces.

Obviously, humans use different cues to determine sex and gender, both outer attributes but also the geometry itself, which is related to masculinity and femininity.

1.3 Geometry or Attributes?

The main question then is, which of those two concepts will FR algorithms pick up on? A Convolutional Neural Network (CNN) will probably pick up on both facial attributes, such as hairstyle, facial hair but also on geometrical aspects such as size of chin and nose, etc. The reason is simply, because they are all visual cues present to different degrees in face images.

It is quite obvious that *Sex* could have an impact on all these cues, while *Gender*, i.e. a persons own perception about gender, cannot change the geometrical aspects, since they are predominantly the results of a persons sex. However, depending on a persons *Gender* they have indeed power to change the other attributes, such as facial hair, makeup and haircut, which the surrounding world would perceive as typically female or male (or neither).

FFVs on the other hand are constructed so that the FR software can be used to recognise a person, regardless of the aforementioned outer attributes, which might reveal the persons gender. Hence, the FFV are more stable in that sense, and therefore obviously tend to pick up on the geometry, which is more related to *Sex*. So for this reason, the title contains the word *Sex Classification*, rather than *Gender Classification*. Nonetheless, as stated before, most researchers will understand both as determining whether the person present in the image is a male or a female.

2 RELATED WORK

Burton et al. [BBD93] compared the human ability to determine sex of persons showing the face only, wearing swimming caps to conceal their hair (outer attribute), compared to a computer approach based on the face geometry, and found that humans could reach an accuracy of 96% and that the computer was at the time approaching human performance of 94% accuracy. These experiments from 30 years ago underlines the importance of face geometry for determining sex.

Golumb et al. [GLS90] devised "SexNet", which was a neural network, working on aligned and scaled face images, and had an average error rate of 8.1% compared to humans, who averaged 11.6% on that particular dataset. Hence, this network would pick up on both the geometry and outer facial cues like facial hair.

Mäkinen and Raisamo [MR08] gives an overview of different gender classification methods with a varying accuracy between 76.87% and 86.54%, on the FERET [PWH98] and IMM [NLSS04] datasets.

Gong et al. [GLJ20] proposed a group adaptive classifier for face recognition, which is designed to customize the classifier for each demographic group, and automatically learns where to use adaptive kernels in a multilayer CNN. They obtained an accuracy for FR of 98.19% using a 5-fold cross-validation on 8 groups of the Racial Faces in the Wild (RFW) dataset [WDH*19]. However, the accuracy for gender classification was only 85%.

Gil and Hassner [LH15] achieve an accuracy of 86.8% using deep CNN's on the Adience Benchmark [EEH14], on which others have reached up to 91% [SBLM17].

Acien et al. [AMVR*19] used the Labeled Faces in the Wild (LFW) dataset [HRBLM07], which will be used also in this paper. They achieved a 94.8% accuracy using VGGFace and 89.01% using ResNet50, where a separate layer was added in both networks to classify gender. It can be noted that the dataset is ethnically mixed between Caucasians, Asians and Blacks.

Sumi et al. [SHIA21] gives an overview of several other works for gender classification, where no algorithm reaches over 97% accuracy. They report themselves an average training and test accuracy of 90% and 83.5%, respectively on the Nottingham Scan Database, which will also be used in this paper.

Both Liu et al. [LLWT15] and Ranjan et al. [RPC16] report results from several implementations trying to classify gender on the CelebA dataset, which is another dataset that will be used in this paper. The accuracy spans from 90% up to 98% depending on which method is being used.

2.1 3D Faces

Interestingly, Abbas et al. [AHM*18] proposed a set of facial morphological descriptors, based on 3D geodesic path curvatures between two key landmarks in 3D faces, obtained from 3D scanning. Hence, it would only pick up on geometry and not on outer facial cues, like facial hair. They achieved a gender classification accuracy of 88.6% using a combination of geodesic distances between landmarks and the new geodesic path features.

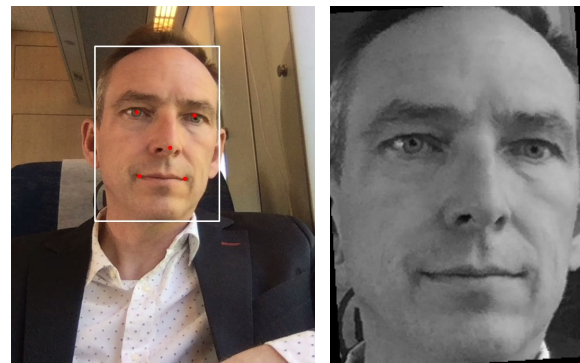
Gilani et al. [GRS*14] investigated how the human cognitive system uses geometric features in perceiving the degree of masculinity/femininity in 3D faces. Their results suggested that humans use a combination of both Euclidean and geodesic distances between biologically significant landmarks of the face for determining gender. Gilani and Milan [GM14] used this approach on 3D scanned faces and obtained an accuracy of 99.32

It should be noted though that the proposed method can not be easily compared to 3D methods, since only 2D face images are used from FR software. However, the important lesson here is that geometry was concluded in both mentioned papers as an important cue for differing between masculine and feminine faces.

3 BACKGROUND

The main advantage of EPSC compared to many deep learning based methods [Sha18] is that EPSC is shallow to its nature, with no hidden layers, and therefore it does not require powerful GPU resources in the training process. Recently, the main idea of backpropagation used by most neural network based approaches, was questioned by Geoffrey Hinton [Hin22] who proposed another alternative in the *The Forward-Forward Algorithm*. Interestingly, the EPSC does not use backpropagation at all and learns only from the feature vectors using PCA, and from the embedding of feature vectors, using dimensionality reduction techniques like t-SNE, UMAP or SOM [HV21]. Consequently, subspaces are created from each cluster [KLR*77, OK83], which constitutes a set of neurons that are specialised on identifying the class variation captured in that cluster. However, in this paper, only one set of neurons, i.e. subspace, will be used for each class, as it turns out to be more efficient for the purpose of sex/gender classification.

Obviously, EPSC does not always outperform the state-of-the-art deep learning approaches when it comes to accuracy. However, Both learning and inference will generally be much faster, due to its simplicity and compactness. Moreover, both the learning and classification processes are inherently easy to both interpret [Kri19, CPC19] and explain [ADRS*19, GSC*19, CPC19], as well as they are easy



(a) Detected face (white box) with landmarks (red dots) (b) Face after alignment

Figure 1: Illustration of automatic face detection and alignment using landmarks.

to visualise. Furthermore, Deep learning does not always solve a problem better than classical machine learning algorithms [DDD*23]. Hence, there are several reasons to look at fast, and sustainable computing alternatives.

4 FACE FEATURE VECTOR EXTRACTION

In this paper the *InsightFace* [Ins23] pipeline is used, which is an integrated Python library for 2D and 3D face analysis. *InsightFace* efficiently implements a rich variety of state of the art algorithms for both face detection, face alignment and face recognition, such as *RetinaFace*. [DGZ*19]. It allows for automatic extraction of highly discriminative FFVs for each face, based on the Additive Angular Margin Loss (ArcFace) approach [DGXZ19].

Face detection algorithms have come a long way since the simple, but efficient Viola-Jones detector [VJ01]. As an example, Figure 1a shows *RetinaFace* can perform automatic face detection that find the white bounding box of the face. Red landmarks are computed that can be used to align the face as shown in 1b.

The FFV, also known as embedding, will have length 512 when using *InsightFace* and the provided *buffalo_l* model, which is the default model pack in latest version of *InsightFace*. Other approaches such as *DeepFace* [TYRW14], *CosFace* [WWZ*18] *FaceNet* [SKP15], *SphereFace* [LWY*17], or [WZLQ16, SJ19], just to mention a few, could also have been used to produce FFVs for the proposed approach, as well as some of the ones mentioned in section 2.

5 DATASETS

In the automatic FR process, images where faces were not properly recognised or images with more than one face were removed. The remaining *face images*, i.e. an image containing one face and producing an FFV

to be processed further, were kept as shown in table 1. This process is referred to as the automatic selection in the following description of the datasets. For some datasets, it was required that several face images of the same person, covering several ages (decades) where chosen, while for others no such selection was done. In this way, quite different datasets are at hand and can be evaluated. Since the datasets used often are subsets of the original dataset, they will be referred to as explained below.

5.1 AgeDB

The *AgeDB* dataset [MPS*17] contains 16,516 images. Of those, 9826 face images were extracted so that each person depicted had about 36 face images on average covering at least three different age decades. Moreover, it was required that each person included should have at least 30 face images. This will ensure that there are several face images of the same person at different ages, or decades, and not only for one. Hence, it will make it possible to verify that the sex classification works for different ages.

5.2 CASIA

Since the *CASIA-WebFace* [YLLL14] is rather large as it contains around 500k images, a much smaller subset was extracted containing 65579 face images, which is still quite large compared to some of the other datasets used. Nevertheless, a similar approach was used as for AgeDB, resulting in more than 50 face images per person on average.

5.3 LFW

The *Labeled Faces in the Wild* (LFW) dataset [HRBLM07, LM14] contains 13,233 images, with 5749 different individuals. It is unbalanced as 1680 people have two or more images and the remaining 4069 have just a single image in the database. After FR, 10792 face images were selected automatically. No further selection was done, i.e. no requirements of age groups.

5.4 CelebA

The *CelebA* dataset [LLWT15] contains 202,599 images of 10,177 persons. The automatic FR extraction resulted in 200,096 face images where kept. No further selection was done.

5.5 UTKFace

The *UTKFace* dataset [ZYQ17] contains 23,709 images, where 23,685 face images were kept after the automatic FR extraction. This set contains a wide age range from 0 to 116 years old, making it rather challenging for sex classification. Moreover, there are quite many images with watermarks, that could influence the face recognition. Anyhow, no further selection was done.

Table 1: Databases used, with the total number of face images, number of women and Men.

Database	Total	Women	Men
AgeDB	9826	4071	5755
CASIA	65579	24077	41502
LFW	10792	2410	8382
CelebA	200096	116985	83111
UTKFace	23685	11308	12377
NSD	100	50	50

5.6 NSD

The *Nottingham Scans Dataset* (NSD) was included as it is rather different from the other dataset, but also because it has been evaluated before [SHIA21]. It only contains 100 face images, all of different persons. However, it is totally gender balanced, with 50 face images of each sex. As can be seen from the table, the other datasets vary quite a lot when it comes to this aspect, which will affect the accuracy, both when it comes to training as well as classification.

6 SUBSPACE CLASSIFICATION

Subspace Classification in pattern recognition was introduced by Watanabe et al. [WP73] in 1967 and was later further developed by Kohonen and others [WLK*67, KLR*77, KO76, KRMV76, OK88]. The following mathematical derivation follows from Oja and Kohonen [OK88] and Laaksonen [Laa07].

Every face image to be classified is represented by a FFV \mathbf{x} with m real-valued elements $\mathbf{x}_j = \{x_1, x_2, \dots, x_m\}, \in \mathbb{R}$, such that the operations take place in a m -dimensional vector space \mathbb{R}^m . In this paper m is equal to the FFV length, i.e. 512. Any set of n linearly independent basis vectors $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$, where $\mathbf{u}_i = \{w_{1,i}, w_{2,i}, \dots, w_{m,i}\}, w_{i,j} \in \mathbb{R}$, which can be combined into an $m \times n$ matrix $\mathbf{U} \in \mathbb{R}^{m \times n}$, span a subspace \mathcal{L}_U

$$\mathcal{L}_U = \{\hat{\mathbf{x}} | \hat{\mathbf{x}} = \sum_{i=1}^n \rho_i \mathbf{u}_i, \rho_i \in \mathbb{R}\} \quad (1)$$

where,

$$\rho_i = \mathbf{x}^T \mathbf{u}_i = \sum_{j=1}^m x_j w_{i,j} \quad (2)$$

Classification of a feature vector can be performed by projecting \mathbf{x} onto each subspace \mathcal{L}_{U_k} . The vector $\hat{\mathbf{x}}$ will in this way be a reconstruction of \mathbf{x} , using n vectors in the subspace through

$$\hat{\mathbf{x}} = \sum_{i=1}^n (\mathbf{x}^T \mathbf{u}_i) \mathbf{u}_i \quad (3)$$

$$= \sum_{i=1}^n \rho_i \mathbf{u}_i \quad (4)$$

$$= \mathbf{U}^T \mathbf{U} \mathbf{x}^T \quad (5)$$

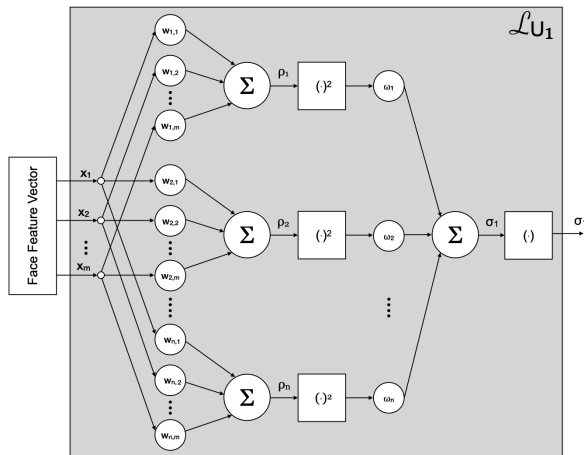


Figure 2: Illustration of the neural net that constitutes one of two subspaces forming the EPSC for Sex Classification. Neurons compute the response ρ from weights w , using the quadratic response function $(\cdot)^2$. In the second layer, the weights ω and the linear response function (\cdot) are used.

By normalising all the vectors in \mathbf{U} , the norm of the projected vector can be simplified as

$$\|\hat{\mathbf{x}}\|^2 = (\mathbf{U}\mathbf{x}^T) \cdot (\mathbf{U}\mathbf{x}^T) \quad (6)$$

$$= (\mathbf{U}\mathbf{x}^T)^2 \quad (7)$$

$$= \sum_{i=1}^n \rho_i^2 \quad (8)$$

Therefore, the feature vector \mathbf{x} , which is most similar to the feature vectors that were used to construct the subspace in question \mathcal{L}_{U_k} , will thereby have the largest norm $\|\hat{\mathbf{x}}\|^2$.

The parameter n discussed above is the projection depth that will be learned, which is discussed in section 7.

In order to construct subspaces, a group of prototypes need to be chosen for each subspace, which is done by the embedding obtained from some dimensionality reduction method such as t-SNE [MH08], UMAP [MH18] or SOM [Koh82]. However, for classification of sex, it was found that one subspace per class was sufficient, and hence all feature vectors for males are used to construct one subspace, and all feature vectors for females are used for the second subspace.

In general, subspace classification can be regarded as a two layer neural network [HLV19, OK88, Laa07], where the weights are not learned using time consuming backpropagation. Instead, all weights are mathematically defined through Principal Component Analysis (PCA) [Laa07]. The resulting Neural Network for one subspace in the EPSC is shown in Fig. 2. The output of the two subspaces are then compared via the *argmax* function to determine which class, male or female, is at hand.

The neurons compute the response ρ using the quadratic response function $(\cdot)^2$, commonly referred to as an activation function. The mathematics of subspaces defines it to be quadratic, since it is deduced from computing the norm as in equation (8). The weights ω in the final layer should, according to the definition of the dot product, all be set to 1. However, it will be shown how it can be changed to make the classification more stable. For the same reason, the response function in the final layer is by definition linear, which still makes sense because it is the output to *argmax*.

7 LEARNING THE PROJECTION DEPTH

As stated earlier, the variable n in equation (1) is the projection depth. It tells how many dimensions in the subspace that are actually used when computing the projected vector $\hat{\mathbf{x}}$ in equation (3).

The projection depth needs to be determined for each type of classification task, and will somehow depend on number of classes and the data at hand. As an example, it was reported to be 28 for MNIST [HLV19] and 6 for the Esposalles word dataset [Has22]. Experimentally it was noted that rather large values for n was necessary for sex classification, and it depended quite a lot on which of the aforementioned datasets were used for training. The obvious way to learn the depth, is to vary it from 1 to m and find the optimal accuracy. Here $m = 512$ for the face feature vectors obtained from *In-sightFace*.

Since the projection into the subspace can be regarded as a reconstruction of the input vector using the vectors in the subspace, it can be generally be observed that using a few vectors, i.e. small depth, gives more errors in the subspace reconstruction. On the other hand, using too many vectors, i.e. large depth, help in generating a *near perfect reconstruction*, making it impossible to suggest which subspace gives the strongest response.

Therefore, it is reasonable to deduce that the initial vectors in the PCA are more important than the subsequent ones, and therefore different weights ω could be applied. Referring to Fig. 2, let σ be the response output function from every subspace projection \mathcal{L}_U , then a closed-form solution of each σ is computed as a weighted sum, presented in the following equation:

$$\omega_i = 1 - \left(\frac{i-1}{n} \right)^2 \quad (9)$$

where i varies from 1 to n . This decaying or dampening function generated the best results in general. The quadratic decay makes sure that the initial dimensions will use an ω closer to one, while the very last dimensions will use an ω closer to zero. Hence, the negative

effect of *near perfect reconstruction* is avoided as the trailing dimensions are hardly used at all.

The main reason for doing this is not only that the classification accuracy actually increases, but rather to avoid using a preset projection depth that accidentally would cause *near perfect reconstruction* for some datasets. Hence, the best projection depth for different kinds of datasets could be chosen with the reassurance that it will always be reliable and never cause the accuracy to drop substantially. All these claims will be proven in the next subsection where the experiments are explained.

7.1 Experiments

First three sets were chosen for learning, AgeDB, CASIA and LFW. Since the EPSC does not perform back-propagation, no epochs are performed. Instead the learning is simply performed by dividing the set for training into one cluster for all males and another one for all females. Only one subspace for each class is then created using PCA, since it was experimentally noted that not much was gained by dividing these subspaces further using, for an example t-SNE as proposed earlier [HLV19]. Then the projected depth is learned by finding the optimal depth using the validation set.

Two different approaches were deployed for computing the best projection depth. The first approach divided each set into a learning set (60%) and a validation set (40%). Here the datasets were split on person, so that the same person was only present in one of the splits. Furthermore, it splits on sex so that there are both 60% of the males and females in the set for learning and 40% of each sex for validation,

The second approach was simply to learn from one of the three sets and validate on each of the other sets. Hence, using the three aforementioned datasets as training sets, and validation sets, one at a time, six different permutations are possible.

In each experiment, the optimal depth for sex classification using the FFVs as input to the EPSC, was determined. The proposed dampening function in equation (9) to set the weights ω was used and then 100 runs were performed using bootstrapping on three datasets at a time, as shown in table 2, where 60% was used as sets for training and the remaining 40% as validation sets. The result can be compared to table 3, where the average accuracy for the validation set, when computing the sex with the provided model from *InsightFace*. The tables show the accuracy for the two classes, and the Macro Average Arithmetic, which is just the mean of the two classes. This is done to avoid the effect of the fact that several of the datasets are heavily imbalanced and that could have a large impact on methods that are better to find one sex than the other. The MAA

Table 2: Average accuracies for 100 runs using bootstrapping with a 60/40 split and depth=360.

Database	Women	Men	MAA
CASIA	0.9995	0.9976	0.9986
AgeDB	0.9995	0.9907	0.9951
LFW	0.9900	0.9913	0.9907

Table 3: Accuracy for different datasets when the gender is classified by the face recognition model.

Database	Women	Men	MAA
CASIA	0.6824	0.7037	0.6930
AgeDB	0.8018	0.7293	0.7655
LFW	0.5651	0.6924	0.6288

Table 4: Optimal Depth for different datasets running 100 times using 60/40 split and bootstrapping. The mean optimal projection depth is 359.

Split	CASIA	AgeDB	LFW	Mean
60/40	361	373	342	359

is generally defined as the arithmetic average of the partial accuracies of each class:

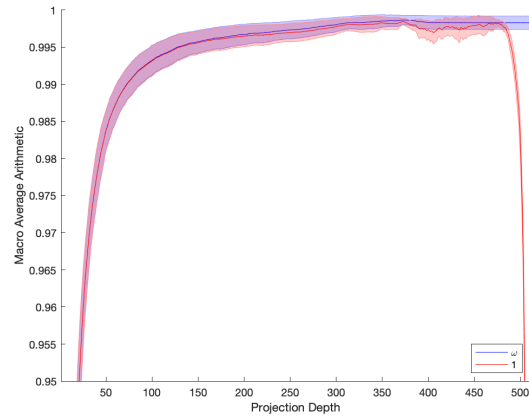
$$MAA = \frac{\sum_{i=1}^N ACC_i}{N} \quad (10)$$

where $N = 2$ for gender classification, as it the datasets have only two genders labeled. Hereby we avoid any discussions about what is actually gender, and how it relates to biological sex and perceived gender.

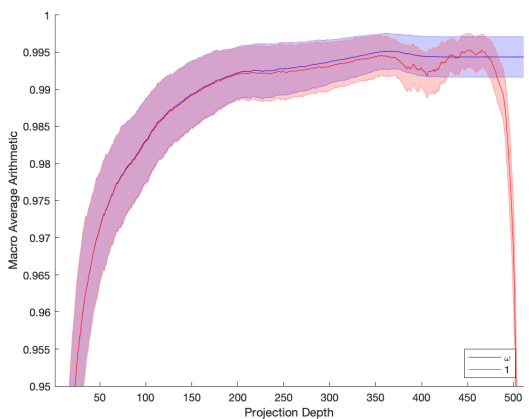
In the first approach it was chosen to compute the the MAA for sex classification using the Bootstrapping method [Koh95, KW96], with stratification because the data is imbalanced, which means that the bootstrap sample is taken from the whole set by using *sampling with replacement*. The experiments were conducted 100 times for each data split, varying the permutations randomly.

The results of the experiments are shown in Fig. 3 and are shown as a so called shaded bars graph, where the MAA is shown as a red curve with its shaded error (standard deviation) for $\omega = 1$. It can be noted that for high projection depths *near perfect reconstruction* is reached and the MAA drops rapidly. While for the blue curve, using equation (9) for computing ω , the curve flattens out. The latter ensures that the projection depth can be set to high values, without the dangers of reaching *near perfect reconstruction*.

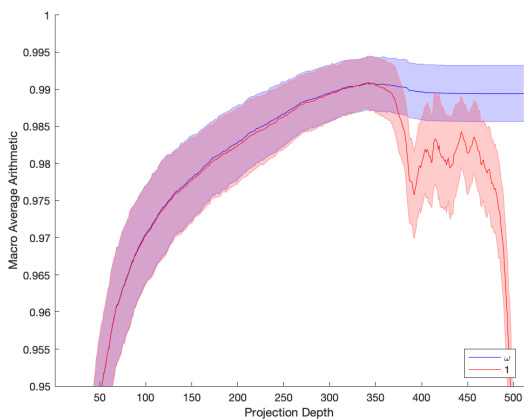
The optimal projection depth for the different datasets are reported in table 4. The optimal depth was computed when using ω in equation (9). The mean value is 359, which would still give a close to optimal MAA for all three datasets.



(a) CASIA



(b) AgeDB



(c) LFW

Figure 3: Macro Average Arithmetic for different depth of projection with shaded error (standard deviation), comparing dampening function ω (blue) for weights and using 1 as weight (red). Note how the former helps lifting the curve for larger projection depths in all three datasets

Table 5: Optimal Depth for different datasets for training and validation. The mean of all permutations, except those on the diagonal, is 360.

Dataset	AgeDB	CASIA	LFW
AgeDB	249	393	382
CASIA	328	395	352
LFW	375	328	347

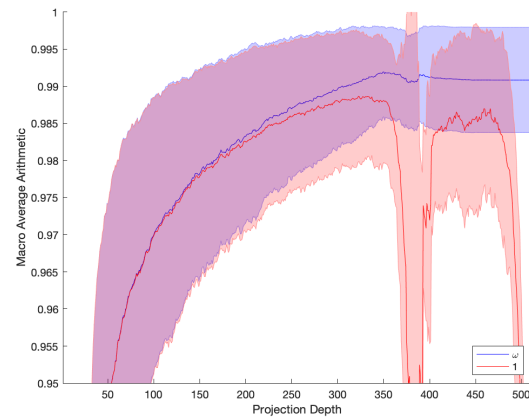


Figure 4: Macro Average Arithmetic for the mean of the 6 permutations of datasets with different depth of projection with shaded error (standard deviation), comparing dampening function ω (blue) for weights and using 1 as weight (red). Note how the former helps lifting the curve for larger projection depths on average for the datasets.

In Table 5 the optimal depth for all 6 possible permutations of training and validation sets is reported. Since using the same set for training and validation yields much better results, the results on the diagonal were not used when computing the mean equal to 359.6667. Once again ω was set as in equation (9).

Fig. 4 tells the mean of all six experiments, and clearly shows that the use of the dampening function for ω lifts the MAA curve and gives better accuracy than for $\omega = 1$.

The conclusion from both types of experiments is that a projection depth can safely be set around 360 for sex classification using different datasets for learning.

8 RESULTS

After concluding that 360 turns out to be a good value for the projection depth in the experiments, using different validation sets, the MAA was computed using this projection depth on different datasets for leaning and testing. Table 6 compares the MAA for using the EPSC approach and what accuracy is obtained by using the gender classification model provided by *Insight-Face*. It can be noted that EPSC produce state of the art

Table 6: MAA for Classification of Sex, using one dataset for learning and another one for testing.

Database	AgeDB	CASIA	LFW	CelebA	UTKFace	NSD
AgeDB	0.9994	0.9921	0.9822	0.9596	0.8847	0.9900
CASIA	0.9960	0.9994	0.9860	0.9769	0.9007	0.9900
LFW	0.9963	0.9977	0.9976	0.9737	0.9065	1.0000
CelebA	0.9966	0.9989	0.9885	0.9893	0.9152	0.9500
UTKFace	0.9974	0.9979	0.9897	0.9813	0.9488	1.0000
InsightFace	0.7593	0.6959	0.6640	0.6887	0.5700	0.5807

results for sex classification and also outperforms the deep learning model.

9 DISCUSSION

The question about a persons *Sex* is easier to understand than *Gender* since the former is something that usually is assigned at birth, even if it sometimes can be a difficult task because of different disorders. The latter on the other hand can be chosen later in life. Nonetheless, as stated before, these two words have become synonyms and are also used as such in this paper. However, it is closer at hand to talk about *Sex* classification for the method presented in this paper. The reason is that since FFVs are used, which are more closely related to the geometric features in a face, and therefore theoretically should be invariant to facial hair, hair style, makeup etc. Hence FFVs captures the geometry, which depends on genetics, rather than facial hair, hair-style, makeup etc, which all can be chosen. In any case, no political stance is taken in this paper about this matter. Furthermore, most datasets do not reveal on what grounds the sex or gender was chosen, so there is no other choice than trusting the labels and classify on them. However, the proposed algorithm, will lean towards classifying *Sex* rather than *Gender*, while many CNN based algorithms might pick up on both geometric features as well as outer features such as facial hair, hair-style and makeup, since these are the things that are visible in the face images.

The EPSC achieves state of the art classification on hard datasets. The way the dampening function is formulated in equation (9), which is used to set the weights ω is one important contribution in this paper. It makes the projection depth variable n less sensitive, and makes the EPSC reliable compared to not using the dampening function.

Looking at the results in table 6 one can note that the proposed approach, using FFVs as input to EPSC performs close to or precede the state of the art. Acien et al. [AMVR*19] [HRBLM07] achieved a 94.8% accuracy using VGGFace on the LFW dataset, while the EPSC achieves well over 98%.

For the CelebA dataset, both Liu et al [LLWT15] and Ranjan et al. [RPC16] reported the results from several implementations with an accuracy spanning from 90%

up to 98%. The table shows that the EPSC is close to 98% or over depending on what dataset was used for learning.

It should be noted though that some methods used a split from the same dataset for training, validation and testing. Hence, the results cannot be compared straight on, but rather gives an indication on how well the proposed algorithm works compared to the state of the art algorithms. Furthermore, the learned projection depth could be set differently depending on the data at hand, but here it was chosen to learn it from three datasets. Learning from the same dataset when doing some kind of cross validation tend to give even better results as shown in table 2. Nevertheless, the overall results are promising for taking on any challenging dataset using the EPSC.

Initially it was supposed that an age balanced dataset would improve the overall classification, and therefore subsets of the AgeDB and CASIA was created. Even subspaces for each age group was created and tested. However, no great improvement was noticed. In the future, it should be tested whether, the approach by Gong et al. [GLJ20] could be used by dividing the learning data into groups, creating a subspace for each race and sex. Nonetheless, several of the datasets used contain a mix of races already, and it seem to work very well anyway.

Interestingly, the UTKFace dataset has the lowest accuracies when it comes to testing, but often yields the best accuracy when it is used as the set for learning. One reason for it being hard to classify, is that it contains quite many small children. They are hard to tell, even for humans, whether they are boys or girls.

The second hardest set to classify was CelebA and similarly it is the second best for learning, with one exception and that is when classifying on the NSD. Nevertheless, it seems like curating a set, requiring different age groups, did not help much. Better is to use a set with a great variation from the start. Also, even if CelebA is much larger than UTKFace, the latter performed better, perhaps because it is more gender balanced.

10 CONCLUSION

The proposed improvements to the EPSC, including how to learn projection depth and the improved damp-

ening function, makes EPSC reliable and stable for sex classification of face images. The face feature vectors from face recognition software, do include geometric information that can be used to determine sex and will not be affected by outer cues, such as facial hair, makeup and hair-style, which might be picked up by CNN based approaches, since they capture whatever is visible in the images. The results show close to state of the art performance, and even precede many of the algorithms published.

11 ACKNOWLEDGMENTS

This work has been partially supported by the Swedish Research Council (Dnr 2020-04652; Dnr 2022-02056) in the projects *The City's Faces. Visual culture and social structure in Stockholm 1880-1930* and *The International Centre for Evidence-Based Criminal Law (EB-CRIME)*. The computations were performed on resources provided by SNIC through UPPMAX under project SNIC 2021/22-918.

12 REFERENCES

- [ADRS*19] Arrieta A. B., D'iaz-Rodríguez N., Ser J. D., Bennetot A., Tabik S., Barbado A., García S., Gil-López S., Molina D., Benjamins R., Chatila R., Herrera F.: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges towards responsible ai. *ArXiv abs/1910.10045* (2019).
- [AHM*18] Abbas H., Hicks Y., Marshall D., Zhurov A., Richmond S.: A 3d morphometric perspective for facial gender analysis and classification using geodesic path curvature features. *Computational Visual Media* 4 (01 2018), 1–16.
- [AMVR*19] Acien A., Morales A., Vera-Rodriguez R., Bartolome I., Fierrez J.: Measuring the gender and ethnicity bias in deep models for face recognition. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (Cham, 2019), Vera-Rodriguez R., Fierrez J., Morales A., (Eds.), Springer International Publishing, pp. 584–593.
- [BBD93] Burton A. M., Bruce V., Dench N.: What's the difference between men and women? evidence from facial measurement. *Perception* 22, 2 (1993), 153–176. PMID: 8474841.
- [BBH*93] Bruce V., Burton A. M., Hanna E., Healey P., Mason O., Coombes A., Fright R., Linney A.: Sex discrimination: How do we tell the difference between male and female faces? *Perception* 22, 2 (1993), 131–152. PMID: 8474840.
- [CPC19] Carvalho D. V., Pereira E. M., Cardoso J. S.: Machine learning interpretability: A survey on methods and metrics. *Electronics* 8, 8 (Jul 2019), 832.
- [DDD*23] Donckt J. V. D., Donckt J. V. D., Deprost E., Vandebussche N., Rademaker M., Vandewiele G., Hoecke S. V.: Do not sleep on traditional machine learning. *Biomedical Signal Processing and Control* 81 (mar 2023), 104429.
- [DGXZ19] Deng J., Guo J., Xue N., Zafeiriou S.: Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4690–4699.
- [DGZ*19] Deng J., Guo J., Zhou Y., Yu J., Kotsia I., Zafeiriou S.: Retinaface: Single-stage dense face localisation in the wild, 2019.
- [EEH14] Eidinger E., Enbar R., Hassner T.: Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security* 9, 12 (2014), 2170–2179.
- [GLJ20] Gong S., Liu X., Jain A. K.: Mitigating face recognition bias via group adaptive classifier, 2020.
- [GLS90] Golomb B. A., Lawrence D. T., Sejnowski T. J.: Sexnet: A neural network identifies sex from human faces. In *NIPS* (1990).
- [GM14] Gilani S. Z., Mian A.: Perceptual differences between men and women: A 3d facial morphometric perspective. In *2014 22nd International Conference on Pattern Recognition* (2014), pp. 2413–2418.
- [GRS*14] Gilani S. Z., Rooney K., Shafait F., Walters M., Mian A.: Geometric facial gender scoring: Objectivity of perception. *PLOS ONE* 9, 6 (06 2014), 1–12.
- [GSC*19] Gunning D., Stefik M., Choi J., Miller T., Stumpf S., Yang G.-Z.: Xai—explainable artificial intelligence. *Science Robotics* 4, 37 (2019).
- [Hai04] Haig D.: The inexorable rise of gender and the decline of sex: Social change in academic titles, 1945-2001. *Arch Sex Behavior* 33 (2004), 87–96.
- [Has22] Hast A.: Magnitude of semicircle tiles in fourier-space : A handcrafted feature descriptor for word recognition using embedded prototype subspace classifiers. *Journal of WSCG* 30, 1-2 (2022), 82–90.
- [Hin22] Hinton G.: The forward-forward algorithm: Some preliminary investigations, 2022.
- [HL20] Hast A., Lind M.: Ensembles and cascading of embedded prototype subspace classifiers. *Journal of WSCG* 28, 1/2 (2020), 89–95.
- [HLV19] Hast A., Lind M., Vats E.: Embedded prototype subspace classification : A subspace learning framework. In *The 18th International Conference on Computer Analysis of Images and Patterns (CAIP)* (2019), Lecture Notes in Computer Science, pp. 581–592.
- [HRBLM07] Huang G. B., Ramesh M., Berg T., Learned-Miller E.: *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.
- [HRGL05] Hoss R. A., Ramsey J. L., Griffin A. M., Langlois J. H.: The role of facial attractiveness and facial masculinity/femininity in sex classification of faces. *Perception* 34, 12 (2005), 1459–1474. PMID: 16457167.
- [HV21] Hast A., Vats E.: Word recognition using embedded prototype subspace classifiers on a new imbalanced dataset. *Journal of WSCG* 29, 1-2 (2021), 39–47.
- [Ins23] InsightFace: Insightface. <https://insightface.ai>, 2023. Accessed: 2023-02-30.
- [KLR*77] Kohonen T., Lehtio P., Rovamo J., Hyvärinen J., Bry K., Vainio L.: A principle of neural associative memory. *Neuroscience* 2, 6 (1977), 1065 – 1076.
- [KO76] Kohonen T., Oja E.: Fast adaptive formation of orthogonalizing filters and associative memory in recurrent networks of neuron-like elements. *Biological Cybernetics* 21, 2 (Jun 1976), 85–95.
- [Koh82] Kohonen T.: Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, 1 (Jan. 1982), 59–69.
- [Koh95] Kohavi R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2* (San Francisco, CA, USA, 1995), IJCAI'95, Morgan Kaufmann Publishers Inc., pp. 1137–1143.
- [Kri19] Krishnan M.: Against interpretability: a critical examination of the interpretability problem in machine learning. *Philosophy & Technology* (2019).
- [KRMV76] Kohonen T., Reuhkala E., Mäkisara K., Vainio L.: Associative recall of images. *Biological Cybernetics* 22, 3 (Sep 1976), 159–168.
- [KW96] Kohavi R., Wolpert D.: Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the Thirteenth In-*

- ternational Conference on International Conference on Machine Learning (San Francisco, CA, USA, 1996), ICML'96, Morgan Kaufmann Publishers Inc., pp. 275–283.
- [Laa07] Laaksonen J.: *Subspace classifiers in recognition of handwritten digits*. G4 monografiaväitöskirja, Helsinki University of Technology, 1997-05-07.
- [LH15] Levi G., Hassner T.: Age and gender classification using convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2015), pp. 34–42.
- [LLWT15] Liu Z., Luo P., Wang X., Tang X.: Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)* (December 2015), pp. 3730–3738.
- [LM14] Learned-Miller G. B. H. E.: *Labeled Faces in the Wild: Updates and New Reporting Procedures*. Tech. Rep. UM-CS-2014-003, University of Massachusetts, Amherst, May 2014.
- [LWY*17] Liu W., Wen Y., Yu Z., Li M., Raj B., Song L.: Sphreface: Deep hypersphere embedding for face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA, USA, jul 2017), IEEE Computer Society, pp. 6738–6746.
- [MH08] Maaten L. v. d., Hinton G.: Visualizing data using t-sne. *Journal of machine learning research* 9, Nov (2008), 2579–2605.
- [MH18] McInnes L., Healy J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints* (Feb. 2018).
- [MPS*17] Moschoglou S., Papaioannou A., Sagonas C., Deng J., Kotsia I., Zafeiriou S.: Agedb: The first manually collected, in-the-wild age database. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2017), pp. 1997–2005.
- [MR08] Mäkinen E., Raisamo R.: Evaluation of gender classification methods with automatically detected and aligned faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 3 (2008), 541–547.
- [NLSS04] Nordström M. M., Larsen M., Sierakowski J., Stegmann M. B.: *The IMM Face Database - An Annotated Dataset of 240 Face Images*. Tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, may 2004.
- [oe23] of Europe C.: Sex and gender. <https://www.coe.int/en/web/gender-matters/sex-and-gender>, 2023. Accessed: 2023-02-23.
- [OK83] Oja E., Kuusela M.: The alsu algorithm - an improved subspace method of classification. *Pattern Recognition* 16, 4 (1983), 421–427.
- [OK88] Oja E., Kohonen T.: The subspace learning algorithm as a formalism for pattern recognition and neural networks. In *IEEE 1988 International Conference on Neural Networks* (July 1988), vol. 1, pp. 277–284.
- [OPD96] O'Toole A., Peterson J., Deffenbacher K.: An 'other-race effect' for categorizing faces by sex. *Perception* 25 (02 1996), 669–76.
- [PWH98] Phillips P., Wechsler H., Huang J., Rauss P. J.: The feret database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing* 16, 5 (1998), 295–306.
- [RPC16] Ranjan R., Patel V., Chellappa R.: Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (03 2016).
- [SBLM17] Samek W., Binder A., Lapuschkin S., Mäkelä K.-R.: Understanding and comparing deep neural networks for age and gender classification. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)* (2017), pp. 1629–1638.
- [Sha18] Shapshak P.: Artificial intelligence and brain. *Bioinformatics* 14, 1 (2018), 38.
- [SHIA21] Sumi T. A., Hossain M. S., Islam R. U., Andersson K.: Human gender detection from facial images using convolutional neural network. In *Applied Intelligence and Informatics* (Cham, 2021), Mahmud M., Kaiser M. S., Kasabov N., Iftikharuddin K., Zhong N., (Eds.), Springer International Publishing, pp. 188–203.
- [SJ19] Shi Y., Jain A.: Probabilistic face embeddings. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 6901–6910.
- [SKP15] Schroff F., Kalenichenko D., Philbin J.: Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 815–823.
- [TYRW14] Taigman Y., Yang M., Ranzato M., Wolf L.: Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1701–1708.
- [VJ01] Viola P., Jones M.: Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (Dec 2001), vol. 1, pp. I–I.
- [WDH*19] Wang M., Deng W., Hu J., Tao X., Huang Y.: Racial faces in the wild: Reducing racial bias by information maximization adaptation network. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (Los Alamitos, CA, USA, nov 2019), IEEE Computer Society, pp. 692–702.
- [WLK*67] Watanabe W., Lambert P. F., Kulikowski C. A., Buxto J. L., Walker R.: Evaluation and selection of variables in pattern recognition. In *Computer and Information Sciences* (1967), Tou J., (Ed.), vol. 2, New York: Academic Press, pp. 91–122.
- [WP73] Watanabe S., Pakvasa N.: Subspace method in pattern recognition. In *1st Int. J. Conference on Pattern Recognition, Washington DC* (1973), pp. 25–32.
- [WWZ*18] Wang H., Wang Y., Zhou Z., Ji X., Gong D., Zhou J., Li Z., Liu W.: Cosface: Large margin cosine loss for deep face recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 5265–5274.
- [WZLQ16] Wen Y., Zhang K., Li Z., Qiao Y.: A discriminative feature learning approach for deep face recognition. In *Computer Vision – ECCV 2016* (Cham, 2016), Leibe B., Matas J., Sebe N., Welling M., (Eds.), Springer International Publishing, pp. 499–515.
- [YL14] Yi D., Lei Z., Liao S., Li S. Z.: Learning face representation from scratch, 2014.
- [ZYQ17] Zhifei Z., Yang S., Qi H.: Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), IEEE.

Perceptions of Colour Pickers in Virtual Reality Art-Making

Marylyn Alex	Burkhard C. Wünsche	Danielle Lottridge
University of Auckland	University of Auckland	University of Auckland
School of Computer	School of Computer	School of Computer
Science	Science	Science
Auckland, New Zealand	Auckland, New Zealand	Auckland, New Zealand
marylyn.alex@auckland.ac.nz	burkhard@cs.auckland.ac.nz	d.lottridge@auckland.ac.nz

ABSTRACT

Virtual reality art is reshaping digital art experiences, especially with the recent release of multiplayer 3D art applications, but may elicit different first impressions across different age groups which can impact their uptake. In particular, popular colour pickers based on HSV colour spaces may appeal differently to younger and older adults. We investigate first impressions of colour selection when shown with a discrete picker or a continuous HSV picker via an online survey with 63 adults and 24 older adults. We found that the discrete picker was seen as having more positive hedonic qualities overall; there were no differences between perceptions of adults and older adults. We discuss the implications of our findings for colour selection tools in virtual reality art-making.

Keywords

Colour selection, Colour picker, Colour palette, Virtual reality, VR art, Creativity tools

1 INTRODUCTION

First impressions are crucial in determining the likeability and thus adoption of a design [Gro16]. Given that a tool's aesthetics influences the user's perception of its utility and formation of the user's persistent attitude towards the tool [HT06, JWTY16, YKW22], it is critical to understand how features impact users' understanding, emotions, and expectations the first moment they are perceived. While there is a substantial body of research on virtual reality (VR) and its use for art making [HBM23], there are limited studies that explore factors that influence the first-impressions of VR art experiences.

Virtual reality can provide rich visual experiences and transferable skills [WFS97] as well as control over dynamic environments and measurements of responses [SR01]. Thus, it has been utilised for art and performance in hobbyist and therapeutic contexts, across age groups [Uge21]. A core function in digital painting is selecting a colour from a colour palette.

Most colour pickers in digital painting combine continuous and discretised subsets of 3D colour spaces such as HSV (Hue, Saturation, Value) and RGB (Red, Green, Blue). Novel colour pickers such as Brushwork's 2021 application [Sun22] offer discretised colours that can be mixed. A study of older artists found that some had reservations in engaging with VR art, and when they did, they had a passion for selecting the 'right' colour but had challenges in using the HSV picker to do so [AWL21]. It is not yet known how initial perceptions of discretised and HSV pickers impact impressions of VR art, in particular across age groups.

In this study, we investigated the first impressions of VR art colour pickers. We expand on preliminary results previously published as a conference poster [AWL22]. Our research question is:

What are perceptions of a discretised colour picker and HSV colour picker for adults (<60 years) and older adults (60+ years)?

2 RELATED WORK

2.1 Colour Selection in VR

Virtual reality art applications tend to feature colour pickers using different representations of the HSV colour space (Figure 1). Examples are Tilt Brush [Til20], Mozilla A-Painter [Ser20], Gravity Sketch [Ben18], and ANIMVR [NVR21b]. Brushwork [Bar21] uses a different approach and employs a discretised colour space.

The Tilt Brush colour picker contains a colour circle depicting hue and saturation and a vertical scroll bar on the right to adjust the brightness. The A-painter colour picker is similar. It also contains a colour circle for hue and saturation selection, and surrounding it is a brightness slider and fields that show the current selected colour and colour history.

The Gravity Sketch colour picker is three dimensional. The circular section displaying hues and saturation can be pushed inwards and outwards in order to change the intensity. The colour circle is surrounded by twelve 3D blocks representing pure hues, and there are other

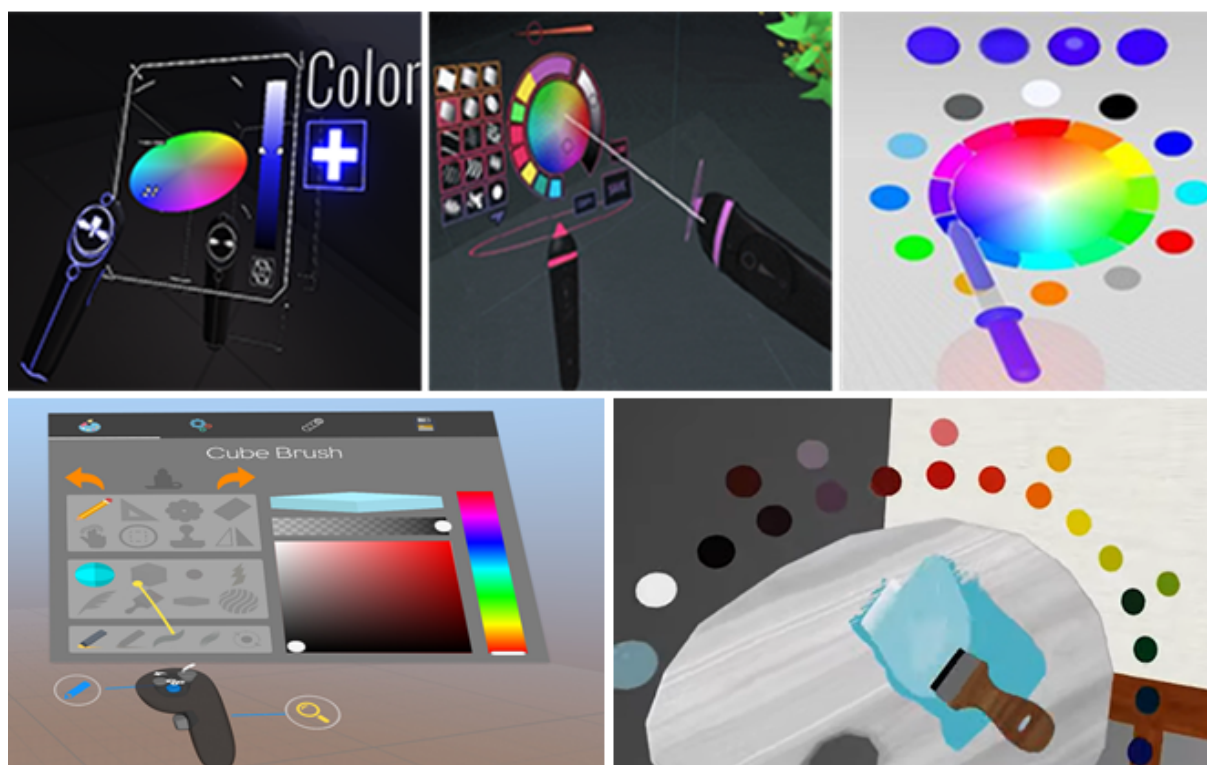


Figure 1: From left to right and top to bottom: Colour picker from Tilt Brush (Credit to: Juegos) [Jue20], Colour picker in Mozilla's A-Painter (Credit to: Fernando Serrano) [Ser20], Colour picker in Gravity Sketch (Credit to: xR.design) [xR.wn], Colour picker in ANIMVR (Credit to: NVRMIND) [NVR21a], Colour picker on Brushwork (Credit to Harry Barker) [Bar21]

smaller circles surrounding the hues displaying the recently selected colours. There are four circles on top of the colour picker that enable users to choose different shading methods for rendering the colour.

In ANIMVR, the square box in the middle adjusts the colour's brightness and a vertical scroll bar on the right adjusts the colour's hue and saturation. The opacity of the colour can be adjusted by dragging the opacity bar's cursor, which is right above the colour box. Above the opacity bar is the background colour bar.

Brushwork has brushes with two modes. The first brush mode allows users to paint a different colour on the existing layer of paint while the second mode allows the user to mix colours together. When in the first mode, users can pick up and hold the colour palette and brushes at any angle to paint. The Brushwork's colour picker offers a relatively small selection of colours, however the second mode allows users to change shade of colours through mixing.

As shown by this review of popular tools, the HSV picker is a common paradigm for digital painting colour picker, however it may not be appropriate for all types of users. Selecting a specific colour in HSV colour spaces can be challenging for novices due to not understanding the underlying colour model and having difficulties in finding a colour within the colour

space [AWL21, LM04]. Furthermore, colour may appear differently in the application than in the colour picker because of simultaneous contrast [EF12]. Typical colour pickers represent individual colours on a very small space which may be as small as the size of a single pixel. This small target makes it difficult to identify specific colour in 2D or 3D colour spaces [PY17]. Discretised colour pickers can help to solve some of the usability challenges with HSV pickers [ALL+20]. However, it is not known how initial perceptions of colour pickers differ, which can influence adoption of an application when first seeing it.

2.2 Therapeutic Uses of VR Art

Art therapy is a popular complementary therapy to treat a wide variety of health problems [JRWB22]. VR art-making has been investigated as an approach to make art therapy more accessible and better address patients' needs. The presence, immersion, point of view, and perspective within the virtual environment, along with virtual materials and unreal characteristics give VR much potential for the practice of art therapy [HRS18]. VR art-making has been found to be enjoyable, engaging and therapeutic for older adults with dementia and depression [PDHR17] and for older adults with neurocognitive disorders [WCWS+13].

Tilt Brush was evaluated in the context of art making programs in an art therapy studio [AWL21] with older adults with physical and/or cognitive impairments. The field study showed that artists tend to draw inspiration from natural scenes and materials in their art-making. In standard digital colour selection tools, the types of natural colours that artists chose for traditional painting were not immediately visible. Additionally, the field study also found that participants were unable to find their desired colours (i.e., brown, black, white, and bright blue). This motivates research on the usability of discrete pickers, which was investigated by Alex et al. who found that participants used different colours when painting with the discrete versus HSV pickers, however they found no differences in usability between them [ALL+20].

3 METHODS

In this study, we were interested in investigating participant's perceptions and first impressions of the colour pickers. These aspects of VR art applications could be customised based on different populations, and thus we want to better understand whether first impressions of older adults differ from adults in order to better customise these applications in the future. We hypothesised that older adults might have different impressions of the discrete picker and HSV picker compared to adults.

3.1 Design of the Stimuli

The VR art-making application was built using Unity3D software. Because popular tools such as Tilt Brush [Til20], Mozilla A-Painter [Ser20], Gravity Sketch [Ben18], and ANIMVR [NVR21b] all contain a space to select hue and saturation with a function to adjust the brightness, we selected an HSV picker that offers similar basic functions. It has a round circle for selecting hue and saturation and a triangle in the middle to adjust the brightness. The cube below the circle shows the colour selected by user. The HSV picker was downloaded from the Unity Asset Store (Figure 2 (right)). We also utilised the discrete picker from Alex et al. [ALL+20]. Discrete picker allows its users to select a colour with a single step without the need to adjust hue and saturation/intensity. It contains seven groups of small colour wheels. These wheels consist of the three primary colours (i.e., yellow, red, blue), three secondary colours (i.e., orange/brown, purple, green), and greyish colours (black to white). The circle in the middle shows the colour selected by the user (Figure 2 (left)).

We created four videos [VRA21] displaying identical scenes of VR art-making (Figure 3). Two of these videos used the discrete picker and two the HSV colour picker, with similar times spend on the colour selection

process. For each colour picker we added to one of the videos an additional 13-second footage of an artificial companion (AC). The evaluation of the artificial companion will be reported in another paper. This resulted in four near-identical videos showing:

1. the discrete picker without the AC
2. the discrete picker with the AC
3. the HSV picker without the AC
4. the HSV picker with the AC.

The two videos without the AC were 4m 03s long and the two videos with the AC were 4m 17s long.

In this paper we will report our findings for participants' perception of the colour pickers. The effects of artificial companions are discussed in previous research [ALW20, AWL22].

3.2 Survey

To answer our research question, we conducted an online survey using Qualtrics. We selected a survey methodology in order to gather first impressions from a larger and more diverse audience than could be reached in another manner. The online survey comprised a pre-video demographic questionnaire, one of two recorded online videos, and a post-video questionnaire. The online survey took approximately 15 - 25 minutes to complete. The post-video questionnaire consisted of three major sections: the first was on VR art-making in general, the second on the colour picker and the third on the AC.

The first section on general art-making consisted of 15 closed-ended questions on general perception of the VR art-making and one open-ended question to obtain qualitative feedback. The section on first impressions of the colour pickers contained 14 closed-ended questions and one open ended question. The first question (closed-ended) assessed the participant's satisfaction with the range of colours in the colour picker. The remainder of the closed-ended questions were semantic differential scales grouped into three subgroups: Pragmatic Quality (PQ), Hedonic Quality (HQ), and APPEAL [HST08] (Table 1). PQ refers to the participants' thoughts on the effectiveness of the colour picker in fulfilling its main task (i.e., painting). HQ refers to participants' sense of how stimulated they were by the colour picker. APPEAL refers to participants' general evaluation of the colour picker. The single open-ended question in the second section gathered qualitative feedback on participants' perceptions of the colour picker. Beyond these subgroups, there was an additional semantic differential item Social/Isolating. The single open-ended question in the second section gathered qualitative feedback

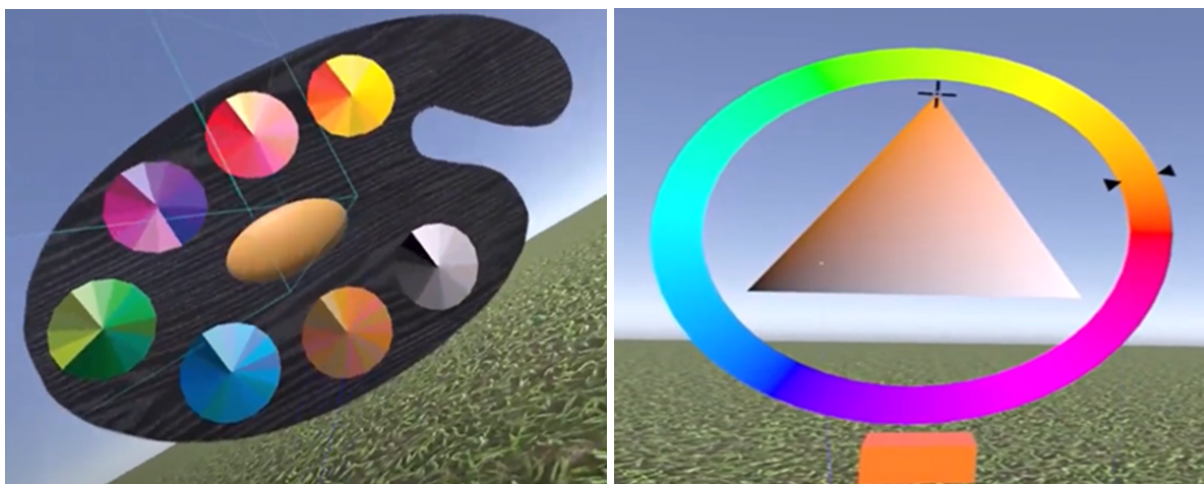


Figure 2: Two colour pickers: Discrete picker (left), HSV picker (right)

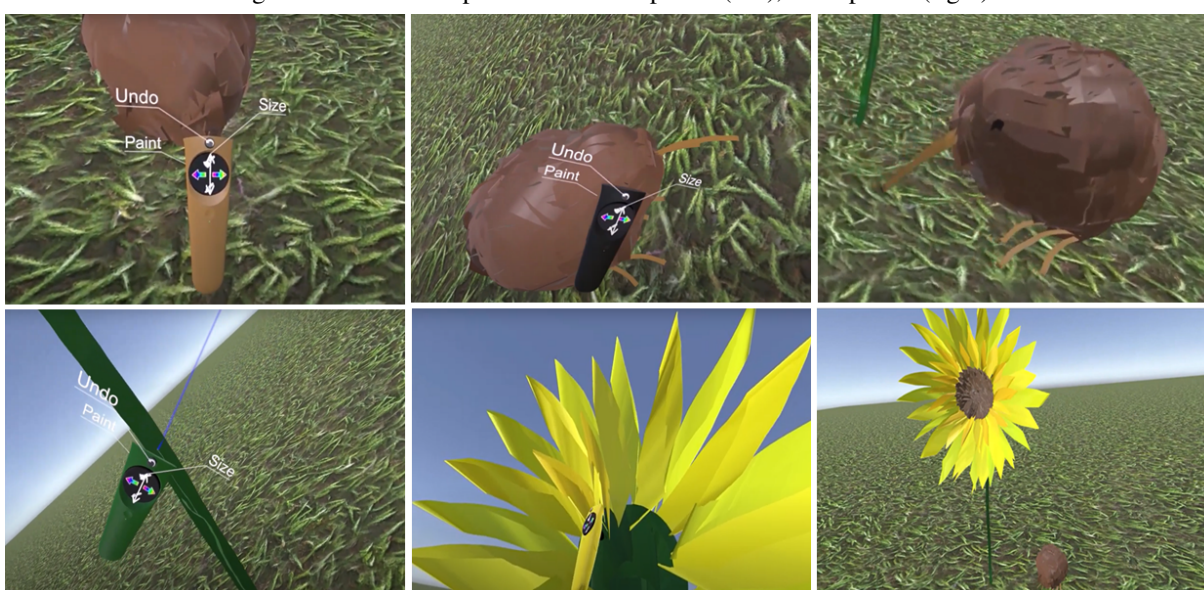


Figure 3: All participants watched the same drawing scenes: successive brushstrokes creating a kiwi bird and a sunflower.

on participants' perceptions of the colour picker. The third section contained seven closed-ended questions and five open-ended questions. This section was only made available to participants who were randomly allocated to watch videos with the AC present. In this paper, we only report the results of the VR art-making colour pickers.

3.3 Analysis

The statistical analysis was conducted using the SPSS statistics software. We did a factor analysis, reliability analysis, analysis of variance (ANOVA), and multivariate analysis of variance (MANOVA) to analyse the data. The factor analysis grouped the semantic differential items into three groups: PQ, HQ, and APPEAL. We scored each group by averaging its items.

We read all the quotes from the survey and created affinity diagrams. We used affinity analysis [Ash20, HB97] to group the quotes from the survey based on 'similarity'.

3.4 Participants

To recruit a wide variety of participants including adults and older adults, we distributed invitations to participate via email and social media to multiple organisations all over the world including stroke organisations, universities, and other organisations (e.g., SeniorNet, retirement homes etc.). In the invitation, we let participants know they could complete the survey with the assistance of a caregiver.

A total of 87 participants participated (Table 2). Participants comprised 24 older adults aged 60 and above with

Scale item	Anchors	
PQ 1	Comprehensible	Incomprehensible
PQ 2	Supporting	Obstructing
PQ 3	Simple	Complex
PQ 4	Clear	Confusing
PQ 5	Controllable	Uncontrollable
HQ 1	Interesting	Boring
HQ 2	Exciting	Dull
HQ 3	Impressive	Nondescript
HQ 4	Original	Ordinary
APPEAL 1	Pleasant	Unpleasant
APPEAL 2	Attractive	Unattractive
APPEAL 3	Motivating	Discouraging
APPEAL 4	Desirable	Undesirable

Table 1: Semantic differential items for Pragmatic Quality (PQ), Hedonic Quality (HQ), and general evaluation (APPEAL) [HST08]

	DISCRETE	HSV
Adult	27	36
Older Adults	16	8

Table 2: Number of participants who watched discrete or HSV picker video

a mean age of 69.9 years, 62 adult people aged 16 - 59 years with a mean age of 33.2 years and one participant did not provide his age.

42 participants identified as male, 43 participants identified as female, one participant identified as non-binary, and one participant did not provide an answer. We are missing two pieces of demographic data from the survey as one participant did not provide his age and another participant did not provide a gender.

4 FINDINGS

A reliability analysis found high values for PQ ($\alpha = .926$), HQ ($\alpha = .943$), and APPEAL ($\alpha = .964$), indicating a high internal consistency among the semantic items within the group. The MANOVA analysis showed that the colour picker had a significant effect on HQ ($F(1, 86) = 5.35, p = .023$), with the discrete picker garnering higher scores, but not PQ ($F(1, 86) = .077, p = .781$) and APPEAL ($F(1, 86) = 2.47, p = .120$). Figure 4 illustrates the results. Overall, there were no differences between adults and older adults. Therefore, our hypothesis that older adults might have different impressions of the discrete picker and the HSV picker compared to adults is rejected. We conducted an ANOVA to examine the satisfaction with the range of colours in the colour pickers and found no statistical differences between the discrete picker and the HSV picker. We grouped participants' open-ended comments into three subthemes which we explain next.

4.1 Colour Pickers' Colour Ranges

There was a mix of satisfaction with the discrete picker's colour range. Most participants were satisfied with the range of colours in the discrete picker, with comments such as: "[It] has all the colour types available [P9]", "There is a wonderful choice of colours [P18]", "Sufficient range of colours [P37]", "There seem to be enough colours to use [P80]", "All the colours are there [...] plenty of choice [P85]". One participant mentioned that there is a reasonable choice of colours, but stated: "Undoubtedly, I would eventually want a colour not on there, but it looks like a reasonable selection of colours nevertheless [P3]". Some participants were dissatisfied with the range of colours in the discrete picker. One participant commented: "It should be variables and options to choose more color [P38]". It seems like the discrete picker may have enough selection for many but not all users. Two other participants felt unable to answer the question. P14 stated: "I don't know much about colours".

There were many participants who were satisfied with the range of colours in the HSV picker. Those who commented about that said, "All the colours of the rainbow seem to be there [P47]", "RGB colour wheel is pretty much every colour [P61]", "Looks beautiful with complete rainbow colours [P63]", "Has almost all colours that's needed to create stuffs [P65]", "[...] most of the colours are available [P69]", "Enough colours to work with [P70]".

There were some participants who were dissatisfied with the range of colours in the HSV picker. One of them stated, "[...] When there are too many colors, one faces a paralysis of choice [P79]". Another participant had the opposite reaction, that the HSV picker was, "[...] very limited [P40]". P40 may have misunderstood the HSV, as happened in Alex et al.'s study [AWL21], where participants thought the colour picker only contained the colours that were immediately visible and did not imagine all the variations of saturation were available. The fully saturated colours around the HSV picker wheel might attract users with strong colour preferences. Two participants commented on the colour preferences for HSV picker such as, "I like the strong colours [P55]", "Like the color [P73]". These comments indicate that they were attracted to vibrant colours which may be more eye-catching relative to the desaturated colours in the discrete picker.

4.2 Design Satisfaction

Participants commented on the design of the colour pickers. Participants commented on the colour arrangement within the discrete picker: "They are well mixed perfectly arranged colours [P10]", "The colours are well organised and beautifully done [P22]", "The mixing of colours are in order I must say it's complete

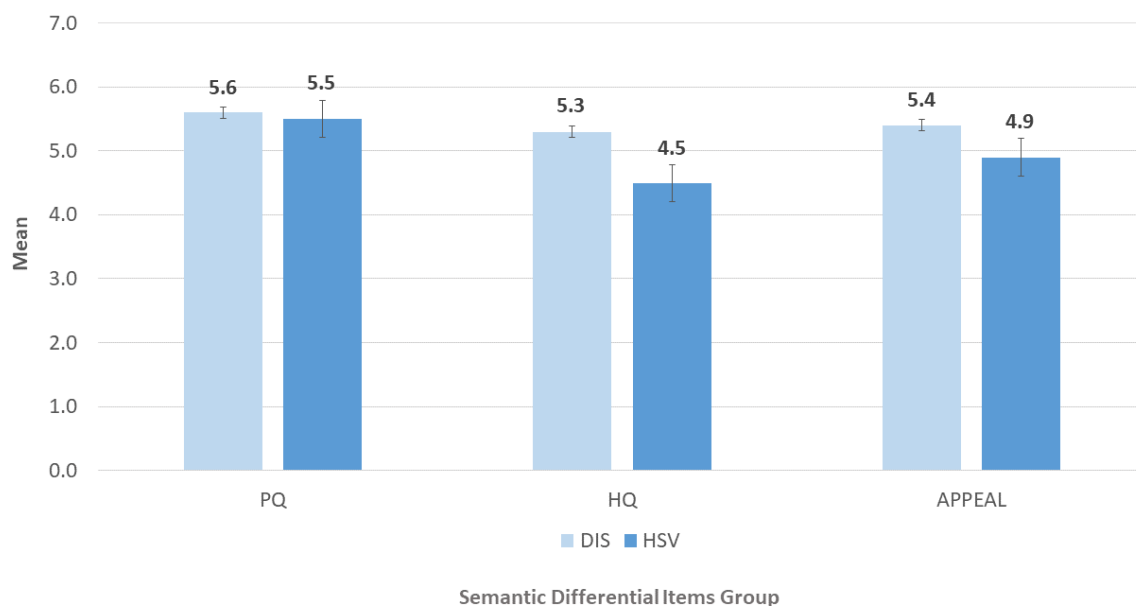


Figure 4: Discrete picker (DIS) versus HSV picker

[P26]”, “Complete and well-arranged [P31]”. One participant mentioned the discrete picker is “very comprehensive [P15]”. P13 suggested that it will be better if it is bigger. We cautiously interpret P13 as referring to the size of the colour picker wheels and propose that it may be beneficial to be able to magnify particular wheels. Three participants provided comments on what they disliked about the discrete picker. P16 said that “it looks mediocre”, P38 stated that “The quality of the colour picture is low density and look unattractive”, and P86 mentioned that “Lots of colours muddled together. How would I choose a colour?”. Another felt unable to comment on the design: P56 said that he has no experience and needed to use it in order to be able to comment.

Participants also gave mostly positive and some negative reviews on the design of the HSV picker. “RGB ring and shade/tint triangle is mostly understandable for anyone exposed to modern digital art tools [P41]”, “The colour picker seems pretty straight forward to use [P49]”, “It looks standard to me. I think it is adequate for what a screen can display [P46]”. These comments seem to indicate that some participants have experience and are already quite familiar with the HSV picker. Two participants were dissatisfied with the HSV picker. P79 said, “There are other ways to pick color, not just a color wheel” and P59 commented: “The HSL colour picker, while a traditional mainstay in professional art applications, seems out of place in the three dimensional world of VR. It also requires understanding of how it operates, and that the outer ring and inner triangle are linked, before it can be used effectively [...]

the colour picker is unsuitable for VR, and potentially daunting for inexperienced artists who have not seen or used it before. A novel colour picker, specifically designed for use in three dimensions so that all available colours can be seen at once, around the user, would be better suited for VR”.

4.3 Customisation

Multiple participants desired customisation. After seeing the discrete picker, P21 stated that, “I’d want a customised palette chooser where you can roll your own”. P80 and P85 asked if it is possible to mix colours.

Two participants who had seen the HSV picker suggested customisation features. P41 suggested having a way to store previously-picked colours that could be retrieved easily. The other participant suggested “[to have] an option to be able to choose from a colour palette [...]” and “[...] a way to save colours used in the art [...] [P61]”.

5 DISCUSSION

We conducted a quantitative survey study to rigorously assess initial perceptions of virtual reality colour pickers within VR art. We were particularly interested in how older adults’ perceptions might differ from adults’ perceptions. We found that the discrete picker scored higher overall for its hedonic qualities of being more interesting, exciting, impressive and original compared to the HSV picker, and there were no differences across age groups.

In our study design, participants were randomly allocated to one of four near-identical videos. The strength

of this study design is that we can attribute causal effects to the manipulated variables. As each participant only sees a single stimulus, there are no order effects or transfer effects. The study design is intended to have some similarities to the experience of first perceptions of new applications, where people tend to see images, videos or advertisements before being able to try a tool themselves. These first impressions are an important step toward deciding whether to try a new application or not. We argue that assessing initial perceptions is worthwhile as negative initial perception may introduce barriers that may not be known if a user is funneled into initial use. A drawback of our study design is that participants do not experience all the options, and so cannot comment on the differences and their preferences. Further, initial perceptions may differ from perceptions after a period of usage. The discrete picker's higher hedonic qualities may be due to its novelty, colour wheels, or other aspects of its design. The semantic differential items could be administered after a period of art-making to see how first impressions and usage impressions are related, and whether habituation to the design impacts perceptions.

The discrete picker is 2-dimensional whereas the HSV enables traversal through 3-dimensional colour space with a second colour selection step. Some initial impressions of the HSV were based on the visible colours — it was intriguing that the vibrant, high saturation colours of the HSV picker's wheel attracted some people's attention. A weakness of the 2-dimensional design is that it does not utilise the full potential of virtual reality. This presents a dilemma on complexity. Even 2D colour pickers may be difficult to understand for novice users, as shown by some of the comments we received. To a novice, the colours displayed around the HSV colour wheel look may appear limited as the HSV picker seems to contain only fully saturated colours. Three dimensional colour pickers would take advantage of the immersive environment however they may be even more difficult to understand.

People who are novices in digital art may require simpler tools [AWL21] whereas those with understanding of colour spaces can be offered complex 3-dimensional tools. Simple tools will be unlikely to satisfy expert digital artists who are accustomed to more choice. For instance, some of our participants mentioned the limited range of colours in the discrete picker. Others mentioned a potential paralysis of choice using the HSV picker because there are too many colours. Thus, our findings are in line with [LM04, ALL+20]: the discrete picker may be more suitable during a novice period of digital art, whereas the HSV picker is more suitable for experienced users who like to explore more colours. We recommend that digital art applications have the ability to evolve the sophistication of the toolset as the artist

becomes more familiar with the toolset, so that the complexity grows as the user is better able to utilise it.

6 LIMITATIONS

Our study has several limitations. Since our study was accessible to everyone self-selection bias is present. It is possible that users participating in the research are more curious, interested in VR, and open to new ideas, and hence findings might not be representative.

Since we assigned participants randomly to each video, the demographics of participants watching each video was not evenly distributed. For example, we had 16 older adults watching videos with the discrete colour picker, but only eight participants watching videos with the HSV colour picker. This also resulted in small sample sizes for some conditions.

Some participants in our online survey might suffer from impairments (cognitive and physical, e.g., stroke) and perceptions and expectations might be different from those of a healthy user.

7 CONCLUSION AND FUTURE WORK

This research investigated perceptions of VR colour pickers on the VR art-making experience from a wider range of potential users. Compared to the HSV picker, the discrete picker had higher hedonic qualities: it was seen to be more interesting, exciting, impressive, and original. Qualitative feedback suggested that the discrete picker may be more suitable for novice users who do not have knowledge in 3D colour space, while experienced users would appreciate the range of colours offered by the HSV picker.

In future work we would like to test the different colour pickers with participants using them for VR art applications and investigate how initial perceptions and actual experiences differ. For example, participants might be unaware that pointing towards a location with a VR controller is for many users harder than when using a mouse, e.g., due to slight hand tremours and not being able to rest the hand on a surface.

More work is needed investigating the tool's accessibility and usability such as adding customisation functions to the colour picker. For example, creating a function that allows users to blend or mix colours or switch colour pickers (e.g., discrete or HSV) and a magnifier tool for the discrete colour picker to make it easier to select a specific colour. We also would like to have an option to enable or disable colour selection with a single controller to support accessibility.

We observed in previous research that for many older adults art-making is a social experience [AWL21]. We

would like to expand the tool to support social interactions, e.g., users discussing colour choices and their affect on the art piece.

Music is a powerful motivator, can improve mood, and support creativity [EASG20]. We hope to integrate music into the art making process both as background music and by visualising the music and enabling users to interact with it [TWM19].

8 ACKNOWLEDGEMENTS

We would like to thank all the organisations who have assisted in this study, and all the participants who took part in this study. We also would like to thank the reviewers for their valuable feedback. The research was approved by University of Auckland Human Participants Ethics Committee reference number UAH-PEC22162.

9 REFERENCES

- [ALL+20] Marylyn Alex, Danielle Lottridge, Jisu Lee, Stefan Marks, and Burkhard C. Wünsche. Discrete versus continuous colour pickers impact colour selection in virtual reality art-making. In 32nd Australian Conference on Human-Computer Interaction, pages 158–169, New York, NY, USA, 2020. ACM.
- [ALW20] Marylyn Alex, Danielle Lottridge, and Burkhard C. Wünsche. Artificial companions in stroke rehabilitation: Likeability, familiarity and expectations. Proceedings of the 53rd Hawaii International Conference on System Sciences (HICCS '20), pp. 3789–3798, 2020.
- [Ash20] Ashima Goel. Analysis of qualitative data using affinity diagram and pareto principle (with working example). <https://medium.com/the-product-clan/analysis-of-qualitative-data-using-affinity-map-and-pareto-principle-with-wor2020>. Accessed: October 22.
- [AWL21] Marylyn Alex, Burkhard C. Wünsche, and Danielle Lottridge. Virtual reality art-making for stroke rehabilitation: Field study and technology probe. International Journal of Human-Computer Studies, 145:102481, 2021.
- [AWL22] Marylyn Alex, Burkhard C. Wünsche, and Danielle Lottridge. Perceptions of Colour Pickers and Companions in Virtual Reality Art-Making. Proc. of the 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Christchurch, New Zealand, pp. 564–565, 2022.
- [Bar21] Harry Barker. Brushwork VR offers free webxr painting in your headset's browser. <https://uploadvr.com/brushwork-vr-webxr-painting/>, 2021. Accessed: October 22.
- [Ben18] Neil Bennett. Gravity sketch is the first pro-level VR app for artists. <https://www.digitalarts.co.uk/news/creative-software/gravity-sketch-is-first-pro-level-vr-app-for-artists/>, September 2018.
- [EASG20] Katherine E. Eskine, Ashanti E. Anderson, Madeline Sullivan, and Edward J. Golob, Effects of music listening on creative cognition and semantic memory retrieval. *Psychology of Music*, 48(4), pp. 513–528, 2020.
- [EF12] Vebjorn Ekroll and Franz Faul. Basic characteristics of simultaneous color contrast revisited. *Psychological Science*, 23(10):1246–1255, 2012.
- [Gro16] Guillaume Gronier. Measuring the first impression: Testing the validity of the 5 second test. *Journal of Usability Studies*, 12(1), 2016.
- [HB97] Karen Holtzblatt and Hugh Beyer. Contextual design: defining customercentered systems. Elsevier, Amsterdam, The Netherlands, 1997.
- [HBM23] Christos Hadjipanayi, Domna Banakou, Despina Michael-Grigoriou. Art as therapy in virtual reality: A scoping review. *Frontiers Virtual Reality*, 4:1065863, pp. 1–16, 2023.
- [HRS18] Irit Hacmun, Dafna Regev, and Roy Salomon. The principles of art therapy in virtual reality. *Frontiers in Psychology*, 9:2082, 2018.
- [HST08] Marc Hassenzahl, Markus Schöbel, and Tibor Trautmann. How motivational orientation influences the evaluation and choice of hedonic and pragmatic interactive products: The role of regulatory focus. *Interacting with Computers*, 20(4-5):473–479, 2008.
- [HT06] Marc Hassenzahl and Noam Tractinsky. User experience-a research agenda. *Behaviour & Information Technology*, 25(2):91–97, 2006.
- [JRWB22] Ronja Joschko, Stephanie Roll, Stefan N. Willich, and Anne Berghöfer. The effect of active visual art therapy on health outcomes: protocol of a systematic review of randomised controlled trials. *Systematic Reviews*, 11(1):1–8, 2022.
- [Jue20] Juegos. Tilt brush by google. <https://vrexperience.es/portfolio/tilt-brush>, 2020. Accessed: November 22.
- [JWY16] Zhenhui Jiang, Weiquan Wang, Bernard C.Y. Tan, and Jie Yu. The determinants and impacts of aesthetics in users' first interaction with websites. *Journal of Management Information Systems*, 33(1):229–259, 2016.
- [LM04] Paul Lyons and Giovanni Moretti. Nine tools for generating harmonious colour schemes. In Proceedings of the 6th Asia Pacific Conference on Computer Human Interaction, pages 241–251, New Zealand, 2004. Springer.

- [NVR21a] NVRMIND. AnimVR code. <https://store.steampowered.com/app/508690/AnimVR/>, 2021. Accessed: October 22.
- [NVR21b] NVRMIND. AnimVR homepage. <https://nvrmind.io/features>, October 2022.
- [PDHR17] Alexander Paczynski, Laura Diment, David Hobbs, and Karen Reynolds. Using technology to increase activity, creativity and engagement for older adults through visual art. In *Mobile e-Health*, pages 97–114. Springer, Berlin/Heidelberg, Germany, 2017.
- [PY17] John H. Pula and Carlen A. Yuen. Eyes and stroke: the visual aspects of cerebrovascular disease. *Stroke and Vascular Neurology*, 2(4):210–220, 2017.
- [Ser20] Fernando Serrano. A-painter: Paint in VR in your browser. <https://medium.com/@fernandojs/a-painter-paint-in-vr-in-your-browser-ecac221fda1d>, 1998-2020. Accessed: November 22.
- [SR01] Maria T. Schultheis and Albert A. Rizzo. The application of virtual reality technology in rehabilitation. *Rehabilitation Psychology*, 46(3):296, 2001.
- [Sun22] SunsetDivision. Brushwork. <https://brushworkvr.com/>, 2022. Accessed: October 2022.
- [Til20] Tilt Brush. Tilt brush by google. <https://www.tiltbrush.com>, 2022. Accessed: November 2022.
- [TWM19] Michael Taenzer, Burkhard C. Wünsche, and Stefan Müller. Analysis and Visualisation of Music. *Proceedings of the International Conference on Electronics, Information, and Communication (ICEIC '19)*, Auckland, New Zealand, pp. 1–6, 2019.
- [Uge21] Jeppe Ugelvig. 8 artists pushing the limits of digital effects and VR. <https://www.artsy.net/article/artsy-editorial-8-artists-pushing-limits-digital-effects-vr>, 2021. Accessed: October 2022.
- [VRA21] VRArtResearch. VR Art code <https://github.com/VRArtResearch/VRArtResearch>, 2021. Accessed: April 2022.
- [WCWS+13] Lise Worthen-Chaudhari, Cara N. Whalen, Catherine Swendal, Marcia Bockbrader, Sarah Haserodt, Rashana Smith, Michael Kelly Bruce, and W Jerry Mysiw. A feasibility study using interactive graphic art feedback to augment acute neurorehabilitation therapy. *NeuroRehabilitation*, 33(3):481–490, 2013.
- [WFS97] Paul N. Wilson, Nigel Foreman, and Danae Stanton. Virtual reality, disability and rehabilitation. *Disability and rehabilitation*, 19(6):213–220, 1997.
- [xR.wn] xR.design. 3d design patterns. <http://www.xr.design/patterns/64-3d-color-picker>, Accessed: November 2022
- [YKW22] Elisa Yansun, Daniel Kim, and Burkhard C. Wünsche. CoXercise - Perceptions of a Social Exercise Game and its Effect on Intrinsic Motivation. *Proceedings of the 2022 Australasian Computer Science Week (ACSW '22)*. ACM, New York, NY, USA, pp. 176–185, 2022.

StarSRGAN: Improving Real-World Blind Super-Resolution

Khoa D. Vo

Faculty of Information Technology (FIT)
University of Science, VNU.HCM
Ho Chi Minh City, Vietnam
20c11008@student.hcmus.edu.vn

Len T. Bui

Faculty of Information Technology (FIT)
University of Science, VNU.HCM
Ho Chi Minh City, Vietnam
btlen@fit.hcmus.edu.vn



Figure 1: Comparison between our models and some standard SRGAN models. (**Zoom in for best view**)

ABSTRACT

The aim of blind super-resolution (SR) in computer vision is to improve the resolution of an image without prior knowledge of the degradation process that caused the image to be low-resolution. The State of the Art (SOTA) model Real-ESRGAN has advanced perceptual loss and produced visually compelling outcomes using more complex degradation models to simulate real-world degradations. However, there is still room to improve the super-resolved quality of Real-ESRGAN by implementing recent techniques. This research paper introduces StarSRGAN, a novel GAN model designed for blind super-resolution tasks that utilize 5 various architectures. Our model provides new SOTA performance with roughly 10% better on the MANIQA and AHQ measures, as demonstrated by experimental comparisons with Real-ESRGAN. In addition, as a compact version, StarSRGAN Lite provides approximately 7.5 times faster reconstruction speed (real-time upsampling from 540p to 4K) but can still keep nearly 90% of image quality, thereby facilitating the development of a real-time SR experience for future research. Our codes are released at <https://github.com/kynthesis/StarSRGAN>.

Keywords

Blind super-resolution, adaptive degradation, dual perceptual loss, multi-scale discriminator, dropout degradation

1 INTRODUCTION

The field of image super-resolution (SR) aims to reconstruct a high-resolution (HR) image from a low-resolution (LR) counterpart, which is a challenging task due to the many-to-one mapping involved. The computer vision research community has devoted significant attention to SR, and deep learning algorithms have

been successfully applied to this problem. These techniques utilize neural networks to train an end-to-end mapping function, such as the SRCNN [Don16], which involves deep convolutional neural networks (CNNs) that generate high signal-to-noise ratio (PSNR) values. Still, the output is often excessively smoothed and needs more high-frequency features.

Researchers suggest using generative adversarial networks (GANs) [Goo14] for image SR tasks to overcome these limitations. A super-resolution GAN comprises a generator network and a discriminator network. The generator takes LR images as input and aims to create images that resemble the original HR image. At the same time, the discriminator distinguishes between "fake" and "real" HR images. However, these methods

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

assume an ideal bicubic downsampling kernel, making them unsuitable for real-world situations.

In contrast, blind SR aims to recover images with unknown and complicated degradations. Existing techniques can be classified as explicit or implicit modelling based on the underlying degradation process. Detailed modelling approaches frequently utilize the conventional degradation model, including blur, downsampling, noise, and JPEG compression. However, real-world degradations need to be simplified to be described by a simple combination of these factors. Current research has focused on emulating a more practical degradation process [Wan21] or designing an improved generator [Wan19]. The performance of the discriminator, which guides the generator to generate superior images, has received relatively little attention. Therefore, it is essential to acknowledge the importance of the discriminator, much like a loss function.

In this paper, we intend to further enhance the perceptual quality of super-resolved images by extending the robust Real-ESRGAN [Wan21] algorithm:

- We utilize the Star Residual-in-Residual Dense Block (StarRRDB), which was inspired by ESRGAN+ [Rak20] and had a higher capacity than the RRDB employed by Real-ESRGAN [Wan21].
- We obtain a performance breakthrough for our SR models by combining the Multi-scale Attention U-Net Discriminator with the present StarRRDB-based generator inspired by A-ESRGAN [Wei21].
- We replaced the standard high-order Real-ESRGAN degradation model with a DASR-inspired [Lia22] efficient Adaptive Degradation Model.
- We use ResNet Loss in addition to VGG Loss, and this Dual Perceptual Loss approach, inspired by ESRGAN-DP [Son22], acquires more sophisticated perceptual characteristics.
- We attempt to apply Dropout Degradation technique, inspired by RDSR [Kon21] to improve the generalization ability from appropriate usage of dropout benefits.
- We construct StarSRGAN Lite, a CNN-oriented compact version that can reconstruct images about 7.5 times faster than StarSRGAN and Real-ESRGAN.

Due to several modifications, our StarSRGAN achieves higher visual performance than Real-ESRGAN, making it more applicable to real-world applications.

2 RELATED WORK

2.1 Super-Resolution Methods

In blind image SR problems, deep CNNs are frequently used before implementing the GAN architecture. These techniques have achieved a remarkable peak signal-to-noise ratio (PSNR) due to the robust modelling capability of CNNs. However, since these PSNR-based approaches use pixel-wise specified losses such as mean squared error (MSE), the output is often overly smoothed, necessitating additional high-frequency information. In practice, most methods assume a bicubic downsampling kernel, which may not be effective for real-world images. Furthermore, current studies aim to incorporate reinforcement learning or GAN before image restoration.

Blind SR has been widely researched, with numerous studies focusing on degradation prediction and conditional restoration. The two processes can be performed separately or in tandem, often iteratively. These techniques rely on predefined degradation models, which may only consider synthetic degradations and fail to perform well with real-world images. Furthermore, inaccurate degradation models can result in unwanted artifacts in the reconstructed images.

Recently, SOTA research has proposed a perceptually-driven approach to improve GANs by more accurately simulating the perceptual loss between images. For instance, ESRGAN [Wan19] and ESRGAN+ [Rak20] have introduced a viable perceptual loss function and generator networks based on RRDB that can convincingly create HR images. Another method, Real-ESRGAN [Wan21], has introduced a high-order degradation model to make even more realistic images, achieving impressive results on the NIQE benchmark. However, these methods depend on a computationally complex backbone network and cannot handle images with varying levels of degradation. Therefore, DASR [Lia22] has introduced a degradation-adaptive framework to address this issue, creating an effective and efficient network for real-world SR challenges.

Our work has incorporated several benefits from various designs to produce a comprehensive solution.

2.2 Degradation Models

Blind SR approaches often rely on the classical degradation model, which may not fully represent the complex degradation in real-world images. Recent technique, such as Real-ESRGAN [Wan21], incorporate a broader range of degradation types and parameters into the modelling process to address this issue. These approaches increase the model's ability to enhance the perceptual quality of challenging LR inputs. However, the sampling of degradation parameters in these methods can be imbalanced, which limits their ability to gen-

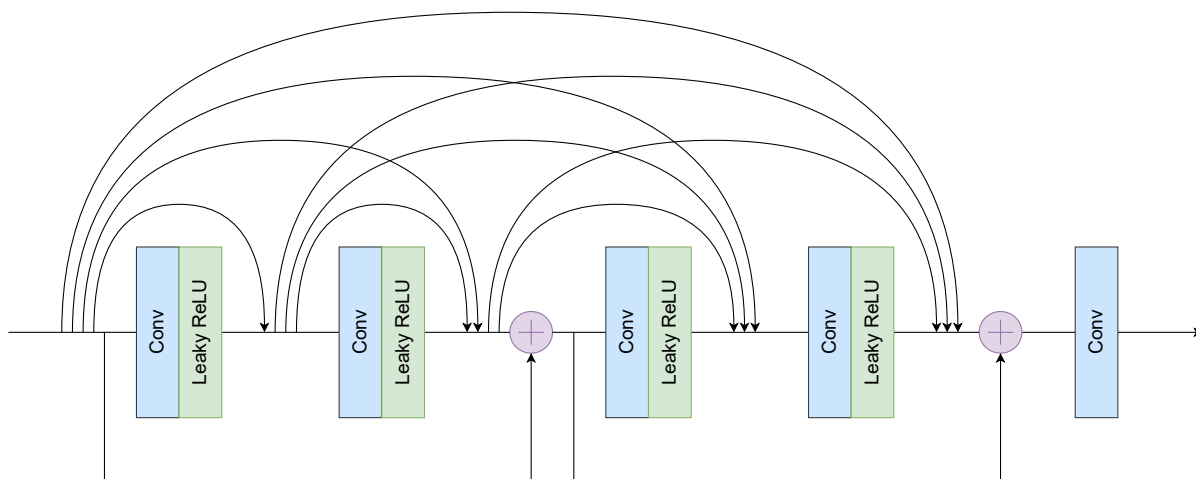


Figure 2: In Star Dense Block, residuals are added every two layers.

erate fine features, particularly for inputs with average degradations.

This study implements an adaptive degradation model that balances the degradation space by dividing it into three frequency-balanced levels. This balanced space optimises the model at different levels and provides a more accurate representation of real-world images.

2.3 Discriminator Models

There have been significant efforts to improve the discriminator model to synthesize high-quality HR images. Two critical challenges in achieving photo-realistic HR images are the need for a broad receptive field for the discriminator to distinguish between the synthesized and ground truth images, which requires either a deep neural network or a big convolution kernel. Additionally, a single discriminator may struggle to provide input on global and local characteristics, leading to potential incoherence in the synthesized image, such as distorted wall textures.

pix2pixHD [Wan17] proposed a unique multiple discriminator architecture to address these challenges. The first discriminator takes downsampled synthetic images as input, has a broader receptive field with fewer parameters, and focuses on comprehending the global perspective. The second discriminator uses the entire synthesized image to learn the image's specifics. Another study [Sch20] employed a U-Net-based discriminator architecture to GAN-based blind SR challenges, preserving the global coherence of synthesized images while offering per-pixel input to the generator.

Our discriminator model combines the advantages of both designs, enabling the discriminator to learn edge representations, enhance training stability, and provide per-pixel feedback to the generator.

2.4 Image Quality Assessment Methods

In recent years, GAN has been widely utilized for restoring low-quality images (e.g., deblur, denoise, super-resolution). Some researchers have focused on assessing images using Image Quality Assessment (IQA) methods. Some synthetic textures look natural due to the GAN approach, making them difficult for humans to distinguish yet easy for machines to detect.

Attention-based Hybrid Image Quality Assessment Network (AHIQ) [Lao22] aims to quantify the human perception of image quality and has the potential for generalizing unknown and complex samples, notably GAN-based distortions. AHIQ won first place in Full-Reference (FR) Track for the NTIRE2022 Perceptual Image Quality Assessment Challenge [Nti22].

Multi-dimension Attention Network for No-Reference Image Quality Assessment (MANIQA) [Yan22] uses the multi-dimensional attention network for perceptual assessment. MANIQA placed first in the No-Reference (NR) Track of the NTIRE2022 Perceptual Image Quality Assessment Challenge [Nti22].

Since no real-world GT exists for SR images, NR-IQA is preferable to FR-IQA for SR visual comparison. Nonetheless, in this study, we use both AHIQ and MANIQA, as well as some classical metrics like PSNR, SSIM, NIQE, and LPIPS. The objective is to determine if StarSRGAN models can achieve new performance levels in many measures.

2.5 Perceptual Loss Methods

Since the groundbreaking SRCNN [Don16] was presented, applying deep learning to tackle the SR problem has garnered increasing interest. In addition to a significant boost in visual quality, it also features a greater variety of optimizations and enhancements.

Further research [Joh16] claimed that smoothing the reconstructed image was as simple as improving the MSE

or PSNR of the pixel space ratio between the GT image and the reconstructed image. In order to enhance the reconstruction effect, the perceptual loss was suggested to minimize the feature space error between the GT image and the rebuilt image.

Based on the concept of SRGAN [Led17], ESRGAN [Wan19] employed a range of strategies to enhance the texture features further. In terms of perceptual loss, they recommended using the output before the activation of the convolutional layer to gain additional feature information, with the error of the feature space before activation being the object to be reduced.

Discussing perceptual loss is essential for enhancing the reconstruction outcomes, particularly the realism of details. This work uses a unique dual perceptual loss function as a combination of ResNet [Kai16] loss and VGG [Sim15] loss to achieve the reduction of unnatural artifacts produced by the perceptual-driven technique.

3 PROPOSED METHODS

3.1 Adaptive Degradation Model

Recently, a high-order degradation model has been proposed to generate LR images more closely approximate real-world conditions. The model executes the same degradation operation multiple times and has advanced from simple bicubic down-sampling to include shuffling and second-order pipelines.

In this research, we incorporate numerous image degradation procedures, including blurring (both isotropic and anisotropic Gaussian blur), resizing (both down-sampling and up-sampling with area, bilinear, and bicubic operations), noise corruption (both additive Gaussian and Poisson noise), and JPEG compression.

Inspired by the DASR [Lia22] approach, our architecture is designed to be adaptive to a wide range of real-world inputs and handle a subspace of images with different degradation levels. We divide the entire degradation space D into three levels: $[D_1, D_2, D_3]$, with one of these randomly chosen to produce LR-HR image pairs during training. The probability distribution for selecting the levels is $[0.3, 0.3, 0.4]$. D_1 and D_2 use first-order degradation with small and large parameter ranges, while D_3 uses second-order degradation. We use isotropic and anisotropic Gaussian kernels for the blur operation with a probability of $[0.65, 0.35]$. If an isotropic blur kernel is supplied, we set $\sigma_1 = \sigma_2$. In the second degradation stage of D_3 , we skip the blur operation with a 20% probability and use sinc kernel filtering with an 80% probability, following the approach used in Real-ESRGAN. We scale the image to the appropriate LR size, a quarter of its original size.

3.2 Network Architecture

StarSRGAN. The core block of ESRGAN enables the network to be very scalable and more straightforward

to train. The Star Dense Block we suggested is replacing the Dense Block to increase the network's capacity. Figure 2 depicts an extra level of residual learning within the Dense Blocks to expand the capacity without increasing complexity. After two layers, a residual is added to each block. This novel architecture produces images with improved perceptual quality by utilizing feature exploitation and exploration.

StarSRGAN Lite. The lightweight version of StarSRGAN focuses on delivering faster reconstruction times while maintaining acceptable visual quality. Like its predecessor, the model seeks to capitalize on the Multi-scale Discriminator, Attention U-Net Discriminator, Dual Perceptual Loss, Dropout Degradation, and Adaptive Degradation. ESPCN [Shi16] influences the network design, a super-resolution CNN which brings asymptotic real-time performance. Figure 3 depicts the architecture of StarSRGAN Lite.

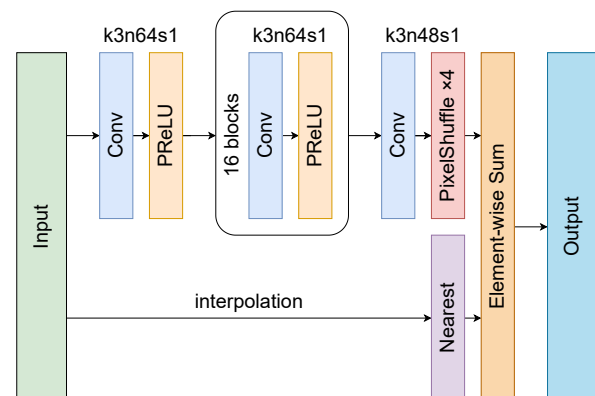


Figure 3: Architecture of StarSRGAN Lite with corresponding kernel size (k), number of filters (n), and stride (s) indicated for each convolutional layer.

3.3 Attention U-Net Discriminator

Taking inspiration from A-ESRGAN [Wei21], we have developed an Attention U-Net Discriminator structure, depicted in Figure 4, that aims to enhance the quality of the reconstructed image while increasing the efficiency of image reconstruction. The structure comprises a downsampling encoding module, an upsampling decoding module, and multiple Attention Blocks. The Attention and Concatenation Blocks are designed following the A-ESRGAN architecture. To perform semantic segmentation of 2D images, we adapted the Attention Gate, initially scheduled for 3D medical images as described in [Okt18]. Furthermore, we incorporated Spectral Normalization regularization [Miy18] to stabilize the training process.

3.4 Multi-scale Discriminator

StarSRGAN has a Multi-scale Discriminator architecture consisting of two identical U-Net discriminators.

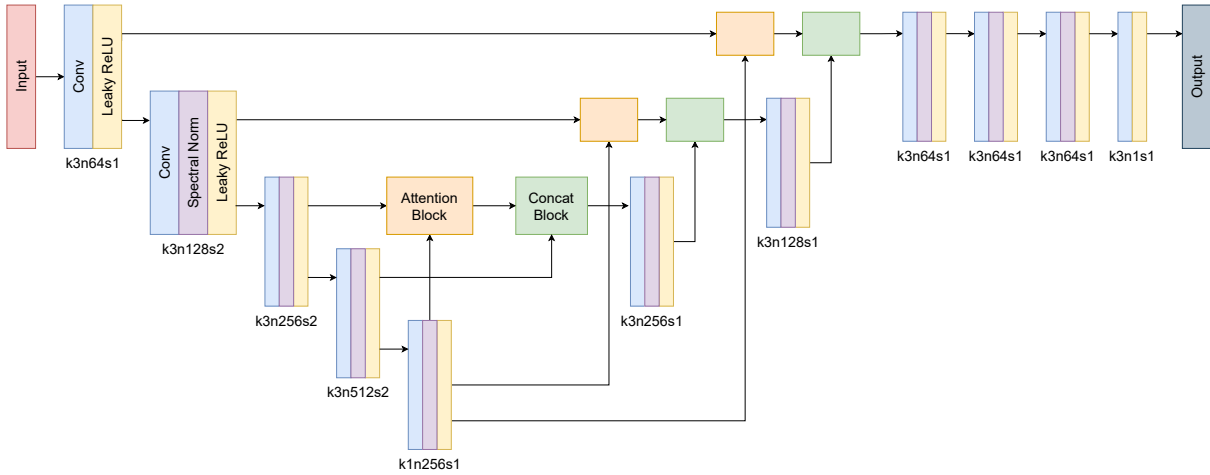


Figure 4: Architecture of Attention U-Net Discriminator with corresponding kernel size (k), number of feature maps (n), and stride (s) indicated for each convolutional layer.

The first discriminator D_1 accepts an original scale image as input, and the second discriminator D_2 accepts a $2\times$ downsampled image as input.

The output of the U-Net discriminator is a $W \times H$ matrix, with each member representing the probability that the pixel it represents is True. We utilize the Sigmoid function to normalize the output and the binary cross-entropy loss to determine the overall loss of one discriminator. Assuming C is the output matrix, we define $D = \sigma(C)$, x_r is 'real' data, and x_f is 'fake' data.

Consequently, we define the loss of one discriminator as

$$L_D = \sum_{w=1}^W \sum_{h=1}^H (-E_{x_r}[\log(D(x_r, x_f)[w, h])] - E_{x_f}[1 - \log(D(x_f, x_r)[w, h])]) \quad (1)$$

Because we have multi-scale discriminators, we will sum the loss of all the discriminators to get the overall loss as

$$L_{Total} = \lambda_1 L_{D_{normal}} + \lambda_2 L_{D_{sampled}} \quad (2)$$

where λ_1 and λ_2 are coefficients. We can also derive the generator loss from a single discriminator as

$$L_G = \sum_{w=1}^W \sum_{h=1}^H (-E_{x_r}[1 - \log(D(x_r, x_f)[w, h])] - E_{x_f}[\log(D(x_f, x_r)[w, h])]) \quad (3)$$

where x_f represents the output of the generator $G(x_i)$.

3.5 Dual Perceptual Loss

By training a deep neural network, we address the SR problem. According to the theory given by Dong et al. [Don16], the following is the goal of optimization:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^n L(G(I_i^{LR}; \theta), I_i^{HR}), \quad (4)$$

Where $I_i^{LR} \in \mathbf{R}^{C \times H \times W}$ and $I_i^{HR} \in \mathbf{R}^{C \times H \times W}$ represent the i -th LR and HR sub-image pairings, respectively, in the training set. $G(I_i^{LR}; \theta)$ is the representation of the up-sampling network. θ is the parameter to be optimized within the neural network. L is the loss function that can be represented as:

$$L = \lambda l_{content} + \eta l_{adversarial} + \gamma l_{percep} \quad (5)$$

where $l_{content}$ is the content loss of the pixel-wise 1-norm distance between images reconstructed by the generator and GT images, $l_{adversarial}$ is the loss derived from the mentioned Multi-scale Discriminator, l_{percep} is the Dual Perceptual Loss we implement. λ , η , and γ are the coefficients of balancing different loss terms.

SRGAN [Led17] proposed defining the VGG loss based on the ReLU activation layer of the pre-trained VGG-19 network. ESRGAN [Wan19] redefined the VGG loss after the convolutional layer and before the activation layer to gain additional feature information. This study uses the VGG loss specified in [Wan19], as the L1 Norm function is utilized to determine the Manhattan Distance between the reconstructed image features and the GT image features:

$$l_{VGG/i,j} = \frac{1}{C_{i,j} W_{i,j} H_{i,j}} \sum_{z=1}^{C_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} |\Phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y,z} - \Phi_{i,j}(I^{HR})_{x,y,z}| \quad (6)$$

where $\Phi_{i,j}$ represents features acquired by the j -th convolution (before activation) in the VGG network before the i -th max-pooling layer. In the VGG network, $C_{i,j}$, $W_{i,j}$, and $H_{i,j}$ are the dimensions of their respective feature spaces.

Based on the concepts of SRGAN [Led17], the ResNet loss is defined based on the ReLU activation layer of

Table 1: Quantitative comparison with SOTA methods on common test datasets using standard IQAs for SR (4×). LPIPS/NIQE ↓: the lower, the better. PSNR/SSIM/MANIQA/AHIQ ↑: the higher, the better. Note that MANIQA and AHIQ are the current SOTA IQA methods. The best and second performance are marked in **red** and **blue**.

Discussion: The unusual city textures on Urban100 are generally intricate and should consider further fine-tuning for particular usage. Models using real-world emulated data synthesis failed on Manga109 (Japanese comic) dataset, as prior predictions. Thanks to the Adaptive Degradation Model have helped StarSRGAN solve this problem partially. StarSRGAN has a slightly lower performance on Set14, which contains many drawing pictures.

Method	DIV2K						Set5						Set14					
	PSNR	SSIM	LPIPS	NIQE	MANIQA	AHIQ	PSNR	SSIM	LPIPS	NIQE	MANIQA	AHIQ	PSNR	SSIM	LPIPS	NIQE	MANIQA	AHIQ
Original HR	80.00	1.0000	0.0000	3.2342	0.6542	0.6289	80.00	1.0000	0.0000	3.2122	0.6338	0.6353	80.00	1.0000	0.0000	5.7318	0.5807	0.6402
SRGAN	25.71	0.7076	0.1808	3.0079	0.5256	0.4847	30.64	0.8361	0.1080	4.3067	0.5084	0.5359	19.94	0.4688	0.2566	2.5155	0.5106	0.4865
ESRGAN	24.94	0.6827	0.1471	3.1900	0.5616	0.4938	29.92	0.8258	0.1117	3.9433	0.5396	0.5461	18.60	0.4209	0.2039	2.3251	0.5329	0.5203
Real-ESRGAN	24.23	0.6646	0.2637	3.1461	0.5728	0.5112	26.06	0.7806	0.1901	3.6883	0.5794	0.5524	18.00	0.4186	0.3578	3.6056	0.5979	0.5233
A-ESRGAN	23.12	0.6032	0.3110	2.9722	0.5398	0.4217	23.95	0.6980	0.2060	2.8405	0.6046	0.4730	18.12	0.3476	0.4927	2.7213	0.6017	0.5245
FeMaSR	22.10	0.6178	0.2236	4.9289	0.5383	0.4426	24.22	0.7202	0.1943	4.4306	0.6101	0.5054	17.49	0.3872	0.3314	3.6338	0.5884	0.4968
Swin2SR	25.65	0.7164	0.4282	5.8883	0.4346	0.4319	28.53	0.8629	0.2855	6.9146	0.5123	0.5681	19.94	0.4756	0.2556	2.6782	0.5689	0.5307
StarSRGAN	25.53	0.7232	0.1365	3.2513	0.6151	0.5529	29.60	0.8576	0.0905	3.3224	0.6195	0.6160	19.22	0.5167	0.2035	3.2480	0.5932	0.5661
StarSRGAN Lite	24.36	0.6708	0.2588	4.3514	0.5016	0.4128	28.19	0.7995	0.1731	3.7280	0.5249	0.5309	17.91	0.4572	0.2780	4.6240	0.5284	0.5011

Method	BSD100						Urban100						Manga109					
	PSNR	SSIM	LPIPS	NIQE	MANIQA	AHIQ	PSNR	SSIM	LPIPS	NIQE	MANIQA	AHIQ	PSNR	SSIM	LPIPS	NIQE	MANIQA	AHIQ
Original HR	80.00	1.0000	0.0000	3.3586	0.8206	0.6761	80.00	1.0000	0.0000	1.8932	0.7743	0.6395	80.00	1.0000	0.0000	3.0512	0.7282	0.7055
SRGAN	20.39	0.6455	0.2679	2.8920	0.6461	0.3503	18.98	0.6557	0.1617	1.6539	0.6303	0.49381	22.58	0.6822	0.1398	2.4046	0.5991	0.5039
ESRGAN	20.26	0.6187	0.2391	2.5170	0.6684	0.3795	18.06	0.6223	0.1435	2.4639	0.6753	0.5204	22.70	0.6735	0.1239	2.4314	0.6266	0.5252
Real-ESRGAN	19.35	0.6127	0.3183	4.6826	0.6815	0.4029	17.75	0.5966	0.2205	4.9107	0.7068	0.4643	20.26	0.6262	0.2203	4.6752	0.6883	0.4366
A-ESRGAN	18.52	0.5729	0.3242	3.4259	0.6583	0.3374	17.77	0.5546	0.2643	1.9287	0.6461	0.4133	20.26	0.5567	0.2757	3.1648	0.6239	0.4398
FeMaSR	19.38	0.6044	0.3193	4.3513	0.6486	0.3586	16.78	0.5580	0.2274	4.2489	0.6234	0.4429	19.28	0.5960	0.2127	3.6849	0.5838	0.4522
Swin2SR	20.97	0.6577	0.4307	3.8591	0.6103	0.3986	20.01	0.6782	0.3591	4.6336	0.5821	0.4864	22.29	0.7138	0.2724	4.5726	0.5663	0.4616
StarSRGAN	20.35	0.6496	0.2302	2.8900	0.7127	0.4508	19.94	0.7167	0.1045	1.9064	0.7299	0.5538	24.61	0.8065	0.1456	3.0859	0.6943	0.5626
StarSRGAN Lite	20.36	0.5736	0.2912	3.1389	0.6124	0.3396	20.50	0.6736	0.2941	3.6585	0.6035	0.4790	22.46	0.7195	0.1919	3.1200	0.6128	0.4994

the 50-layer pre-trained ResNet network presented in [Kai16]. Since the ResNet network structure differs from the VGG network, each feature space is specified by a unique block. ResNet-50 architecture is divided into four stages, each containing several bottleneck layers. The extracted perceptual features use the output value of the bottleneck layer at each stage, and the ResNet loss can also be expressed as:

$$l_{RES/m,n} = \frac{1}{C_{m,n} W_{m,n} H_{m,n}} \sum_{z=1}^{C_{m,n}} \sum_{x=1}^{W_{m,n}} \sum_{y=1}^{H_{m,n}} \left| \beta_{m,n} (G_{\theta_G} (I^{LR}))_{x,y,z} - \beta_{m,n} (I^{HR})_{x,y,z} \right| \quad (7)$$

where $\beta_{m,n}$ represents features obtained by the n -th bottleneck layer (after activation) at the m -th stage. $C_{m,n}$, $W_{m,n}$, and $H_{m,n}$ are the dimensions of their respective feature spaces in the ResNet network.

Finally, the Dual Perceptual Loss l_{percep} function under the two perceptual losses is expressed as:

$$l_{DP} = l_{VGG} + \frac{1}{\mu} \zeta_{l_{VGG}, l_{RES}} l_{RES}, \quad (8)$$

where the ResNet loss l_{RES} is weighted dynamically and the weight value is $\frac{1}{\mu} \zeta_{l_{VGG}, l_{RES}}$, μ is a nonzero constant. The $\zeta_{l_{VGG}, l_{RES}}$ can be expressed as:

$$\zeta_{l_{VGG}, l_{RES}} = \frac{l_{VGG} + c}{l_{RES} + c} \quad (9)$$

Where c is a small positive constant when its job only prevents the denominator from becoming zero, therefore, $\frac{1}{\mu} \zeta_{l_{VGG}, l_{RES}}$ is only a value that fluctuates with the ratio of l_{VGG} to l_{RES} . Consequently, $\frac{1}{\mu} \zeta_{l_{VGG}, l_{RES}}$ is regarded as the weight value under the ResNet loss, which can only alter the update range of network parameters and not the update direction.

3.6 Dropout Degradation

In high-level vision tasks, dropout is intended to reduce the overfitting problem. However, it is rarely used in low-level vision tasks such as image SR. As a traditional regression problem, SR behaves differently for high-level tasks and is sensitive to the dropout process. RDSR [Kon21] dropout research demonstrates that appropriate dropout utilization benefits SR networks and improves generalizability. In our study, we employ this approach primarily for observational purposes.

In particular, we add the dropout layer before the final output layer. According to the results of our experiments, this technique improves network performance in a multi-degradation condition.

4 EXPERIMENTS

4.1 Implementation

To better compare the functionality of various mechanisms, including: Adaptive Degradation Model (Adapt-Deg), Attention U-Net Discriminator (Attn-Unet), Multi-scale Discriminator (Multi-Disc), and

Table 2: Upsampling benchmark and inference performance of Real-ESRGAN and StarSRGAN models.
System specification: Ubuntu 22.04 LTS, AMD Ryzen 5 5600X, NVIDIA GeForce RTX 3080 Ti, CUDA 12.1

Method	Upsampling Benchmark (FPS)										Performance			
	360p to 1080p		480p to 1440p		540p to 4K		720p to 5K		1080p to 8K		Image Quality		Inference Time	
	Python	C++	Python	C++	Python	C++	Python	C++	Python	C++	NR-IQA	FR-IQA	PyTorch	NCNN
Real-ESRGAN	2.85	6.94	1.63	3.97	1.29	3.14	0.73	1.78	0.30	0.73	100%	100%	100%	100%
StarSRGAN	2.59	6.42	1.56	3.81	1.24	3.02	0.69	1.72	0.28	0.69	107%	112%	94%	95%
StarSRGAN Lite	21.32	53.68	11.67	28.43	9.36	22.79	5.44	13.59	2.48	6.14	89%	93%	739%	753%

Table 3: Quantitative comparison of StarSRGAN variations on common test datasets for SR (average IQA).
LPIPS/NIQE ↓: the lower, the better. PSNR/SSIM/MANIQA/AHIQ ↑: the higher, the better. Note that MANIQA and AHIQ are the current SOTA IQA methods. The best and second performance are marked in **red** and **blue**.

Discussion: The dropout degradation technique needs to be more predictable when implement on SR models. Some models even have better NIQE than the HR. The IQA may not be accurately sufficient for blind SR tasks.

Method	Adapt-Deg	Attn-Unet	Multi-Disc	Dual-Loss	Drop-Out	PSNR	SSIM	LPIPS	NIQE	MANIQA	AHIQ
Original HR						80.00	1.0000	0.0000	3.4135	0.6986	0.6543
Real-ESRGAN						20.94	0.6166	0.2618	4.1181	0.6378	0.4818
StarSRGAN V1						23.43	0.6893	0.1361	2.5619	0.6320	0.5073
StarSRGAN V2	✓					24.56	0.7152	0.1220	2.7378	0.6422	0.5154
StarSRGAN V3	✓	✓				24.38	0.7065	0.1226	2.7281	0.6582	0.5442
StarSRGAN V4	✓	✓	✓			24.43	0.7185	0.1211	2.8669	0.6590	0.5333
StarSRGAN V5	✓	✓	✓	✓		23.21	0.7117	0.1518	2.9507	0.6608	0.5504
StarSRGAN+ V1					✓	23.12	0.6769	0.1361	2.6444	0.6368	0.5483
StarSRGAN+ V2	✓				✓	23.86	0.7144	0.1475	2.4271	0.6458	0.5681
StarSRGAN+ V3	✓	✓			✓	23.60	0.7035	0.1187	2.3328	0.6421	0.5705
StarSRGAN+ V4	✓	✓	✓		✓	23.60	0.7031	0.1206	2.3468	0.6403	0.5673
StarSRGAN+ V5	✓	✓	✓	✓	✓	24.75	0.7142	0.1255	2.2946	0.6444	0.5357
StarSRGAN Lite V1						19.97	0.5452	0.3684	4.0249	0.5317	0.3410
StarSRGAN Lite V2	✓					22.49	0.6524	0.2340	3.8435	0.5344	0.4287
StarSRGAN Lite V3	✓	✓				21.68	0.6385	0.2373	3.4966	0.5546	0.4489
StarSRGAN Lite V4	✓	✓	✓			22.08	0.6373	0.2350	3.6288	0.5578	0.4732
StarSRGAN Lite V5	✓	✓	✓	✓		22.30	0.6490	0.2479	3.7701	0.5639	0.4605
StarSRGAN Lite+ V1					✓	18.40	0.5688	0.2960	3.6561	0.5161	0.3842
StarSRGAN Lite+ V2	✓				✓	20.30	0.5609	0.2472	3.7053	0.5635	0.4518
StarSRGAN Lite+ V3	✓	✓			✓	18.73	0.5693	0.2491	3.9238	0.5614	0.4487
StarSRGAN Lite+ V4	✓	✓	✓		✓	19.65	0.5646	0.2363	4.9272	0.5813	0.4569
StarSRGAN Lite+ V5	✓	✓	✓	✓	✓	19.93	0.5714	0.2532	5.1208	0.5197	0.4221

Dual Perceptual Loss (Dual-Loss), we build 5 different StarSRGAN models corresponding models, including:

- StarSRGAN (V1): nearest similar to Real-ESRGAN but using the novel Star Dense Block.
- StarSRGAN V1 + Adapt-Deg (V2): using the Adaptive Degradation Model instead of the High-order Degradation Model.
- StarSRGAN V2 + Attn-Unet (V3): using the Attention U-Net Discriminator instead of U-Net Discriminator.
- StarSRGAN V3 + Multi-Disc (V4): using the Multi-scale Discriminator instead of Single Discriminator.
- StarSRGAN V4 + Dual-Loss (V5): using the Dual Perceptual Loss instead of Single Perceptual Loss.

Similarly, we also have 5 StarSRGAN Lite models (from V1 to V5), representing corresponding variations of StarSRGAN in compact architecture.

To observe the benefit of the dropout degradation (Drop-Out) technique, we also conducted a separate similar experiment with dropout layers, denoted by the plus sign (e.g. StarSRGAN+, StarSRGAN Lite+).

We trained 5 StarSRGAN, 5 StarSRGAN+, 5 StarSRGAN Lite, and 5 StarSRGAN Lite+ models on DIV2K [Nti17a] and Flickr2K [Nti17b] datasets. The training HR size is set to 256. We train our models on an NVIDIA A100 GPU with a batch size of 32 by using Adam optimizer. We train the StarSRNet models for 2000K iterations with a learning rate 2×10^{-4} while training the StarSRGAN models for 1000K iterations with a learning rate 1×10^{-4} . We also adopt the Exponential Moving Average (EMA) for more reg-

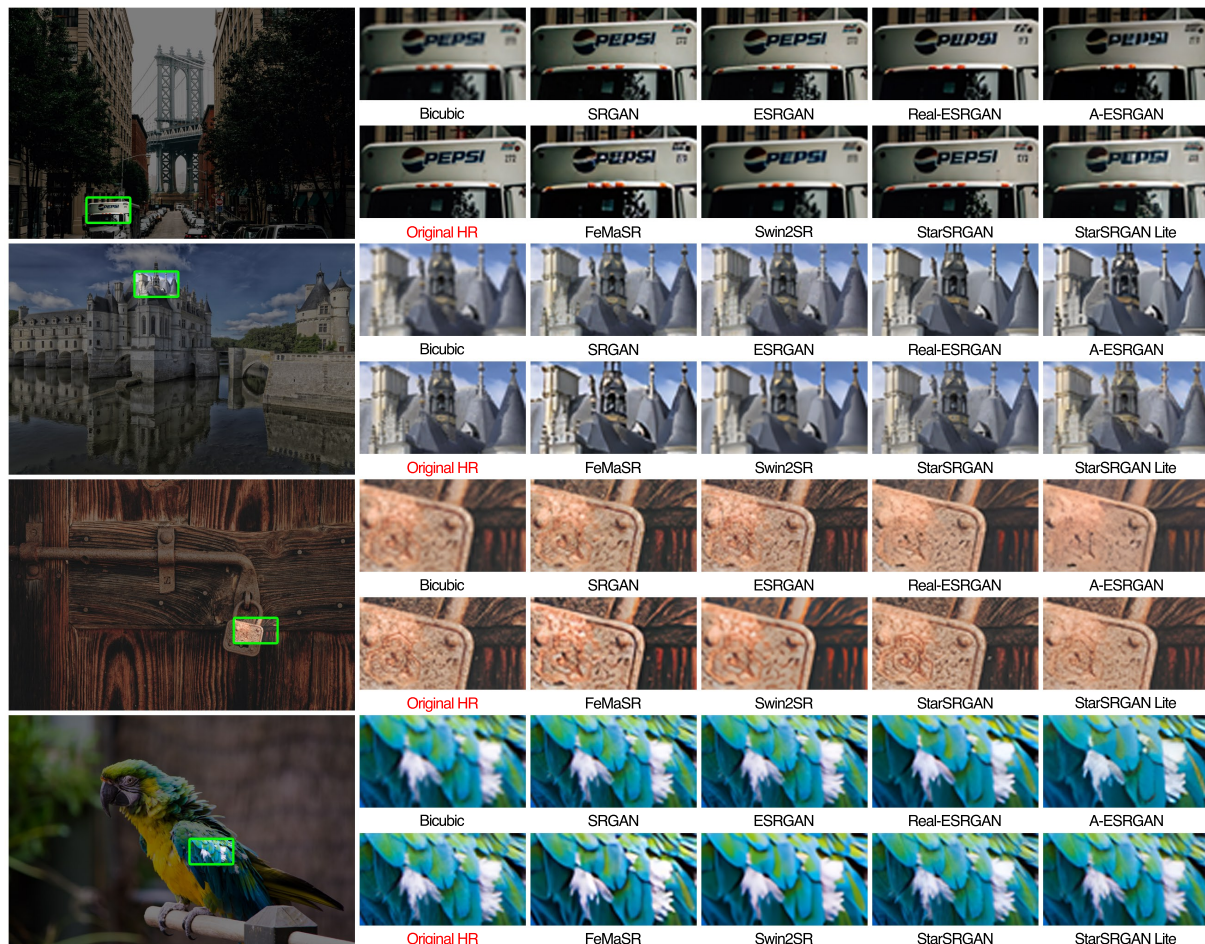


Figure 5: Visual comparisons of standard SR models with StarSRGAN models in 4 \times . (**Zoom in for best view**)

Discussion: The logo on the truck and the carving of the lock were recovered precisely on StarSRGAN. The image quality reconstructed by StarSRGAN Lite stays caught up with other models. All models deliver pleasant details and textures on the parrot feathers and the bell tower. Artifacts are also hard to be spotted on every model.

ular training and better performance. StarSRGAN and StarSRGAN Lite are trained with a combination of L1 Loss, Perceptual Loss, and GAN loss, with weights [1, 1, 0.1], respectively. Models which implement the Multi-scale Discriminator are composed of two discriminators, D_{normal} and $D_{sampled}$, which have the input of 1 \times and 2 \times down-sampled images as the input. The weight for GAN loss of D_{normal} and $D_{sampled}$ is [1, 1]. Our implementation is based on the BasicSR [Wan22].

4.2 Datasets

Previous studies have typically evaluated blind image SR models using synthetic LR images manually degraded from HR images. However, these images may not accurately represent the LR images resulting from real-world degradation processes, which often involve complex combinations of multiple degradation processes. Additionally, no publicly available datasets contain LR images from real-world sources. As an alternative, we have used real-world images scaled up by a factor of 4 for testing purposes. We employ

real-world images from 5 classical benchmarks to evaluate our approach, including Set5, Set14, BSD100, Urban100, Manga109, and the modern DIV2K Validation dataset [Nt17a]. These datasets contain images from diverse categories, such as portraits, landscapes, and structures. A reliable SR model should perform well on most of these standard datasets.

4.3 Compared Methods

We examine the proposed StarSRGAN and StarSRGAN Lite models with the SRGAN [Led17], ESRGAN [Wan19], Real-ESRGAN [Wan21], A-ESRGAN [Wei21], FeMaSR [Che22], and Swin2SR [Con22] models. The architecture of StarSRGAN V1 is the nearest similar to the architecture of Real-ESRGAN. More specifically, Residual Dense Block has been replaced with the novel Star Residual Dense Block, which can help evaluate the effectiveness of StarSRGAN even with a slight modification. On the other hand, StarSRGAN Lite models aim to reduce the reconstruction time of super-resolution. Therefore,

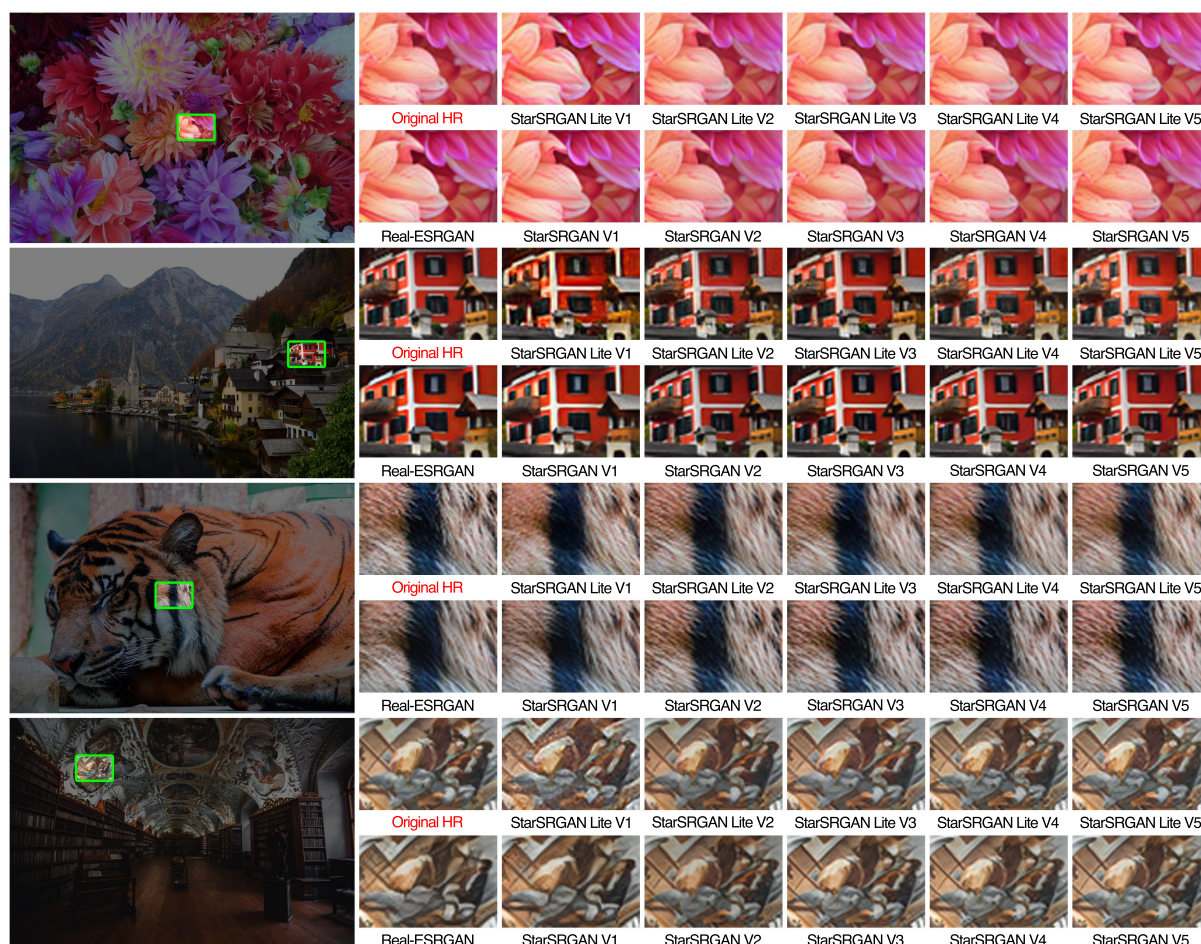


Figure 6: Visual comparisons of StarSRGAN variations with an upsampling scale of $4\times$. (**Zoom in for best view**)

Discussion: If we pay close attention, we can observe that the StarSRGAN V5 delivered more tiny details on the petals. The chimneys and the windows also look more similar to the HR. Every model reconstructed the fur of the tiger with outstanding quality. The coat color was also better restored on StarSRGAN V5 than on StarSRGAN V1 or Real-ESRGAN. The images enhanced by StarSRGAN Lite models are acceptable compared to its predecessor.

in addition to comparing perceptual quality, we also compare reconstruction time in frame rate (expressed in frames per second or FPS) between the Real-ESRGAN and StarSRGAN, and StarSRGAN Lite models.

4.4 Experiment Results

Table 1 compares StarSRGAN with other SR models on several standard test datasets for SR. The results indicate that the classical IQAs like PSNR, SSIM, and NIQE are no more suitable for SR evaluation. Previous research claims that to achieve high PSNR, the model tends to generate over-smooth results. The baseline model, SRGAN, usually obtains a better SSIM index than more advanced models. Moreover, some models even have a better NIQE score than the HR, making this measure the most unpredictable IQA in this work. On the other hand, the measurements that come from MANIQA and AHQ are anticipated and reasonable.

Models with real-world emulated data synthesis perform poorly on illustration. Fortunately, StarSRGAN

with Adaptive Degradation Model has partially solved this issue. Unnatural city-featured textures like windows, roads, and bricks tend to be more complicated to reconstruct than other textures. Fine-tuning models on extra training data is promising to sort out the problem.

Through upsampling benchmark results shown in Table 2, we can find that StarSRGAN has traded off its inference time to achieve better image quality compared to Real-ESRGAN. In the opposite direction, StarSRGAN Lite sacrifices its image quality to gain impressive performance in reconstructed time. Specifically, the lightweight architecture has brought real-time performance with more than 20 FPS when upscaling from 540p to 4K with C++ optimized executable file.

Table 3 shows that even our most straightforward variation, StarSRGAN V1, outperforms the Real-ESRGAN method in most metrics. Variations applied Dropout Degradation technique are not steady enough, and further research should be conducted. From visual com-

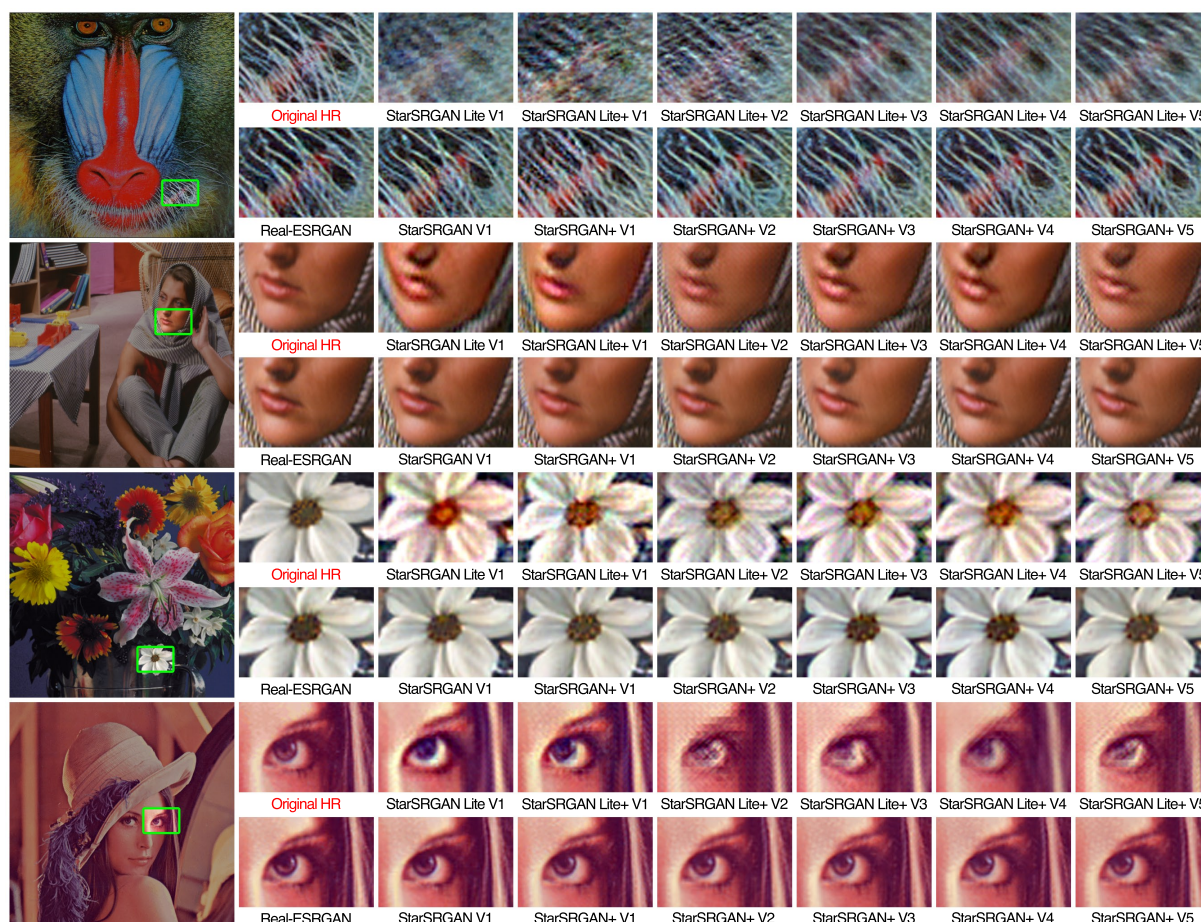


Figure 7: Further observation of StarSRGAN+ models applying Dropout technique. (**Zoom in for best view**)

Discussion: We can easily observe that Dropout Degradation brings unstable performance on StarSRGAN Lite models. On the contrary, the technique adequately integrated with StarSRGAN without apparent conflicts.

parison (some examples are shown in Figure 6 and Figure 7), we observe that our methods can recover sharper edges and restore better texture details. Although, it is hard for human-eye to distinguish because both Real-ESRGAN and StarSRGAN bring excellent perceptual quality. Note that the StarSRGAN V5 models have been selected for comparison in Table 1 and Table 2.

5 CONCLUSIONS

The present study introduces two novel GAN-based models, StarSRGAN and StarSRGAN Lite, for blind SR tasks. StarSRGAN integrates advancements from 5 previous research works and yields new SOTA performance, surpassing the leading SR method, Real-ESRGAN, by 10% on both SOTA No-Reference IQA and Full-Reference IQA methods (MANIQA and AHIQ). StarSRGAN Lite, a lightweight version of StarSRGAN, also inherits improvements from its predecessor and offers real-time inference performance, processing upsampled frames from 540p to 4k at over 20 FPS when executed on a C++ optimized executable file. Several directions for further enhancement of

StarSRGAN architectures are recommended. For instance, retraining the models with newly released datasets such as DIV8K or Unsplash. Additionally, other activation functions such as SiLU and GELU, could be a better alternative for the familiar ReLU. Super-resolving only interested objects and disregarding unnecessary regions like the background could improve StarSRGAN inference performance. Applying Video SR techniques and leveraging spatiotemporal data could also be a promising direction for further research. Currently, StarSRGAN models support only the 4× upscale factor, and other upscale factors such as 2×, 8×, and 16× are also necessary. Another approach is employing an image classifier to distinguish between real-life and unreal images and choosing the most optimized for each case. Better batch size and more iterations could be explored with more robust hardware. In conclusion, these directions could facilitate the development of future research.

Acknowledgement. This research is funded by University of Science, VNU-HCM project CNTT 2023-08.

6 REFERENCES

- [Don16] Dong, C. et al. (2016) "Image Super-Resolution Using Deep Convolutional Networks", IEEE TPAMI 2016, 38(2), pp. 295-307. doi: 10.1109/TPAMI.2015.2439281.
- [Joh16] Johnson, J. et al. (2016) "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016, pp. 694-711. doi: 10.1007/978-3-319-46475-6_43
- [Led17] Ledig, C. et al. (2017) "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", IEEE/CVF CVPR 2017, pp. 105-114. doi: 10.1109/CVPR.2017.19.
- [Wan19] Wang, X. et al. (2019) "ESRGAN: Enhanced super-resolution generative adversarial networks", Springer 2019, pp. 63-79. doi: 10.1007/978-3-030-11021-5_5.
- [Goo14] Goodfellow, I. et al. (2014) "Generative Adversarial Networks", NeurIPS 2014, 3. doi: 10.1145/3422622.
- [Wan21] Wang, X. et al. (2021) "Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data", IEEE/CVF IC-CVW 2021, pp. 1905-1914. doi: 10.1109/IC-CVW54120.2021.00217.
- [Rak20] Rakotonirina, N. C. and Rasoanaivo, A. (2020) "ESRGAN+: Further Improving Enhanced Super-Resolution Generative Adversarial Network", IEEE ICASSP 2020, pp. 3637-3641. doi: 10.1109/ICASSP40776.2020.9054071.
- [Wei21] Wei, Z. et al. (2021) "A-ESRGAN: Training Real-World Blind Super-Resolution with Attention U-Net Discriminators" arXiv 2021 [eess.IV]. doi: 10.48550/arXiv.2112.10046.
- [Lia22] Liang, J. Zeng, H. and Zhang, L. (2022) "Efficient And Degradation-Adaptive Network For Real-World Image Super-Resolution", ECCV 2022, pp. 574-591. doi: 10.1007/978-3-031-19797-0_33.
- [Son22] Song, J. et al. (2022) "Dual Perceptual Loss for Single Image Super-Resolution Using ESRGAN", arXiv 2022 [eess.IV]. doi: 10.48550/arXiv.2201.06383.
- [Sch20] Schonfeld, E., et al. (2020) "A U-Net Based Discriminator for Generative Adversarial Networks", arXiv 2020 [cs.CV]. doi: 10.48550/arXiv.2002.12655.
- [Wan17] Wang, T. et al. (2017) "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs", arXiv 2017 [cs.CV]. doi: 10.48550/arXiv.1711.11585.
- [Yan22] Yang, S. et al. (2022) "MANIQA: Multi-Dimension Attention Network for No-Reference Image Quality Assessment", arXiv 2022 [cs.CV]. doi: 10.48550/arXiv.2204.08958.
- [Lao22] Lao, S. et al. (2022) "Attentions Help CNNs See Better: Attention-Based Hybrid Image Quality Assessment Network" arXiv 2022 [cs.CV]. doi: 10.48550/arXiv.2204.10485.
- [Nti22] Gu, J. et al. (2022) "NTIRE 2022 Challenge on Perceptual Image Quality Assessment", IEEE/CVF CVPRW 2022, pp. 950-966. doi: 10.1109/CVPRW56347.2022.00109.
- [Kai16] He, K. et al. (2016) "Deep Residual Learning for Image Recognition", IEEE CVPR 2016, pp. 770-778. doi: 10.1109/CVPR.2016.90.
- [Sim15] Simonyan, K. and Zisserman, A. (2015) "Very deep convolutional networks for large-scale image recognition", ICLR 2015, pp. 1-14. doi: 10.48550/arXiv.1409.1556.
- [Shi16] Shi, W. et al. (2016) "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network", IEEE CVPR 2016, pp. 1874-1883. doi: 10.1109/CVPR.2016.207.
- [Nti17a] Agustsson, E. and Timofte, R. (2017) "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study", IEEE CVPRW 2017, pp. 1122-1131. doi: 10.1109/CVPRW.2017.150.
- [Nti17b] Timofte, R. et al. (2017) "NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results", IEEE CVPRW 2017, pp. 1110-1121. doi: 10.1109/CVPRW.2017.149.
- [Wan22] Wang, X. et al. (2022) "BasicSR: Open Source Image and Video Restoration Toolbox". GitHub 2022. <https://github.com/XPiPixelGroup/BasicSR>.
- [Okt18] Oktay, O. et al. (2018) "Attention U-Net: Learning Where to Look for the Pancreas" arXiv 2018 [cs.CV]. doi: 10.48550/arXiv.1804.03999.
- [Miy18] Miyato, T. et al. (2018) "Spectral Normalization for Generative Adversarial Networks", ICLR 2018. doi: 10.48550/arXiv.1802.05957.
- [Kon21] Kong, X. et al. (2021) "Reflash Dropout in Image Super-Resolution." arXiv 2021 [cs.CV]. doi: 10.48550/arXiv.2112.12089.
- [Che22] Chen, C. et al. (2022) "Real-World Blind Super-Resolution via Feature Matching with Implicit High-Resolution Priors", ACM 2022, pp. 1329-1338. doi: 10.1145/3503161.3547833.
- [Con22] Conde, M. V. et al. (2023) "Swin2SR: SwinV2 Transformer for Compressed Image Super-Resolution and Restoration", Springer 2023, pp. 669-687. doi: 10.1007/978-3-031-25063-7_42

Modeling and Rendering with eXpressive B-Spline Curves

Hock Soon Seah
Nanyang Technological
University
ashsseah@ntu.edu.sg

Budianto Tandianus
Singapore Institute of
Technology
budianto.tandianus@singaporetech.edu.sg

Yiliang Sui
Nanyang Technological
University
yiliang.sui@ntu.edu.sg

Zhongke Wu
Beijing Normal
University
zwu@bnu.edu.cn

Zhuyan Zhang
Nanyang Technological
University
d190004@e.ntu.edu.sg

ABSTRACT

eXpressive B-Spline Curve (XBSC) is a resolution-independent and computationally efficient technique for vector-based stroke modeling and rendering with the flexibility in defining and adjusting the shape and other parameters of the stroke. It generalizes the existing Disk B-Spline Curve (DBSC) geometric representation, which itself is a generalization of the Disk Bézier curve. XBSC allows flexible shape and color manipulation and rendering of strokes with asymmetrical shape control and rich color management. These properties make XBSC suitable for modeling freeform stroke shapes and animation, specifically in squash and stretch, a common technique to bestow elasticity and flexibility in shape changes. During the squash and stretch animation computation, we constrain the shape of the XBSC stroke to conserve its area. To achieve this, we apply the simulated annealing algorithm to iteratively adjust the XBSC while maintaining its area. We show several drawings, rendering and deformation examples to demonstrate the robustness of XBSC.

Keywords

XBSC, DBSC, B-spline, Vector Graphics, Diffusion Curve, Deformation, Simulated Annealing, Computer Animation.

1. INTRODUCTION

A vector-based stroke means outlining a shape with some line thickness and color. Such stroke is resolution-independent and can be scaled without losing image quality. An example is the Disk B-Spline Curve (DBSC) [Sea05a, Wu21a], which enables varying thickness on a B-Spline curve by storing a radius parameter at each control point. For example, in Fig 1, each stroke in the Chinese calligraphy, painting, and portrait can be represented by a single DBSC. With conventional representations such as B-Spline and line segment, the strokes must be defined using several discrete lines or polygonal approximation to form the close regions. Modifying such a close region would not be as simple as modifying a DBSC stroke. However, DBSC is symmetrical about its skeleton

and has only one stroke cap style (i.e., semi-circle). XBSC addresses these limitations by extending the DBSC representation in both shape modeling and color management.



Figure 1. Chinese calligraphy (left), Chinese painting (center), Portrait (right). Image from [Wu21a].

2. RELATED WORK

Several stroke representations, such as parametric curves [Sio90a, Pud94a], elliptical arcs [Com15a], line segments [Str86a], and raster image [Whi83a], have been proposed. These representations allow translation from physical to digital medium. However, the abovementioned representations have their respective disadvantages. Raster image representation is resolution-dependent, which results in large amount of data and is not editable.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Boundary-based method may cause mismatch between the upper and the lower boundary curves. Hsu's method requires complex calculation of the offset distances. Interval B-spline equation is only C^0 continuous causing difficulty in studying differential properties. To represent a non-zero-width stroke shape, typically existing solutions (including Poisson Vector Graphics [Hou18a] and Generalized Diffusion Curves [Orz13a, Jes16a]) represent its shape with multiple curves that form the stroke region.

To address these shortcomings, DBSC was proposed. DBSC is a culmination of previous works in stroke drawing based on B-Spline representation. For example, stroke representation with a centerline and thickness [Hsu94a], using a B-spline as stroke skeleton [Pha91a], and interval B-spline as strokes [Su02a]. DBSC itself has also undergone development over the years. For example, intersections between DBSCs [Ao18a], brush shape modeling and in between drawing generation using DBSC [Sea05a, Sea05b]. Owing to its properties, DBSC has also been used to draw digital Chinese calligraphy [Wan16a, Fu16a].

Based on the DBSC formulation, a DBSC has symmetrical shape along its skeleton, which limits its capability in modeling a freeform shape. Thus, XBSC, which is a generalization of DBSC by enabling asymmetrical shape, was proposed [Sea22a]. Owing to its properties, XBSC allows flexible shape and color editing, compact stroke representation, asymmetrical shape control, and exciting color management. As a result, XBSC is suitable for drawing and representing freeform shapes. Fig. 2 demonstrates the differences in modeling and rendering between DBSC and XBSC. Using the same B-spline skeleton, XBSC can produce more wide-ranging shapes compared to DBSC. DBSC would require the artist to shape the skeleton in a complex manner to achieve the same drawing as XBSC.

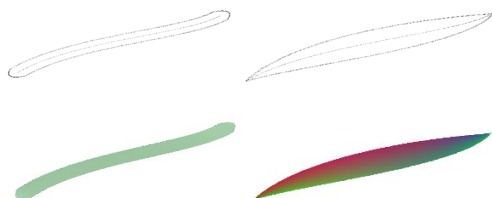


Figure 2. DBSC with symmetrical outline and uniform coloring (left) and XBSC with non-symmetrical outline and diffused coloring (right).

Shape deformation is an important topic in computer graphics [Gon13a]. Moreover, the conservation of area during deformation is

necessary to simulate realistic elastic deformations. Shape deformation is typically an expensive calculation. We investigated different deformation techniques in both 3D [Por03a] and 2D [Che98a] and decided to advance a step further with area/volume conservation along with deformation. Diziol et al. [Diz11a] proposed a volume-conserving deformation method. However, it is not intuitive to use it in an interactive application as it is not possible to specify deformation constraints.

Compared to our preliminary paper on XBSC [Sea22a], the main contributions in this paper are the incorporation of color diffusion, stroke deformation with area preservation, and user interaction design to draw and manipulate an XBSC stroke.

3. PROPOSED SOLUTIONS: EXPRESSIVE B-SPLINE CURVES (XBSC)

Definition

The basic XBSC curve enables users to change the radii and color defined at each control point. It also enables user to change the shape of its end sections. The XBSC equation is defined in the following equation:

$$X(t) = \begin{cases} h_X^1(t), & t < k_0 \\ \sum_{i=0}^n N_{i,p}(t) \mathbf{P}_i, & k_0 \leq t \leq k_m \\ h_X^2(t), & t > k_m \end{cases} \quad (1)$$

where $N_{i,p}$ is a B-spline basis function with degree p and \mathbf{P}_i is the i -th control point. $\mathbf{P}_i = \langle \mathbf{x}_i; r_i^1; r_i^2; \mathbf{c}_i^1; \mathbf{c}_i^2; \varphi \rangle$ comprises \mathbf{x}_i which are the spatial coordinates of the control point, r_i^1 and r_i^2 which are the radii of the two sides of the XBSC skeleton, \mathbf{c}_i^1 and \mathbf{c}_i^2 which are the colors of the two sides of the XBSC, and φ which is a color function used to control the transition between colors. Its default is linear and can be changed to radial. We define $h_X^1(t)$ and $h_X^2(t)$ to be the cap functions for both stroke ends (with the parameter t before and after the knot values k_0 and k_m).

Details

Referring to Eqn. 1, assume that $N_{i,p}(t)$ is the i -th item of the basis function of a B-spline curve whose degree is p , and knot vector $\mathbf{T} = \{t_0, t_1, \dots, t_m\}$, then $m = n + p + 1$. When an XBSC is drawn with $n + 1$ control points with a specified degree p , the knot vector \mathbf{T} is automatically computed. By default, $p = 3$. Fig. 3 shows an XBSC whose radii of the two sides of the XBSC skeleton are defined independently and two

different end caps. This allows wider flexibility in shape drawing compared to DBSC such as brush and ink pen drawings.

In terms of stroke cap, XBSC allows other stroke cap shapes beyond the semi-circle cap as defined in DBSC. This enables simulating various drawing styles as shown in Fig. 4. As for the color, XBSC uses Diffusion Curve (DC) [Orz13a] which allows more expressive coloring compared to DBSC.

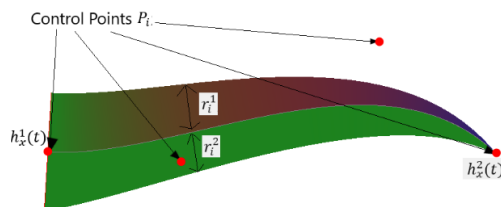


Figure 3. An XBSC with four control points P_i , $i = 0..3$, with radii r_i^1 and r_i^2 respectively, and end caps $h_x^1(t)$ and $h_x^2(t)$.

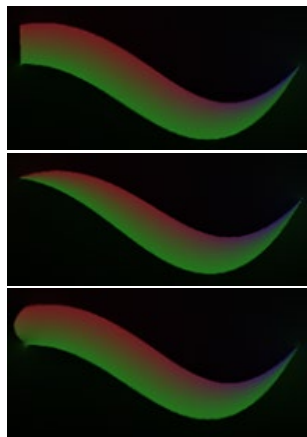


Figure 4. Different caps at the left side of a same stroke. Flat (top), Pointed (middle), Circular (bottom).

Computation of color dispersion in DC is based on a physics calculation technique used to calculate an electrical potential field. DC calculates color potential field instead of calculating electrical potential field by solving Poisson equations. In DC, thin curves serve as boundary values and the color for both sides of the curves are transformed to Laplacian values in the Poisson equations. In XBSC, the envelopes are considered as DCs and color of both sides of DCs correspond to external (e.g., canvas) and internal (within stroke) colors.

Using the RGB format, we calculate the dispersion separately for each color channel by using diffusion. The RGB values are converted into vectors, and we represented the partial derivatives of the Poisson equation in matrix form:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (2)$$

with \mathbf{b} being the divergence of the color gradient of each pixel in the XBSC image, \mathbf{A} is a diagonal matrix representing the partial derivative coefficient, and \mathbf{x} is the color of each pixel to be solved, which will give the color for every pixel throughout the canvas and stroke.

In Fig 4, the internal colors are the mix of red and green, with the black color as external color to match the canvas background. The internal colors determine the colors within the XBSC stroke. The external or outer colors are used to match colors outside of the XBSC stroke. The outer colors are used to match the canvas or background color, it can also be used when drawing multiple XBSC strokes next to each other to make sure the colors blend properly. Finally, the outer color is used when one draws XBSC stroke within another XBSC stroke.

DBSC can be applied to animation by keyframing its control point properties such as position and radius. Thus, in the user interface, the user adjusts the control points directly in each keyframe. An animation aspect that has not been tackled in DBSC is constraint-based deformation. In this paper, we apply deformation on the strokes with constraint on its area and it is generally known as squash and stretch. Other constraints, such as fixing a particular control point, is possible. To realize the deformation, we use simulated annealing [Kir83a] as the optimization algorithm. Given an initial stroke configuration, the user adjusts its skeleton and recompute the XBSC radii by using simulated annealing. The user may also constrain some radii. Hence, the constraint in the optimization is the skeleton shape and radii of some control points, and the outputs are the radii of the other control points that are not set as constraints.

User Interactions

To illustrate the ease of drawing and expressiveness of XBSC strokes, we designed the user interactions and develop a visualization tool, which we refer to as eXpressiveDrawing. Fig. 5 shows the step-by-step process of creating an XBSC stroke, which has a default degree of 3. The number of control points can be any value with a minimum of 4. The user first draws the required number of control points, followed by deciding the radii and colors to use for each control points. Finally, the caps at both ends of the stroke are decided by the user. Each end cap is an XBSC, and its shape can be either a point, flat line, semi-circle, or general shape. Some of these end caps are shown in Fig 6.

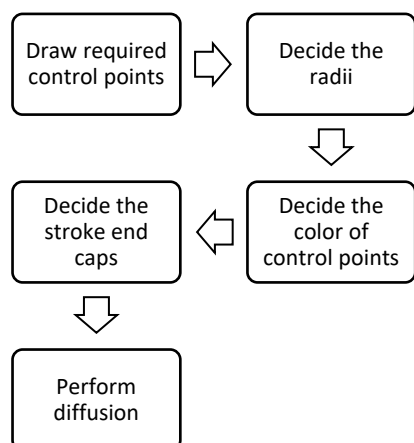


Figure 5. Steps to drawing an XBSC.

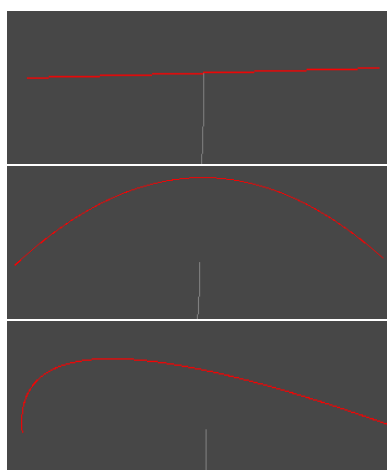


Figure 6. The red color lines are the end caps, the white lines are the XBSC skeleton. A flat end cap (top), a semi-circle cap (middle), a general shape cap (bottom).

As shown in Fig. 7, a window will pop up at the cursor location where a control point position is to be added. The window allows a user to change the key attributes of the control point. As the user changes the colors and radii, they will be visually reflected as colored circles, depicting intuitive changes in both colors and radii at the same time. If the user decides not to add the current control point to the current curve, he will click the “Cancel” button. Additionally, color and radius attributes are usually similar between neighboring control points. As a result, the design of the pop-up window in eXpressiveDrawing adheres to a scheme where the initial value of the attributes will be the same as the last confirmed control point.

At each control point, there will be two other colors which control the color of outer sides of an XBSC for diffusion computation. Since it is less frequently used, it is defaulted to null. If needed, a

smaller color picker button is implemented to open another window for color selection. Keeping the narrative for the base case to a minimum lead to clean and simple operation of the user interface under normal circumstances.

In eXpressiveDrawing, a single right click will group all ungrouped control points in the canvas into a single curve. In Fig. 8, the outlines of a stroke are formed from four control points. The white color curve in the middle represents the central skeleton of the stroke. The order of the control point in the curve depends on the order that the user draws the control point. For example, the last control point created by the user right before the right click will be the end control point of the XBSC.

The two concentric circles around the control points represent the color and size for each envelop of the XBSC curve. The user can drag the circles to decide the width of the envelop and change the colors using the interface shown in Fig 7.

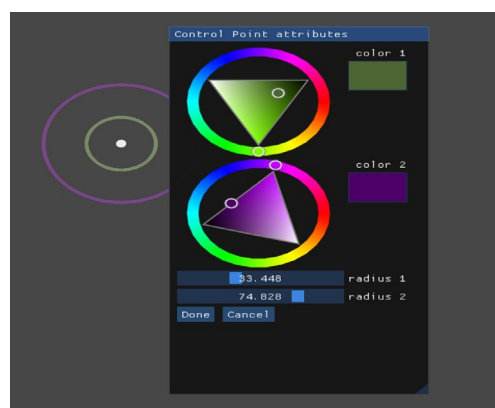


Figure 7. Interface for adding a control point in eXpressiveDrawing. At the left, the white dot depicts the position of the control point, the two other circles depict the colors and radii of the two sides of the XBSC. A pop-up window in the middle allows user to set the colors and the radii.

Fig. 8 shows the results of four control points with different concentric circles forming an XBSC accordingly. Furthermore, eXpressiveDrawing is designed to be interactive. When the user changes the properties of control points like the color and radius, the changes will be reflected instantly on the outline of the stroke.

Many artists use existing images as reference or inspiration when creating their own drawings or paintings. This can help them to accurately depict certain elements or details, or to capture a certain mood or atmosphere. In the window menu shown in Fig. 9, eXpressiveDrawing provides users with

the functionality of loading an image file as a background. The user can then freely create a stroke outline on top of the background image.

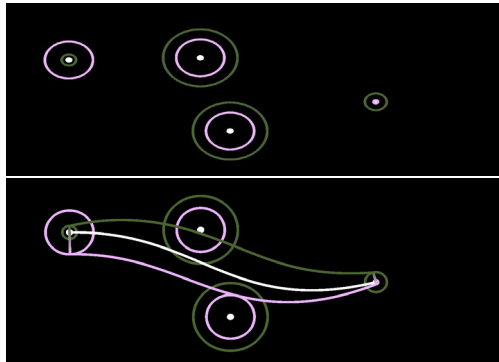


Figure 8. Four control points (top) and an XBSC curve formed from the control points (bottom). The white line depicts the skeleton, and the green and purple lines depict the envelopes of the XBSC.



Figure 9. Tracing the outline of an eye based on background image, which was taken from [Orz13a].

4. EXPERIMENTAL RESULTS

Shape Modeling and Rendering

With XBSC, we can create a variety of shapes using the two radii on both sides of the skeletal B-spline curve. The color also follows the same principle and formula as the radii resulting in a gradual change and flow of color. The XBSC allows colors to diffuse along the curve. Adding diffusion properties allows the color to blend completely without the clear divide in the middle.

We recreated the eye example in Fig. 9 from [Orz13a] to illustrate the ease of drawing with XBSC. The skeletons of the XBSC lines are shown in Fig. 10 (top). The eyelid, the eyeball and the white of the eye are each created using an XBSC stroke. The color is defined on each control point of the skeleton, after diffusion, the image of the eye is rendered as shown in Fig. 10 (bottom).

Fig. 11 shows an example of a drawing with 5 XBSCs to form an apple.

Shape Deformation

The extended capabilities of the XBSC to control the radii of the shapes drawn allows us to perform Shape Deformation. By altering the radii and moving the control points, the size and shape of any drawings can be quickly altered from one form to another without any constraints. However, to create realistic deformation, a user may want to constrain the shape of the deformation to preserve the area of the shapes during the deformation process. Since this shape deformation is done by changing the radii of the control points, it can only deform a single XBSC stroke to preserve the area. As the area is computed for a single XBSC stroke, multiple XBSC strokes need to be deformed separately.

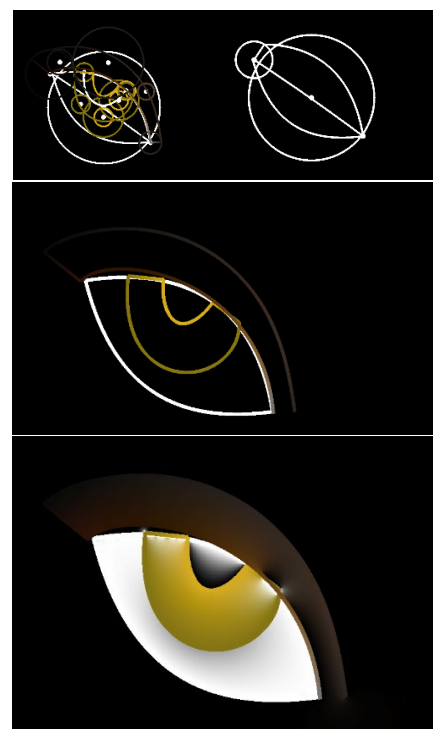


Figure 10. Zoomed-in view of XBSC skeletons of an eye drawing with control points (top), the drawing before diffusion (middle) and the final rendered result (bottom).

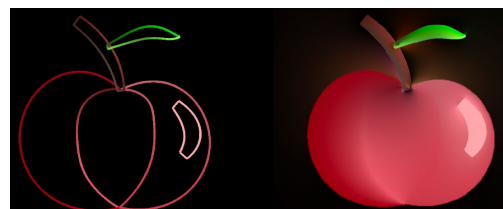


Figure 11. Five XBSCs form an apple outline (left) and rendered (right).

To calculate the surface area, we tessellate the XBSC into a triangle mesh, see Fig. 12. We compute the area of each triangle using cross product of two of its edge vectors. The area of the triangle mesh is obtained by summing the areas of all the triangles in the mesh. Using the same method, we also calculated the areas of the end caps of the XBSC.

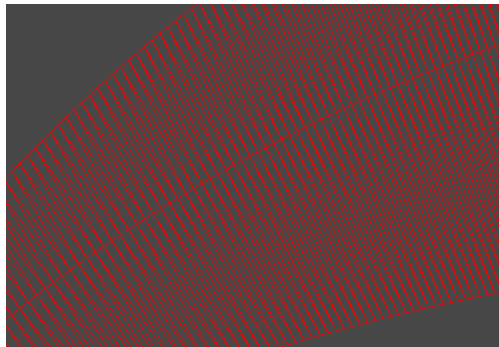


Figure 12. A section of a triangulated XBSC. Top and bottom curves are the envelopes and central red curve is the skeleton of the XBSC.

With the area calculated, by using a simulated annealing optimization formula, we iteratively enable the XBSC to change the radii to obtain a new area to approximate the previous area. The previous area and the new area are saved, and the user can decide which control points should not be changed through the entire iteration of the formula. This provides user with some control over the resulting shapes. The users can decide the number of iterations, the rate of change for the radii and which radii to remain unchanged. As it is impossible for the user to determine the exact changes for each radius, a random value that ranges from 0.0 to 1.0 will be chosen and will multiply the rate of change previously determined by the user. These values will be added or deducted from the radii. The area will be recalculated using these new radii and compared to its previous area. Using the acceptance probabilities of simulated annealing and depending on how close this new area is compared to the previous area, the program will either keep these radii or revert to the previous radii values. With each iteration, the changes in the radii will alter the area of the current shape to closer to the previously decided saved area. How close the area value depends on the number of iterations and the difference between the previous area and the new area.

Furthermore, users may draw XBSC markings in the inside of an XBSC object or shape. When the object is deforming, the deformation will also affect the markings accordingly. The markings'

control points are tied to the closest point on the deforming XBSC (i.e., either side of its envelop or its skeleton). Hence, when the main XBSC object deforms, their inner marking control points change along with it. Fig. 13 shows an example of such a deformation. Having reduced its height, the vase deforms while maintaining the same area and its markings change accordingly.

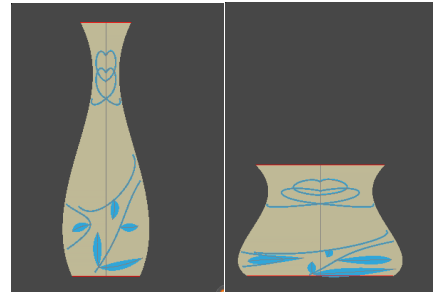


Figure 13. A vase object being deformed (left to right). The blue markings on the vase are constrained with the same deformation.

Fig. 14 shows another example of the vase being stretched and squashed to illustrate the ease with which constant area (but not the same shape) can be maintained in an animation.

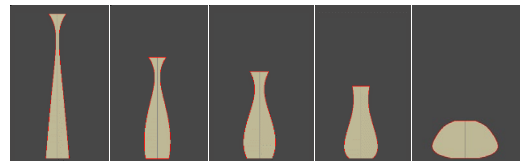


Figure 14. Vase being stretched and squashed while maintaining the same area. The original vase (middle), the stretched vases (left), the squashed vases (right).

Computational Cost

The computational cost of drawing an XBSC without shape deformation and color diffusion is only marginally more than drawing a B-spline curve with similar quality. The shape deformation computes the change of the radii of the XBSC. It takes around 1 second for 100 iterations, which is not too high. Using the biconjugate stabilization method to solve Eqn. 1, the diffusion computation takes around 2 seconds to complete a 700x700 image. This is the most compute intensive task.

5. CONCLUSION

We have discussed the properties of XBSC and its use in modeling and rendering shape and deformation. From our investigation we show that by using XBSC artists have more freedom of drawing expression not only in terms of shape and color, but also in terms of shape deformation. As

mentioned before, deformation is currently limited to a single stroke, which can be improved to enable multiple stroke deformation. XBSC has potentials in the future and there are rooms for improvement. For instance, applying GPU-based Poisson equation solver to accelerate the diffusion curve computation, keyframing the deformation to produce coherent and smooth deformation animation, and applying XBSC in 3D space.

6. ACKNOWLEDGMENTS

This research is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RG 22/20). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of the Ministry of Education, Singapore.

7. REFERENCES

- [Ao18a] Ao, X., Fu, Q., Wu, Z., Wang, X., Zhou, M., Chen, Q., Seah, H.S. An intersection algorithm for disk B-spline curves. *Computers & Graphics*. 70, 99–107, 2018.
<https://doi.org/10.1016/j.cag.2017.07.021>.
- [Che98a] Cheng, S. W., Edelsbrunner, H., Fu, P., & Lam, K. P. Design and analysis of planar shape deformation. In *Proceedings of the Fourteenth Annual Symposium on Computational geometry*, pp. 29–38, 1998.
- [Com15a] Company, P., Plumed, R., Varley, P.A.C. A fast approach for perceptually-based fitting strokes into elliptical arcs. *Vis Comput*. 31, 775–785, 2015.
<https://doi.org/10.1007/s00371-015-1099-6>.
- [Diz11a] Dziol, R., Bender, J., Bayer, D. Robust real-time deformation of incompressible surface meshes. In: *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. pp. 237–246. Association for Computing Machinery, New York, NY, USA, 2011. <https://doi.org/10.1145/2019406.2019438>.
- [Fu16a] Fu, Q., Wu, Z., Ying, X., Wang, M., Zheng, X., Zhou, M. Generating chinese calligraphy on freeform shapes. In: Gavrilova, M.L., Tan, C.J.K., and Sourin, A. (eds.) *Transactions on Computational Science XXVIII: Special Issue on Cyberworlds and Cybersecurity*. pp. 69–87. Springer, Berlin, Heidelberg, 2016.
https://doi.org/10.1007/978-3-662-53090-0_4.
- [Gon13a] González Hidalgo, M., Torres, A.M., Gómez, J.V. (eds.) *Deformation models: tracking, animation and applications*. Lecture Notes in Computer Vision and Biomechanics, vol. 7. Springer, New York, 2013.
- [Hou18a] Hou, Fei, Qian Sun, Zheng Fang, Yong-Jin Liu, Shi-Min Hu, Hong Qin, Aimin Hao, and Ying He. Poisson vector graphics (pvg). *IEEE Transactions on Visualization and Computer Graphics* 26, no. 2, 1361–1371, 2018.
- [Hsu94a] Hsu, S.C., Lee, I.H.H. Drawing and animation using skeletal strokes. In: *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*. pp. 109–118. ACM, New York, NY, USA, 1994.
<https://doi.org/10.1145/192161.192186>.
- [Jes16a] Jeschke S. Generalized diffusion curves: An improved vector representation for smooth-shaded images *Comput. Graph. Forum*, vol. 35, no. 2, pp. 71–79, 2016.
- [Kir83a] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. Optimization by simulated annealing. *Science*. 220, 671–680, 1983.
<https://doi.org/10.1126/science.220.4598.671>.
- [Orz13a] Orzan, A., Bousseau, A., Barla, P., Winnemöller, H., Thollot, J., Salesin, D. Diffusion curves: a vector representation for smooth-shaded images. *Commun. ACM*. 56, 101–108, 2013.
<https://doi.org/10.1145/2483852.2483873>.
- [Pha91a] Pham, B. Expressive brush strokes. *CVGIP: Graphical Models and Image Processing*. 53, 1–6, 1991. [https://doi.org/10.1016/1049-9652\(91\)90013-A](https://doi.org/10.1016/1049-9652(91)90013-A).
- [Por03a] Portells, M.M., Mir, A., Perales, F. Shape deformation models using non-uniform objects in multimedia applications. In: Perales, F.J., Campilho, A.J.C., de la Blanca, N.P., Sanfeliu, A. (eds) *Pattern Recognition and Image Analysis*. IbPRIA 2003. *Lecture Notes in Computer Science*, vol 2652. Springer, Berlin, Heidelberg, 2003.
https://doi.org/10.1007/978-3-540-44871-6_61
- [Pud94a] Pudet, T. Real time fitting of hand-sketched pressure brushstrokes. *Computer Graphics Forum*. 13, 205–220, 1994. <https://doi.org/10.1111/1467-8659.1330205>.
- [Sea05a] Seah, H.S., Wu, Z., Tian, F., Xiao, X., Xie, B. Artistic brushstroke representation and animation with disk b-spline curve. In: *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. pp. 88–93. ACM, New York, NY, USA, 2005.
<https://doi.org/10.1145/1178477.1178489>.
- [Sea05b] Seah, H.S., Wu, Z., Tian, F., Xiao, X., Xie, B. Interactive free-hand drawing and In-between generation with DBSC. In: *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. pp. 385–386. ACM, New York, NY, USA, 2005.
<https://doi.org/10.1145/1178477.1178561>.
- [Sea22a] Seah, H.S., Tandianus, B., Sui, Y., Wu, Z. Expressive B-spline curves: A pilot study on a flexible shape representation. In: Muramatsu, S., Nakajima, M., Kim, J.-G., Guo, J.-M., and Kemao, Q. (eds.) *International Workshop on Advanced Imaging Technology (IWAIT) 2022*. p. 87. SPIE, Hong Kong, China, 2022.
<https://doi.org/10.1117/12.2626063>.

- [Sio90a] Siong Chua, Y. Bézier brushstrokes. *Computer-Aided Design*. 22, 550–555, 1990. [https://doi.org/10.1016/0010-4485\(90\)90040-J](https://doi.org/10.1016/0010-4485(90)90040-J).
- [Str86a] Strassmann, S. Hairy brushes. In: *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*. pp. 225–232. ACM, New York, NY, USA, 1986. <https://doi.org/10.1145/15922.15911>.
- [Su02a] Su, S.L., Xu, Y.-Q., Shum, H.-Y., Chen, F. Simulating artistic brushstrokes using interval splines. In: *Proceedings of the 5th IASTED International Conference on Computer Graphics and Imaging*. pp. 85–90. Citeseer, 2002.
- [Wan16a] Wang, M., Fu, Q., Wang, X., Wu, Z., Zhou, M. Evaluation of Chinese calligraphy by using DBSC vectorization and ICP algorithm. *Mathematical Problems in Engineering*. 2016, 1–11, 2016. <https://doi.org/10.1155/2016/4845092>.
- [Whi83a] Whitted, T. Anti-aliased line drawing using brush extrusion. In: *Proceedings of the 10th Annual Conference on Computer Graphics and Interactive Techniques*. pp. 151–156. ACM, New York, NY, USA, 1983. <https://doi.org/10.1145/800059.801144>.
- [Wu21a] Wu, Z., Wang, X., Liu, S., Chen, Q., Seah, H.S., Tian, F. Skeleton-based parametric 2-D region representation: Disk B-spline curves. *IEEE Computer Graphics and Applications*. 41, 59–70, 2021. <https://doi.org/10.1109/MCG.2021.3069847>

Training Image Synthesis for Shelf Item Detection reflecting Alignments of Items in Real Image Dataset

Tomokazu Kaneko, Ryosuke Sakai, Soma Shiraishi

NEC Visual Intelligence Research Laboratories

211-8666, Kawasaki, Kanagawa, Japan

{tomokazu-kaneko, rsakai_zzkot, s-shiraishi}@nec.com

ABSTRACT

We propose a novel cut-and-paste approach to synthesize a training dataset for shelf item detection, reflecting the alignments of items in the real image dataset. The conventional cut-and-paste approach synthesizes large numbers of training images by pasting foregrounds on background images and is effective for training object detection. However, the previous method pastes foregrounds on random positions of the background, so the alignment of items on shelves is not reflected, and unrealistic images are generated. Generating realistic images that reflect actual positional relationships between items is necessary for efficient learning of item detection. The proposed method determines the pasting positions for the foreground images by referring to the alignment of the items in the real image dataset, so it can generate more realistic images that reflect the alignment of the real-world items. Since our method can synthesize more realistic images, the trained models can perform better.

Keywords

Object detection, Training data synthesis, Retail item recognition, Automatic annotation

1 INTRODUCTION

Image-based retail item recognition contributes to the efficient operation of stores. For example, monitoring item shelves with surveillance cameras can provide out-of-stock detection or planogram analysis services. The automatic method to create item image databases from shelf images has also been proposed in [6]. For these applications, item detection models are required to localize the position of items in the captured images.

Training data annotated with the bounding box of the item position is required to train item detection models. However, the annotation cost is high due to many items being densely aligned on the shelves. The SKU-110K [9] is a public dataset for item detection, but it only contains images taken in a specific country or region, which means it cannot support items sold locally.

The cut-and-paste method [4] is a method for synthesizing large amounts of training data for object detection. The cut-and-paste method can generate various patterns of images at a low cost by pasting foreground images onto background images. Therefore, by pasting images of local items onto the background shelf im-

age, the training dataset for local item detection can be generated without shooting the items on shelves in real-world stores.

Conventional cut-and-paste methods paste the foreground image at a random position in the background image. Such random pasting methods are effective when objects appear in random positions in the image. However, in the case of shelf item detection, the items are regularly aligned, and there is less occlusion between items. As items can have complex textures, irregular occlusion between items due to pasting in random positions makes the boundaries and textures of the items too complex and difficult for training.

This paper proposes a new cut-and-paste method that reflects the alignments of the item positions. The proposed method realistically arranges shelf images by referring to the positional information of items from a real image dataset to determine the position to paste them (Figure 1). Using public datasets as reference datasets, no additional annotation costs are required, and realistic training data can be generated at a low cost.

Realistic images contribute to the training of high-performance detection models. In particular, the proposed method reproduces the regular alignment of items on real-world shelves, which allows the correct boundaries of items to be learned without generating too complex occlusions. We show that the proposed method can generate more realistic images, and the model trained on these images performs better in evaluation experiments on real store images.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

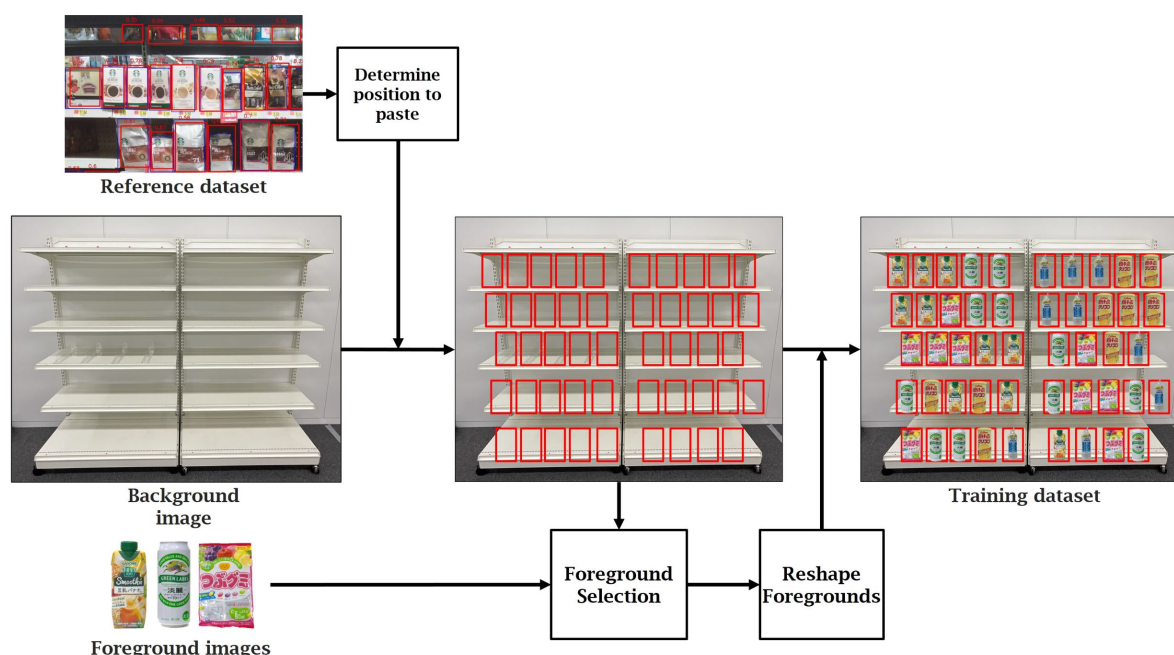


Figure 1: Outline of our approach. The details are explained in Section 3.

2 RELATED WORK

Cut-and-paste methods are proposed for many applications to synthesize a training dataset since it is low cost but effective [4, 5, 8, 10, 14, 19, 20, 21]. In these approaches, foreground images are pasted on a background image and positions of the pasted images are annotated automatically. Several papers reported the effect of the above approaches in industrial applications [14, 19]. It was shown that the cut-and-paste approach was effectively applied to an item recognition task for a self-checkout system. Since these methods determine positions on which foregrounds are pasted randomly, they are effective in a situation when target objects are placed on random positions such as a self-checkout system. However, the methods fail to synthesize realistic images in cases where the target objects are arranged following a pattern, as in the items on a shelf. The model, then, fails to learn the relationships between objects using the data.

There are advanced approaches based on a cut-and-paste method, which consider the positions on which the foreground images are pasted [1, 3, 7]. In [7], they use a depth sensor to estimate the support surface on which a real object is likely placed, floor and desk in background images. Foreground images are pasted on these surfaces, and realistic images are synthesized consequently. However, we need to prepare a depth sensor to use this approach, and moreover, generated images do not reflect the positional relationship between objects. The method in [1] also estimates realistic positions in a background image on which foregrounds

are pasted. Its target is driving scenes, so the suitable positions to place car images are on the road. Since there are many driving scene datasets in public and the road is distinctive, the road estimator can be made robust through training on RGB images. It becomes strict when there are many variations of backgrounds and enough background images cannot be collected. For generic tasks, the approach in [3] is effective. In their approach, the context convolutional neural network (CNN) that estimates the context of backgrounds and foreground images is trained and selects the foreground image that is suitable to paste on each position of a background. In this way, extra sensors are not necessary, and the estimator can learn the context of the image from a generic image set. However, since this approach learns the relationship between foregrounds and backgrounds, it does not work on the scene like objects placed densely and the background is covered such as planogram analysis, and this method also does not reflect the positional relationship between foregrounds.

A 3D simulator is another way to synthesize realistic images. On a 3D simulator, we can place objects anywhere, and using a physics engine, we can simulate stability or interactions between objects. In fact, 3D simulators are used to synthesize training images for object detection [2, 11, 12, 13, 15, 18]. In these methods, the positions of objects in rendered images are annotated automatically, and therefore, a large amount of training data is generated. However, to reproduce target scenes on a 3D simulator, we need 3D models of target objects and backgrounds. Since preparing 3D models is

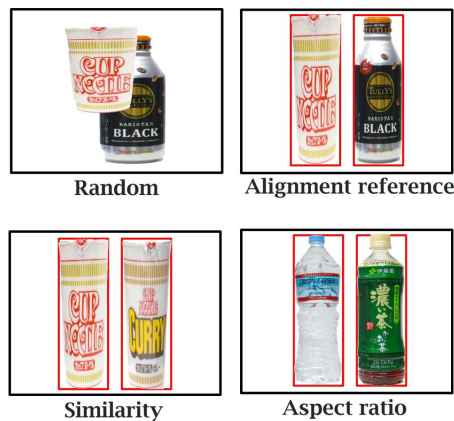


Figure 2: Differences between approaches. The random approach pastes foregrounds on random positions, thus, there is no alignment and foregrounds may occlude each other. The alignment reference approach reproduces a realistic alignment of objects; however, foregrounds are reshaped unrealistically. In the similarity approach, objects in the same category are placed in the neighborhood. The aspect ratio approach selects foregrounds fit to bounding boxes, thus, its aspect ratio does not change so much, and therefore, a realistic appearance is achieved

expensive, the cost of covering many objects is higher than that in the cut-and-paste method. This is more critical in the item detection task for retail stores, in which hundreds of new products are introduced every week.

3 PROPOSED APPROACH

We propose a novel approach to synthesize a dataset for shelf item detection based on the cut-and-paste approach reflecting the alignments of real-world items. First, in this section, we explain the algorithm to determine positions on which foregrounds are pasted by referring to the alignment, which is the core idea of our approach. Then, we explain two specific methods for foreground selection to synthesize more realistic images: the first is based on the object similarity, the second is based on the aspect ratio of the bounding box and the foreground image. Finally, we explain how to combine the two methods. The differences between each approach are summarized in Figure 2.

3.1 Cut and paste referring to object alignment

Figure 1 shows the outline of our approach. The proposed method uses three datasets. The reference dataset is an image dataset of item shelves taken in real stores. The images of the reference dataset are taken at a specific location, so the items we want to detect do not appear. The reference dataset is annotated with a bound-

ing boxes representing the item positions. The proposed method uses the annotation data and size information ($W_{\text{ref}}^k, H_{\text{ref}}^k$) of each image. If the size information is provided as metadata, preparing images of the item shelf is unnecessary for generating process. The background images are the image set of empty shelves. The foreground images are the image set of the items to be detected in which the foreground area has been cropped out.

We define the following notations for the background image set \mathcal{B} , foreground item image set \mathcal{A} , and bounding box annotations of the reference dataset \mathcal{T} as,

$$\mathcal{B} = \left\{ b^k \in \mathbb{R}^{W_{\text{bg}}^k \times H_{\text{bg}}^k \times 3} \right\}_{k=1}^{N_{\text{bg}}}, \quad (1)$$

$$\mathcal{A} = \left\{ a^k \in \mathbb{R}^{W_{\text{fg}}^k \times H_{\text{fg}}^k \times 3}, m^k \in [0, 1]^{W_{\text{fg}}^k \times H_{\text{fg}}^k} \right\}_{k=1}^{N_{\text{fg}}}, \quad (2)$$

$$\mathcal{T} = \left\{ T^k \in \mathbb{R}^{4 \times N_k} \right\}_{k=1}^{N_{\text{ref}}}. \quad (3)$$

Where $N_{\text{bg}}, N_{\text{fg}}, N_{\text{ref}}$, and N_k denote the number of background images, foreground images, reference dataset images, and objects in the k -th reference image, respectively. Furthermore, where $W_{\text{bg}}^k, H_{\text{bg}}^k, W_{\text{fg}}^k$, and H_{fg}^k are the width and height of the background image and the foreground image, respectively, for the k -th image, taking into account that they may differ from image to image. The foreground image is a transparent image to be pasted onto the background, where m^k represents the alpha mask of the foreground.

At first, in the pasting process, the proposed method selects one background image b^k and one reference annotation $T^k = \{(x_l^k, y_l^k, w_l^k, h_l^k)\}_{l=1}^{N_k}$ at random by the uniform distribution. Next, the method selects one bounding box $t_l^k = (x_l^k, y_l^k, w_l^k, h_l^k) \in T^k$ and determines the foreground pasting position by resizing the bounding box to fit the background image,

$$(x_l, y_l, w_l, h_l) = \left(\frac{W_{\text{bg}}^k}{W_{\text{ref}}^k} x_l^k, \frac{H_{\text{bg}}^k}{H_{\text{ref}}^k} y_l^k, \frac{W_{\text{bg}}^k}{W_{\text{ref}}^k} w_l^k, \frac{H_{\text{bg}}^k}{H_{\text{ref}}^k} h_l^k \right), \quad (4)$$

where W_{ref}^k and H_{ref}^k denote width and height of selected reference image, respectively. After that, one foreground image $(a^l, m^l) \in \mathcal{A}$ is selected and resized to fit into the bounding box,

$$(\tilde{a}^l, \tilde{m}^l) = (R_{w_l, h_l}(a^l), R_{w_l, h_l}(m^l)), \quad (5)$$

where $R_{w, h}(\cdot, \cdot)$ denotes the function that resizes an image to $w \times h$ size. Finally, the method pastes the resized image at the bounding box position with alpha blending. This is repeated until there are no more empty bounding boxes.

$$I_{ij} = \left(1 - \sum_{l=1}^{N_k} P_{x_l, y_l}(\tilde{m}_{ij}^l) \right) \cdot b_{ij}^k + \sum_{l=1}^{N_k} P_{x_l, y_l}(\tilde{m}_{ij}^l) \cdot \tilde{a}_{ij}^l, \quad (6)$$

where i and j denote pixel coordinates of images and $P_{x,y}(\cdot)$ denotes the offset function, which shifts the image coordinates i and j to the pasting coordinates x and y .

Images synthesized in this way reflect the alignment in the real scene, and therefore, they achieve more realistic appearances than randomly synthesized images.

3.2 Foreground selection based on object similarity

Real-world shelves have a feature that similar items are placed in the neighborhood of each other. For example, items on a beverage shelf may be collected from the same category, such as coffee, tea, or milk, in which similarly shaped bottles. To reproduce this appearance, we propose a method to select similar images when selecting foreground images.

To paste similar images close to each other, the proposed method first selects a bounding box in the neighborhood of the already pasted bounding box t_l^k ,

$$(x_{l+1}, y_{l+1}, w_{l+1}, h_{l+1}) = \arg \min_{t^k \in \bar{T}^k} d(t_l^k, t^k), \quad (7)$$

where $\bar{T}^k \subset T^k$ represents the set of bounding boxes to which the foreground has not yet been pasted, and $d(\cdot, \cdot)$ is a function that calculates the distance between two bounding boxes. We use Euclidean distance between centers of bounding boxes. The foreground image is then selected from similar images to image (a^l, m^l) which was pasted to the neighboring bounding box,

$$(a^{l+1}, m^{l+1}) = \arg \max_{(a,m) \in \bar{\mathcal{A}}} S(a, a^l), \quad (8)$$

where $\bar{\mathcal{A}} \subset \mathcal{A}$ represents the set of foreground images that have not been pasted, and $S(\cdot, \cdot)$ is a function that outputs the similarity between the two images. The proposed method pastes the selected foreground image onto the selected bounding box position, as described in Section 3.1.

There are several ways to select a similar foreground. One way is to select from the same category or product code. Another way is to use a feature extractor and measure the feature similarity of extracted feature vectors of foreground images. In the following experiments in section 4, we select similar foregrounds by selecting the images of the same product code but viewed from different angles. This way, we can synthesize the appearance of the shelf on which the same objects are placed next to each other facing differently bit by bit.

3.3 Foreground selection based on aspect ratio

The bounding boxes in the reference dataset have various aspect ratios, and the aspect ratios change due to

Algorithm 1 Cut-and-paste procedure referring object alignment, similarity and aspect ratio.

Input: $\mathcal{A}, b^k, \bar{T}^k, p \in [0, 1]$

```

1: SimilarityFlag  $\leftarrow$  False
2: Last_bbox  $\leftarrow$  []
3: annotation  $\leftarrow$  []
4: for  $l \leftarrow 1 \dots \text{len}(\bar{T}^k)$ 
5:   if SimilarityFlag then
6:     Select  $t_l^k$  from  $\bar{T}^k$  by Eq. (7)
7:     Select  $(a^l, m^l)$  from  $\mathcal{A}$  by Eq. (8)
8:   else
9:     Randomly select  $t_l^k$  from  $\bar{T}^k$ 
10:    Select  $(a^l, m^l)$  from  $\mathcal{A}$  by Eq. (9)
11:   end if
12:    $t_l \leftarrow$  Reshape  $t_l^k$  by Eq. (4)
13:    $(\tilde{a}^l, \tilde{m}^l) \leftarrow$  Reshape  $(a^l, m^l)$  to fit  $t_l$  by Eq. (5)
14:   Paste  $\tilde{a}^l$  at position  $t_l$  in  $b^k$ 
15:   Append  $t_l$  to annotation
16:   rand  $\leftarrow$  a random number between 0 and 1
17:   if rand  $< p$  then
18:     SimilarityFlag  $\leftarrow$  True
19:   else
20:     SimilarityFlag  $\leftarrow$  False
21:   end if
22: end for
23: return  $b^k$ , annotation

```

the transformation of Equation (4). The foreground image dataset may also contain images with varying aspect ratios. Due to these factors, the aspect ratio of the foreground image changes during the transformation in Equation (5).

To synthesize a realistic image, the aspect ratio of the foreground image must not change too much from the original. The following algorithm selects a foreground image whose aspect ratio is close to the aspect ratio of the bounding box. The algorithm first calculates the aspect ratio of the bounding box and selects a foreground image with a similar aspect ratio.

$$(a^l, m^l) = \arg \min_{(a,m) \in \bar{\mathcal{A}}} |r(a) - w_l/h_l|, \quad (9)$$

where $r(\cdot)$ is a function to calculate the aspect ratio of the image. After that, we reshape the foreground image to fit the bounding box and paste the foreground. This approach allows the foreground image to be pasted to fit into a bounding box while preserving its aspect ratio. Thus, the synthesized image becomes more realistic with no extremely reshaped items.

3.4 Inclusion of all approaches

The method containing all of the above approaches is shown in Algorithm 1. Our method basically selects a foreground image in accordance with its aspect ratio.



Random [4] Ours
Figure 3: Examples of synthesized images

After pasting one foreground, the algorithm determines whether to select a foreground in accordance with similarity probabilistically. In this way, two foreground selection processes can be included in one algorithm.

4 EXPERIMENTS

We evaluate the proposed approach on the task of shelf item detection. The purpose of the proposed approach is to train a better object detector on synthesized images. To evaluate from this perspective, we compare detection scores of object detectors trained on the synthesized images by the baselines and the proposed approach. To evaluate in a realistic situation, we use shelf images shot in real stores as the evaluation dataset.

4.1 Training dataset

We prepare a training dataset in addition to the public dataset. We add the synthesized dataset to the public dataset to train from both real and synthesized images. This is because there is a domain gap between real and synthesized images and training only on synthesized images suffers from this gap. Training on both domains mitigates this adverse effect.

We adopt SKU-110K as the base dataset. Since SKU-110K does not contain images shot in locale-specific stores or shot from angles of surveillance cameras, the model trained only on SKU-110K does not work well enough in these situations. By adding synthesized data from foreground images of locale-specific items or background images of surveillance angles, the trained models become robust to the uncovered situation.

We use item images shot on a turntable as foregrounds. This image set contains 39,559 images of 1,000 items. Each item is captured from multiple orientations by rotating the turntable. We cut out foregrounds by GrabCut [16] from the captured images. These cut out images of

	Front	Upper	Upper-left	average
SKU-only [9]	0.946	0.893	0.712	0.850
Random [4]	0.945	0.880	0.735	0.853
Ours	0.951	0.894	0.755	0.866

Table 1: Detection scores (AP_{50}) of trained models. SKU-only, Random, and Ours indicate the methods to synthesize training images. Front, Upper, and Upper-left indicate camera angles of the evaluation dataset. All of the scores are the means of three trials of training on different random seeds.

items are used as the foreground images. We use images of empty shelves as backgrounds. This set contains 989 images of five types of shelves. These images are shot from various angles and under various lighting conditions. Using the above foreground and background images, we synthesize a training dataset by each approach. We synthesize 1,000 images for training by each approach and add to SKU-110K training dataset that contains 8,185 real images.

We compare three methods: SKU-only [9], random [4], and our approach. SKU-only means training on SKU-110K dataset only. The random approach is the cut-and-paste method whose pasting position is determined randomly. In the random approach, we paste 147 foregrounds on average on one background image. This number is the same as the average number of objects in one image of SKU-110K. With this condition, the number of foreground objects contained in one training image is the same among comparison methods. In the proposed method, we also use SKU-110K as a reference dataset. We set the parameter p to 0.5, which is the probability of selecting a foreground by similarity and pasting it on the neighborhood. To increase the variation of the appearance of the foreground, for all comparison methods, we randomly rotate the foreground image with a probability of 0.1 when pasting it.

Figure 3 shows examples of synthesized images. Figure 3-(a) is synthesized by the random approach, whose foregrounds are pasted on random positions and frequently occlude each other. On the other hand, in the proposed approach shown in Figure 3-(b), items are lined up in accordance with the object alignment in SKU-110K. The reshaping of foregrounds is realistic, and the same items shot from various angles are pasted close together, this reproduces a more realistic appearance of shelves.

We adopt EfficientDet-d0 [17] as a detector model. Hyper-parameters, training epochs, and the learning rate, are tuned by the validation dataset that consists of the SKU-110K test-set and 100 synthesized images by each method.

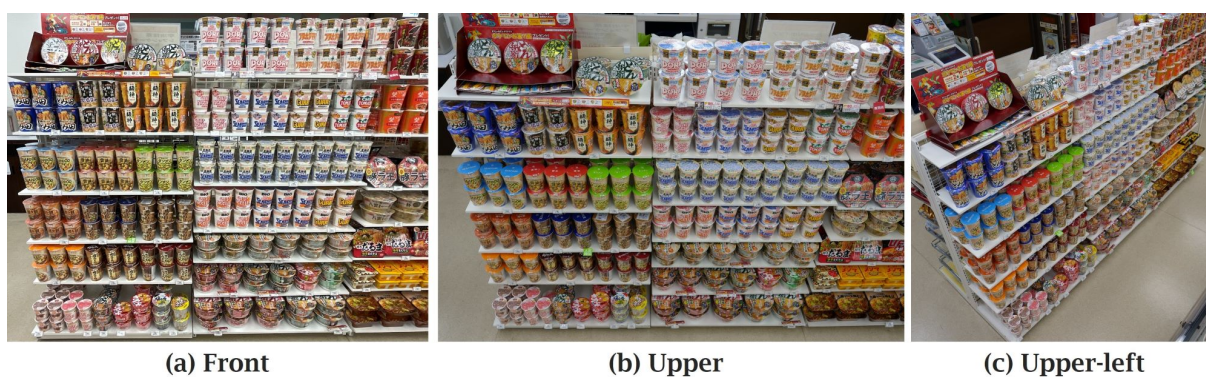


Figure 4: Examples of evaluation images. (a), (b), and (c) show the images taken from the Front, Upper, and Upper-left angles, respectively. The items in the Upper and Upper-left are deformed compared to the Front image due to parallax.



Figure 5: Detection results on the front angle data. Green and red bounding boxes represent outputs of the model with confidence scores more than 0.4. Green bounding boxes have IoU with ground truth more than 0.5, and red bounding boxes are less than 0.5.

4.2 Evaluation dataset

We use images shot in real stores as the evaluation set (Figure 4). This image set was shot in two convenience stores for four days. The target shelves are drinks, snacks, and instant noodles. There are three variations of camera angles: front, upper and upper-left, where each set consists of 32 images.

The detection targets are items on the target shelves, whose whole body is within the image, and therefore, items on non-target shelves are not subject to aggregation. The metric of evaluation is average precision (AP) of all items in one class.

4.3 Results

The experimental results are shown in Table 1. For all evaluation sets, the proposed method performs the best. For the front and the upper angle data, the scores of random approach decrease relative to SKU-only. This shows that unrealistic images synthesized by the random approach adversely affect the training. On the

other hand, the proposed approach positively affects all of the targets.

Detection results are shown in Figure 5. One notable example is the chocolate box on the upper left corner of Figure 5-(a) and (b). In the random approach, the chocolate box is recognized as two objects. The random approach synthesizes crowded and complex images as shown in Figure 3. Due to this, the detection model trained on such images tends to split objects of complex texture into two different objects. On the other hand, in the proposed approach, the model recognizes the chocolate box correctly.

In Figure 5-(c), some noodles stacked in two layers on the bottom row are detected as one object in the random method. On the other hand, in our approach they are detected correctly as shown in Figure 5-(d). This is the effect of the alignment approach with object similarity, that is, our approach can detect objects in the scene with similar objects that are stacked and aligned densely.

	Alignment reference	Object similarity	Aspect ratio	Front	Upper	Upper-left	average
Random [4]				0.945	0.880	0.735	0.853
Align only	✓			0.943	0.868	0.718	0.843
Align + Sim	✓	✓		0.946	0.872	0.732	0.850
Align + Aspect	✓		✓	0.949	0.843	0.701	0.831
Align + Sim + Aspect	✓	✓	✓	0.951	0.894	0.755	0.866

Table 2: Ablation study of our method. Align, Sim, and Aspect denote the approaches described in Section 3.1, 3.2, and 3.2, respectively.



Alignment only

Alignment + Similarity

Alignment + Aspect

Figure 6: Examples of synthesized images in ablation study.

4.4 Ablation study

We conduct an ablation study of our approach in Table 2. The alignment reference approach without considering the object similarity and aspect ratio (Align only) is worse than the random approach. One reason is unrealistic reshaping of foreground images. This can be confirmed in results on the front data, that is, the score of the proposed approach while considering the aspect ratio (Align + Aspect) is higher than that of the random approach. However, on the other data, this tendency is not clear on the upper and upper-left images. This is because the objects shot from the upper or upper-left angles have reshaped appearance, so there are cases where it is better not to select foregrounds on the basis of the aspect ratio. The approach considering object similarity (Align + Sim + Aspect) is better on all of the evaluation data.

Figure 6 shows synthesized images by the approaches in the ablation study. In the alignment only approach, some foreground images are reshaped extremely and their appearance is unrealistic. On the other hand, in the alignment + aspect approach that takes into account the

aspect ratio of the box, the appearance of foregrounds is not so different from reality and objects are aligned. In the alignment + similarity approach that takes into account the object similarity, similar objects are placed in the neighborhood, which achieves a similar appearance to shelves in stores. As above, each approach has advantages for realistic synthetization, and combining all of the approaches, the most realistic image in Figure 3-(b) is synthesized.

5 CONCLUSION

We proposed a novel cut-and-paste method for training shelf item detection models. The proposed method determines the pasting positions for the foreground images, referring to the annotations of a dataset of real-world shelves images. Furthermore, to generate more realistic images, the proposed method selects the foreground image with reference to the similarity of the items and the aspect ratio of the bounding box of the pasting position. Experiments show that the proposed method can generate more realistic images of the item shelves than the conventional random pasting method

and that the dataset of the images can be used to train more accurate item detection models.

As the proposed method determines the pasting positions without specifying the positions of the shelves in the image, if the positions of the shelves in the reference dataset image and the background image change drastically, the items will be placed at locations other than the shelves. In order to generate a more realistic image where the items are accurately placed on the shelves, annotations of the shelf positions in the background image should be prepared, and the pasting positions should be adjusted based on the annotations. Verification of such a generation method is future work.

6 REFERENCES

- [1] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother. Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes. *International Journal of Computer Vision*, 126(9):961–972, Sept. 2018.
- [2] E. Bochinski, V. Eiselein, and T. Sikora. Training a convolutional neural network for multi-class object detection using solely virtual world data. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 278–285, 2016.
- [3] N. Dvornik, J. Mairal, and C. Schmid. Modeling visual context is key to augmenting object detection datasets. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 375–391, Cham, 2018. Springer International Publishing.
- [4] D. Dwibedi, I. Misra, and M. Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017.
- [5] H.-S. Fang, J. Sun, R. Wang, M. Gou, Y.-L. Li, and C. Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 682–691, 2019.
- [6] M. Filax, T. Gonschorek, and F. Ortmeier. Semi-automatic Acquisition of Datasets for Retail Recognition. *Computer Science Research Notes*, 3201:86–94, 2022.
- [7] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka. Synthesizing training data for object detection in indoor scenes. 2017.
- [8] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2917–2927, 2021.
- [9] E. Goldman, R. Herzig, A. Eisenschlat, J. Goldberger, and T. Hassner. Precise detection in densely packed scenes. In *Proc. Conf. Comput. Vision Pattern Recognition (CVPR)*, 2019.
- [10] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [11] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige. On pre-trained image features and synthetic images for deep learning. In L. Leal-Taixé and S. Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 682–697, Cham, 2019. Springer International Publishing.
- [12] S. Hinterstoisser, O. Pauly, H. Heibel, M. Marek, and M. Bokeloh. An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Instance Detection. Feb. 2019.
- [13] T. Hodaň, V. Vineet, R. Gal, E. Shalev, J. Hanzelka, T. Connell, P. Urbina, S. N. Sinha, and B. Guenter. Photorealistic image synthesis for object instance detection. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 66–70, 2019.
- [14] S. Koturwar, S. Shiraishi, and K. Iwamoto. Robust multi-object detection based on data augmentation with realistic image synthesis for point-of-sale automation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9492–9497, July 2019.
- [15] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.
- [16] C. Rother, V. Kolmogorov, and A. Blake. "Grab-Cut": Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers, SIGGRAPH '04*, pages 309–314, New York, NY, USA, 2004. Association for Computing Machinery.
- [17] M. Tan, R. Pang, and Q. V. Le. EfficientDet: Scalable and efficient object detection. *CoRR*, abs/1911.09070, 2019.
- [18] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Bochoon, and S. Birchfield. Training deep networks

- with synthetic data: Bridging the reality gap by domain randomization. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1082–10828, 2018.
- [19] X.-S. Wei, Q. Cui, L. Yang, P. Wang, and L. Liu. RPC: A large-scale retail product checkout dataset. *CoRR*, abs/1901.07249, 2019.
- [20] S.-F. Wu, M.-C. Chang, S. Lyu, C.-S. Wong, A. K. Pandey, and P.-C. Su. FlagDetSeg: Multi-nation flag detection and segmentation in the wild. In *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8, 2021.
- [21] W.-H. Yun, T. Kim, J. Lee, J. Kim, and J. Kim. Cut-and-Paste Dataset Generation for Balancing Domain Gaps in Object Instance Detection. *IEEE Access*, 9:14319–14329, 2021.

SAIL: Semantic Analysis of Information in Light Fields: Results from Synthetic and Real-World Data

Robin Kremer
Saarland University
Saarland Informatics Campus
Campus Building C6 3
Germany 66123, Saarbrücken, Saarland
kremer@cs.uni-saarland.de

Thorsten Herfet
Saarland University
Saarland Informatics Campus
Campus Building C6 3
Germany 66123, Saarbrücken, Saarland
herfet@cs.uni-saarland.de

ABSTRACT

Computational photography has revolutionized the way we capture and interpret images. Light fields, in particular, offer a rich representation of a scene's geometry and appearance by encoding both spatial and angular information. In this paper, we present a novel approach to light field analysis that focuses on semantics. In contrast to the uniform distribution of samples in two-dimensional images, the distribution of samples in light fields varies for different scene regions. Some points are sampled from multiple directions, while others may only be captured by a small portion of the light field array. Our approach provides insights into this non-uniform distribution and helps guide further processing steps to fully leverage the available information content.

Keywords

Light fields, Computational photography, semantic analysis, information content, MPEG-I

1 INTRODUCTION

The objective of enhancing the immersive experience for visual content has been a long-standing pursuit, dating back several decades, starting at analog photography and progressing over digital cameras to 3D video [Sch09]. In recent years computational photography has become one of the most important parts of the complete visual pipeline and is used to optimally use all available data [Lam03; Lib19; Sam21]. The latest frontier in immersive content is the creation of interactive experiences that enable users to freely adjust their viewpoint in real-time. This type of content can encompass a range of immersive experiences, from 3 Degrees of Freedom (DoF), where users can change the direction of their viewpoint, to 6 DoF content, which allows users to also move their position in space [MPE18]. To enable such interactive applications, it is essential to capture a scene from multiple viewpoints, which is typically achieved using light field cameras or light field arrays. Although the resulting data is also visual in nature, there are several significant differences between light fields and traditional 2D imaging. One of the

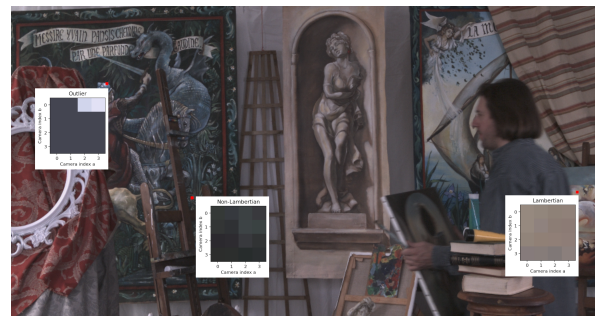


Figure 1: Painter scene from InterDigital [Sab17] captured on a 4 by 4 light field array. Highlighted are three scene regions (froxels). The displayed diagrams show the rays assigned to each froxel. As each ray is captured by a different camera, the color distributions give insights about the view dependency of the regions.

main being that, properties such as view-dependent appearances and occlusions caused by scene geometry are lost in traditional 2D imaging, whereas these effects are captured in light fields. Although the raw data rate of light field capture systems can be upwards of ten gigabits per second [Che20], which presents significant challenges for current processing, network, and storage devices, this additional data provides unique post-processing options. In contrast to recent work that focuses on these applications and makes certain assumptions about the available data (e.g. everything is Lambertian), this work presents a method for analyzing the distribution of information in light fields.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2 RELATED WORK

The underlying theory behind light fields has been established for several decades [Ber91; Lev96], similar to the field of machine learning. However, many applications only recently became computationally feasible. Today, a diverse range of light field filters and processing techniques is available.

With MPEG Immersive video (MIV), which is part of ISO/IEC 23090 MPEG-I, the Moving Picture Experts Group introduced a standard to store and distribute highly immersive 3D content. A content database has been published with scenes captured on a variety of different setups. The general idea is to remove redundancy by merging all views into one and creating a patch atlas for occluded regions. In practice, diverse options to create the atlases are available.

Methods like "Linear Volumetric Focus" [Dan15], "Calibration and Auto-Refinement for Light Field Cameras" [Ani21] and "Fourier Disparity layers" [Le19] are based on traditional algorithms and filters. These techniques enable a range of effects, such as adjusting the focal distance and depth of field or synthesizing new views in real-time. To achieve these results, certain assumptions are often made, such as treating the entire scene as being Lambertian or assuming limited occlusions. If a scene does not adhere to these assumptions, visual artifacts may occur.

Similar to many other fields, such as image segmentation or gigapixel compression, machine learning methods also became a popular choice for light field processing. Techniques like "Deepview" [Fly19] and "Immersive Video" [Bro20] use gradient decent optimisation to represent a scene as a Multi-Plane Images (MPI) or Multi-Sphere Images (MSI) enabling real-time view interpolation even for light field videos. However, it should be noted that the training of these techniques is computationally intensive and can take multiple tens of hours per frame. In the work of "Local Light Field Fusion" (LLFF) [Mil19], MPIs are also utilized, but they employ a single trained network to promote each input view into an MPI (local light field). This significantly reduces the total time from capture to view synthesis (roughly 10 minutes), while still enabling real-time view interpolation.

One of the most disruptive innovations for the field of light field processing in recent years has been the emergence of Neural Radiance Fields (NeRFs) [Mil20]. These encode the information of a light field in multi-layer perceptron (MLP) neural network. Specifically, the MLP takes both the 3D spatial location and viewing direction as input and outputs color and volume density information. In the context of light field theory this amounts to a continuous representation of the underlying plenoptic function and thereby enables synthesis of arbitrary viewpoints. Training NeRFs is

computationally expensive, requiring multiple GPU hours. Additionally, synthesizing novel views from the original NeRF implementation is not feasible in real-time, typically requiring multiple tens of seconds. Although network inference is relatively fast, volumetric rendering necessitates the use of multiple samples per pixel, resulting in millions of inferences required to produce a high-definition view. Despite these limitations, the visual quality of NeRF-generated images is exceptionally high and capable of gracefully handling non-Lambertian and opaque objects. In addition, scenes represented as NeRFs require significantly less memory compared to LLFF, and often are even smaller in size than the original input views.

Numerous techniques have been developed that build upon the fundamental idea of NeRF, enabling the handling of specific types of scenes and addressing limitations of the original implementation. Mip-NeRF [Bar21] increases the visual fidelity by sampling conical frustums instead of rays. This was further developed in Mip-NeRF 360 [Bar22] to better handle unbounded 360 degree scenes. While methods like "NeRF in the dark" [Mil21] and "HDR-NeRF" [Hua22] focus on dynamic range and noise handling. Other methods like "Fastnerf" [Gar21] speed up the inference time significantly rendering 100+ frames per second on modern GPUs. "PixelNeRF" [Yu21], on the other hand, drastically reduces the number of input images compared to traditional NeRF. Works like "Instant Neural Graphics Primitives" [Mül22] can produce high quality results after just 5 minutes of training. Recent techniques such as "Space-time NeRF" [Xia21] and others [Par21; Pum21] have extended the capabilities of NeRFs to be able to handle videos.

In summary, since their introduction, NeRFs have emerged as the most prominent method for processing light fields and have spawned a new research field focused on extending their capabilities. While neural techniques are likely to dominate light field processing, it is crucial for applications such as codecs, post-processing and novel view generation to be capable of real-time operation and to possess an understanding of the underlying scene properties. As a result, this paper does not aim to compete with the processing capabilities of NeRF and its derivatives. Rather, it seeks to address a more fundamental aspect of the complete visual pipeline, namely, what information is captured by a light field array and how it is distributed.

3 THEORY OF LIGHT FIELDS

The theory behind light fields is based around the plenoptic function, which contains all information about the propagation of light in a certain space-time region [Ber91]. A light field is created by sampling this continuous function at certain positions using

cameras. As such, only a small portion of the overall information contained in the plenoptic function is sampled. Nonetheless, the amount of information contained within a light field is substantial and allows for a vast variety of post processing techniques as described earlier.

Although static scenes can be captured by moving a camera on a gantry or handheld, the majority of light fields are captured using multiple cameras that are rigidly fixed together in a camera array (light field array) like the "Stanford Multi-Camera Array" [Wil05] or the light field array from InterDigital [Sab17]. Light fields captured by these rigs are called "forward-facing" since all cameras are oriented into the same direction. The two-plane parameterization (compare Figure 2) is particularly intuitive, as it closely aligns with the physical arrangement of the cameras [Cam98]. The first plane corresponds to the plane on which the cameras are mounted (a,b-plane / camera), while the second plane represents the plane on the camera sensors (u,v-plane / pixel). Due to this close correspondence to the physical arrangement of cameras, the two-plane parameterization is a common starting point for a wide range of light field processing techniques.

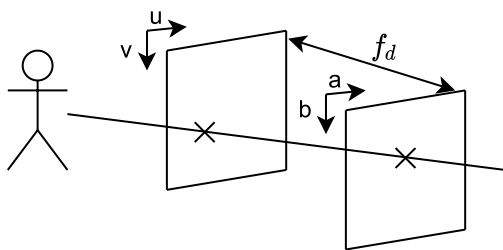


Figure 2: In the two-plane parameterization a light ray is described by its intersection with two parallel planes.

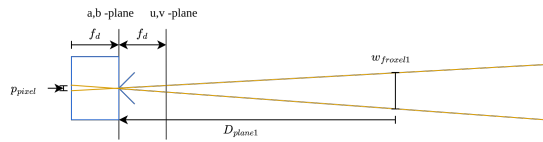
One characteristic of the two-plane parameterization, as well as similar formats like the Direction and Point Parameterization (DPP) and Two-Sphere Parameterization (2SP) [Cam99], is that they present the light field information in a camera-centric format. Although this is intuitive, it hides how the captured information is arranged in a light field. This information distribution is a crucial difference between light fields and traditional 2D imaging. Because unlike single-camera imaging, where pixels evenly sample rays from a scene and capture each visible point exactly once, light fields can have a non-uniform sample distribution. Some scene points are visible in all cameras and are therefore sampled multiple times, while others may be occluded for most cameras and are only sampled by a small subset. The shape of this distribution plays a pivotal role in determining which post-processing techniques can be effectively applied to the captured light field data. For instance, achieving high-quality results with the "Light Field Superresolution" technique [Bis09] requires a sufficient number of samples per scene region, making the

distribution of captured information a critical factor for the effectiveness of this and many other methods. Nevertheless, many light field processing techniques rely on certain assumptions about the distribution of captured information and deviations from these assumptions can lead to artifacts as in [Le 19; Dan15]. In the subsequent chapters, a scene-centric light field parameterization will be explored that enables straightforward analysis of captured information distribution.

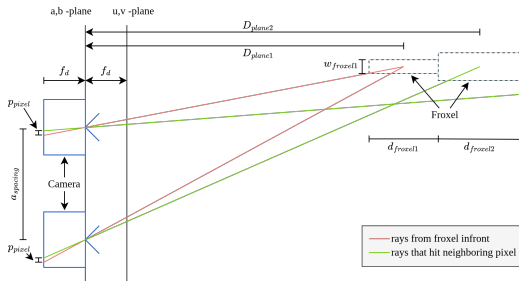
4 THE IDEA OF FROXELS

In order to analyze the distribution of information that is contained in a light field more easily, a scene centric parameterization is needed. In order to achieve this, we make use of the "froxel" concept, which involves discretizing the view frustum of the light field array into frustum-shaped voxels [Eva15]. This is accomplished by populating the view frustum with froxels of specific sizes, which are designed to match the resolution of the light field array. By choosing the size of the froxels appropriately, we can achieve a discretization raster that perfectly matches the array resolution. As a result, if an object in the scene is moved by one froxel, its image will shift exactly one pixel on a camera sensor. Unlike a single camera, which does not capture any information about scene depth and therefore does not require discretization along the depth axis, light field arrays do capture this information [Ber91]. As a result, the view frustum of a light field array has to be discretized in all three dimensions. For the two axes parallel to the camera plane, this discretization scheme is straightforward, since the region covered by a single pixel increases linearly with the distance from the camera (compare figure 3a). However, the resolution along the depth axis of the light field array, and thus the size of a froxel, is dependent on the specific geometry of the array. In light fields, depth information is captured as the disparity experienced by objects within the scene. As a result, the disparity is responsible for the depth resolution and, ultimately, the size of the froxels along the depth axis. Because disparity is inversely proportional to depth, froxels that are closer to the camera plane have smaller depth and become larger as they move further away from the camera. As the largest disparity is experienced between the furthest cameras in an array, the size of the froxels is chosen such that moving an object one froxel closer or further away from the camera plane results in a one-pixel change in its position between these two cameras (compare figure 3b). The exact dimension of the froxels can be calculated with (1) and (2). Where w_{froxel} , h_{froxel} and d_{froxel} are the width, height and depth of a froxel at a certain distance D_{plane} from the camera plane.

$$w_{froxel} = h_{froxel} = \frac{p_{pixel} D_{plane}}{f_d} \quad (1)$$



(a) The froxel width w_{froxel} and height h_{froxel} scale linearly with the distance D_{plane} from the camera plane (a,b-plane)



(b) The froxel depth is based on the maximum disparity that the array can capture

Figure 3: The froxel width, height and depth are chosen to perfectly match the resolution of the light field array

$$d_{froxel} = D_{plane}^2 / \left(\frac{f_d \cdot s_{max}}{p_{pixel}} - D_{plane} \right) \quad (2)$$

It is assumed that all cameras have the same intrinsic parameters such as f_d , which is the focus distance. The maximum distance between two cameras in the light field array is denoted as s_{max} and governs the largest disparity that can be observed at a given depth.

Once the discretization raster is created, each ray captured in the light field is assigned to the froxel that contains the object from which it originated in the scene. This origin is calculated by combining the two-plane parameterization with a depth map. Due to the way the raster is designed, two rays captured by the same camera can not be assigned to the same froxel. However, when two rays are captured by different cameras and originate from the same scene point, they will be assigned to the same froxel. This means one froxel can at most have as many rays assigned to it as there are cameras in the light field array. Froxels that have rays assigned to them will be referred to as non-empty froxels. Once all rays have been assigned to their respective origin froxel, the resulting set of non-empty froxels contains all of the information captured by the light field. The resulting froxel parameterization represents the information in a scene-centric manner, in contrast to the two-plane parameterization, which is camera-centric. The term "scene-centric" refers to the fact that this parameterization allows for easy analysis of the light field captured from a specific scene region, as it facilitates a straightforward examination of how rays and information are distributed throughout the scene.

5 OPTIMIZING THE FROXEL REPRESENTATION

As discussed in the previous section, the transformation from two-plane parameterization to the froxel parameterization relies on depth maps to determine the origin of a captured ray. For the froxel parameterization to be most effective, it is essential that rays originating from the same scene point are assigned to the same froxel. As a result, the accuracy of the depth maps has a significant impact on the achievable quality. This is particularly true, since the froxel sizes are designed to precisely match the resolution of the light field array.

Thus, the most crucial characteristic of the depth maps used in the froxel parameterization is good multi-view consistency, which means that the depth maps of each camera must assign the same depth to a given scene point. This consistency ensures that rays captured by different cameras and originating from the same scene point are assigned to the same froxel, resulting in an accurate representation of the underlying scene.

Upon analyzing multiple datasets that contained depth maps generated using various techniques, it was determined that while the resulting froxel parameterizations were acceptable, there remained potential to improve the meaningfulness. In theory, a wall that is parallel to the camera plane should result in a plane of non-empty froxels located exactly at the depth of the wall. However, in practice, the froxel representations are often narrowly distributed around the true position of the wall. To improve the meaningfulness of the parameterization and reduce the total number of non-empty froxels, a consolidation step is employed. This is based on the idea of reassigning rays from non-empty froxels with few rays to other froxels that already have more rays assigned to them. When reassigning a ray to a different froxel, only the non-empty froxels along the ray's original path are considered to avoid altering the representation too much. By searching for a new froxel within a few neighboring layers, the consolidation step can already reduce the total number of non-empty froxels significantly.

6 SEMANTIC ANALYSIS

Once a light field has been transformed into the froxel representation, it becomes significantly easier to analyze how the captured information is distributed. The number of rays assigned to each froxel following the conversion is a good starting point to analyze the information distribution. Firstly, since the majority of scenes typically contain a significant amount of free space, many froxels will remain unoccupied following the conversion process. Consequently, the froxels that do have rays assigned to them approximate the hull of the scene. However, within these non-empty froxels, there can be substantial differences in the amount of

information present. For instance, a froxel that corresponds to a scene point, which is captured by all of the cameras in a light field array, should have the same number of rays assigned to it as there are cameras in the array. Other froxels that are occluded for part of the array contain fewer rays. Consequently, the number of rays per froxel directly indicate how densely the underlying scene region is sampled. One approach for visualizing this distribution is to use fristograms (froxel + histogram) [Her21]. They are created by grouping froxels according to the number of rays assigned to them and then generating a histogram based on these groupings (compare 4a). They provide an initial indication of the level of uniformity with which a scene is sampled.

Another approach for visualizing the distribution of samples is to count the number of non-empty froxels along a ray. If there is more than one, it indicates that the corresponding scene point is likely occluded in the current viewpoint and was captured from a different perspective by a different camera. This allows to easily locate occluded scene regions that are only visible by a subset of all cameras in the light field array (compare figure 5d). This information may be used in various applications, such as virtual viewpoint rendering or aid in the generation of atlases.

Analyzing the distribution of rays within a froxel reveals additional semantic information. All rays that are assigned to a single froxel originate from the same scene point, but were captured from different directions. Consequently, by analyzing the color distribution of these rays, it is possible to infer the visual properties of the underlying object. If the rays within a froxel exhibit similar colors, this is an indication that the corresponding object behaves as a Lambertian radiator. On the other hand, if there is a significant amount of color variation among the rays, this suggests non-Lambertian behavior [Kop14]. This information is crucial for post-processing, as different techniques may only be effective for certain types of surfaces. The froxel representation also provides a means for quantifying the information captured by a light field, allowing for the comparison of different capture setups. By analyzing the distribution of froxels and their associated rays, it is possible to evaluate the level of scene sampling and coverage achieved by a particular light field capture setup. To quantify the information content I_{total} of a light field, each ray is assigned a specific value that reflects its contribution to the overall scene information. For example, rays originating from a Lambertian surface point may be assigned a lower value compared to those from non-Lambertian or occluded regions, as the former contribute less unique information. In practice, this is often done by grouping rays of a froxel together if their color differs by less than a just-noticeable-difference (JND) [Sha17]. The probability of a ray p_i is then calculated with (4), where n_{rays}

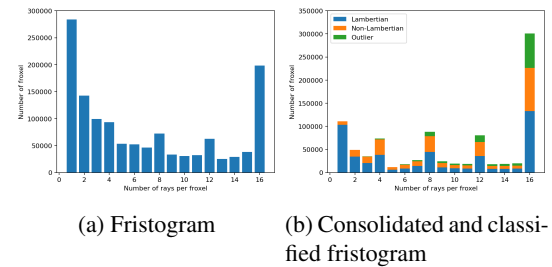


Figure 4: Fristograms of the painter scene from Inter-Digital

is the total number of rays in the light field and $n_{cluster_i}$ is the size of the cluster to which the ray belongs. From this the total information content of the light field can be calculated with (3) [Sha48].

$$I_{total} = \sum_{i=1}^{n_{rays}} -\log(p_i) \quad (3)$$

$$p_i = \frac{n_{cluster_i}}{n_{rays}} \quad (4)$$

This information can be used to optimize the design of future light field acquisition systems for specific applications.

To demonstrate the effectiveness of the froxel representation, the surface properties present in a scene, where analysed by a simple froxel classification. The proposed technique works by analyzing the color distribution of rays assigned to individual froxels. Froxels are classified as non-Lambertian when the standard deviation of their associated rays surpasses a predetermined threshold, while those whose standard deviation is below the threshold are considered Lambertian. Additionally, a third category of "Outliers" is established by identifying non-Lambertian froxels that have at least one ray with a z-score that exceeds a certain value. This indicates that while the majority of the rays associated with a froxel have a uniform distribution of colors, there are a few outliers that do not conform to this pattern. This can be caused by specular highlights or due to wrongly assigned rays (compare figure 1).

The semantics acquired through this method can be utilized to direct post-processing procedures in a manner that minimizes visual artifacts while maximizing the use of all available information.

7 RESULTS

The developed pipeline was tested on synthetic data generated in blender, an open source 3D animation software, and real-world data sourced from the MPEG-I content database. The depth maps utilized during development were either generated in Blender, exported

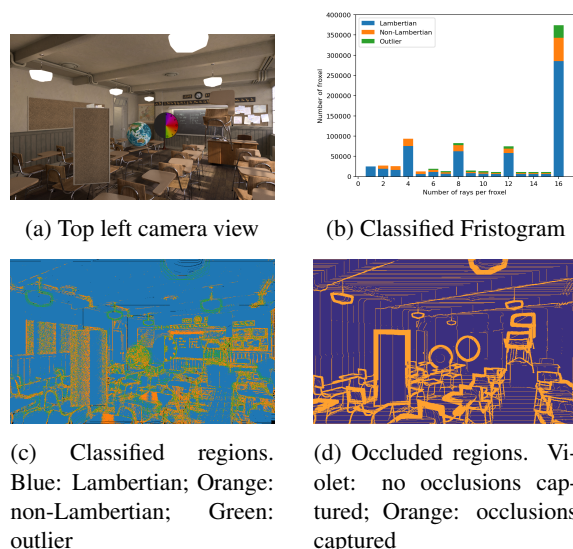


Figure 5: Custom Blender Classroom scene, captured on a 4-by-4 light field array with blender depths

from a NeRF, or provided together with the content. Although our methods are capable of accommodating arbitrary forward-facing light field arrays, the results presented in this paper are all based on light fields captured by uniform 4-by-4 arrays to increase conciseness.

Figure 5 displays results generated with high quality depth maps that were generated in blender. Upon examining the corresponding fristogram, it is clear that there is a prominent peak at 16 rays per froxel. This indicates that the majority of the scene was captured by all the cameras in the 4-by-4 array. Additionally, distinct bumps can be observed at 4, 8, and 12 rays per froxel, which correspond to edges in the scene that align with the arrangement of the cameras in the array, such as the floating cork board in the foreground. These edges create occlusions for multiple cameras simultaneously, resulting in noticeable patterns in the fristogram. Moreover, by including the froxel classes, the fristogram reveals that most of the scene behaves in a Lambertian manner. Examining Figure 5c, it is apparent that the wall and ceiling are classified as Lambertian, whereas the table desks with a glossy finish and the reflective metal chair legs are classified as non-Lambertian. Occlusions that occur in the light field are displayed in 5d. The presence of orange stripes on the edges of objects signifies the existence of samples that lie behind the foreground object within the light field. This information can be leveraged to reveal occluded areas within the scene, or even to identify objects that could potentially be completely eliminated during view reconstruction.

In the shown example of the Classroom scene, the depth maps were generated within Blender, which allowed for access to the scene geometry and, as a result, yielded depth maps of exceptionally high quality. Since such

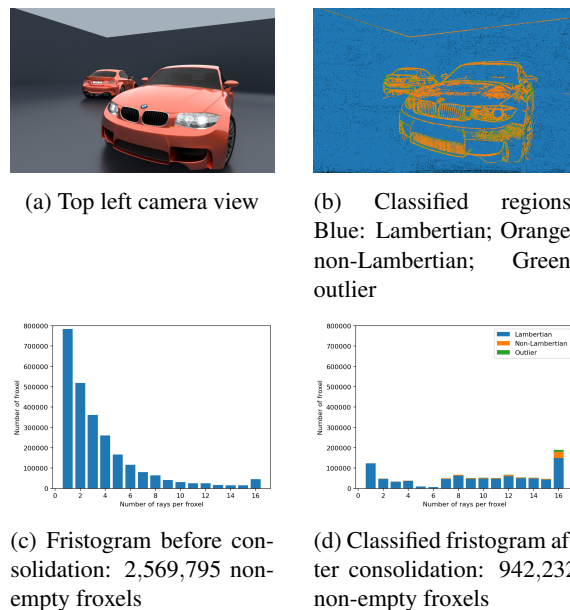


Figure 6: Blender BMW scene with depth maps extracted from NeRF

high quality depth maps are not always available especially for real world scenes[Luo20; Jan20; Kop21], the developed methods were also tested on depth maps acquired by other means. Specifically, we showcased the compatibility of our approach with NeRF by training a NeRF model, extracting the corresponding depth maps, and using them into our pipeline.

Upon examining the fristogram of the BMW scene (see figure 6c) generated from the NeRF depth maps, it becomes evident that the shape is markedly different from that of the Classroom scene. Despite the fact that much of the scene is visible to all 16 cameras, the majority of the froxels are assigned fewer than four rays. An inspection of the scene reveals that a substantial portion of it consists of featureless, monotonous background, which presents inherent challenges in generating depth maps accurately from visual data [Sch16]. This leads to bad multi-view consistency, which artificially inflates the number of non-empty froxels. To address this issue, we leverage the techniques outlined in Chapter 5 to consolidate froxels. This drastically reduced the number of non-empty froxels and created a clear peak at 16 rays per froxel. Although, not as pronounced as previously small peaks at 8 and 12 rays per froxel are also visible. Looking at the resulting classification (see figure 6b) the background is correctly marked as Lambertian, while reflective features on the car are identified as non-Lambertian. This demonstrates that our method is capable of generating dense froxel representations that hold significant meaning, even in situations where access to the scene geometry is not available. Nevertheless, the information value that can be extracted increases with the quality of the depth maps.

Table 1: Information Content

Scene	Captured	Minimum
Classroom	182,664,675 bits	176,354,697 bits
BMW	178,224,039 bits	176,354,697 bits
Painter	182,190,389 bits	170,124,571 bits

Furthermore, we showcase the potential of the froxel representation for real-world scenes. As an example, Figure 7 depicts the Painter scene sourced from Inter-Digital [Sab17], which was captured using a 4-by-4 light field array. This scene is listed in the MPEG-I content database and comes supplied with depth maps. An examination of the fristogram (refer to Figure 7b) reveals distinct peaks at 16 rays per froxel indicating that most of the scene is sampled by all 16 cameras. Peaks at 12, 8, and 4 rays per froxel suggest occlusions that are roughly aligned to the camera pattern of the light field array. These regions can be seen in figure 7d. Looking at the distribution of Lambertian, non-Lambertian and outlier froxels it becomes evident, that these scene contains many more than the previous two. This is a consequence caused by the limitations of color matching between cameras and the fact that real objects always exhibit at least some level of Lambertian reflectance [Geo07]. Therefore, the classification thresholds could be adjusted for real world scenes, but for the sake of comparison, they were kept the same.

Calculating the information content for each scene, based on the method described in chapter 6, reveals the additional information captured due to occlusions and non-Lambertian surfaces. Table 1 displays the result for the three discussed scenes. The listed minimum information content would be achieved if all cameras captured exactly the same information (e.g. a Lambertian wall a depth infinity) and therefore is only depends on the total number of rays and cameras. The Classroom and BMW scenes were captured using the same virtual light field camera, enabling direct comparison of their results. Notably, the Classroom scene exhibits a significantly higher information content due to a larger number of occlusions compared to the BMW scene.

This semantic analysis can be used to guide further post processing steps. As an example a surface aware ray reduction was implemented. This is based on the idea that a Lambertian surface can be accurately described with a single view-independent sample, while non-Lambertian surfaces require multiple samples. In practice, the rays assigned to a Lambertian froxel were filtered using a mean filter, whereas those assigned to non-Lambertian froxels remained unaltered. The original views of the light field were generated using this reduced set of froxels and compared against views generated using all rays, as well as ones generated using

Table 2: Impact of ray reduction on visual quality

Method	Classroom			BMW		
	all rays	one sample	Ours	all rays	one sample	Ours
PSNR \uparrow	30.460	29.400	30.180	36.360	32.890	35.620
SSIM \uparrow	0.9153	0.8882	0.9076	0.9801	0.9677	0.9778
LPIPS \downarrow	0.0596	0.0922	0.0689	0.0276	0.0489	0.0357
Ray Count	9.21 M	1.1 M	3.22 M	9.21 M	1.03 M	1.81 M

only one sample per froxel. The results of the proposed ray reduction technique are presented in Table 2. It can be observed that the visual quality achieved with the reduced set of rays is comparable to that of the unfiltered representation, while containing significantly fewer rays. Although the representation that utilizes only one sample per froxel contains even fewer rays, it results in notably lower quality.

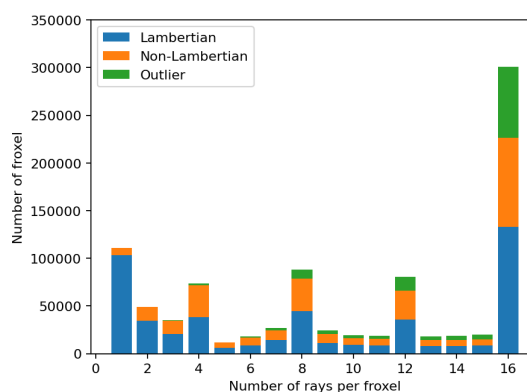
We evaluated the entire processing pipeline on supplementary Blender scenes, such as "The Wanderer" by Daniel Bystedt and "Mr. Elephant" by Glenn Meilenhorst, yielding consistent results. Obtaining further real-world data posed challenges due to the limited availability of suitable datasets.

8 CONCLUSION

In this paper we demonstrated how the froxel representation can be leveraged to perform semantic analysis of the information contained within a light field. Specifically, we illustrated methods for quantifying the sampling density of a captured scene and classifying surface properties. Rather than challenging methods like NeRF, that prioritize novel view reconstruction, our proposed approach instead enables visualization and quantization of the information distribution. This can be leveraged to effectively adapt post-processing steps to the available data. This enables creative professionals to understand the types of processing feasible with the acquired data, while also facilitating efficient light field encoding. One such application was demonstrated with a surface property aware ray reduction. Furthermore, we showed that our pipeline is robust against imperfect depth maps and can be applied to real-world scenes. A limitation, that the current pipeline shares with MPEG Immersive Video (MIV) is the assumption that the region between the cameras and the scene hull is free space. While in theory the froxel parameterization is capable of handling a more nuanced representation, this limitation is due to the fact, that the used depth maps only assign one specific depth to each ray. This limitation could be overcome by utilizing more complex depth formats and would permit better analysis of complex visual phenomena such as fog. Moreover, the presented method of semantic analysis is compatible with the notion of time, enabling the analysis of light fields video (e.g. quantify the difference in information content captured by sub-framing).



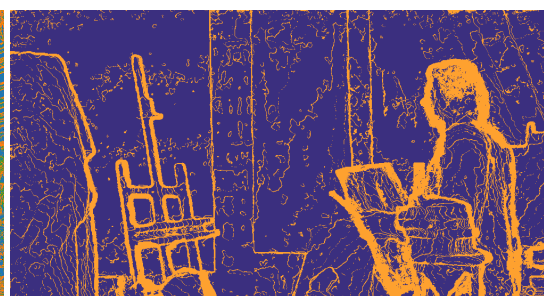
(a) Top left camera view



(b) Classified Fristogram after consolidation



(c) Visualization of the classified regions. Blue: Lambertian; Orange: non-Lambertian; Green: outlier



(d) Visualization of occluded regions. Violet: no occlusions captured; Orange: occlusions captured

Figure 7: Example visualization of the painter scene from InterDigital [Sab17]

9 ACKNOWLEDGMENT

This work is supported by the German Research Foundation (DFG) under grant number 429078454.

REFERENCES

- [Ani21] Yuriy Anisimov, Gerd Reis, and Didier Stricker. “Calibration and Auto-Refinement for Light Field Cameras”. In: *arXiv preprint arXiv:2106.06181* (2021).
- [Bar21] Jonathan T Barron et al. “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5855–5864.
- [Bar22] Jonathan T Barron et al. “Mip-nerf 360: Unbounded anti-aliased neural radiance fields”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5470–5479.
- [Ber91] James R Bergen and Edward H Adelson. “The plenoptic function and the elements of early vision”. In: *Computational models of visual processing* 1 (1991), p. 8.
- [Bis09] Tom E Bishop, Sara Zanetti, and Paolo Favaro. “Light field superresolution”. In: *2009 IEEE International Conference on Computational Photography (ICCP)*. IEEE. 2009, pp. 1–9.
- [Bro20] Michael Broxton et al. “Immersive light field video with a layered mesh representation”. In: *ACM Transactions on Graphics (TOG)* 39.4 (2020), pp. 86–1.
- [Cam98] Emilio Camahort, Apostolos Leros, and Donald S Fussell. “Uniformly sampled light fields.” In: *Rendering Techniques* 98 (1998), pp. 117–130.
- [Cam99] Emilio Camahort and Don Fussell. “A geometric study of light field representations”. In: *Technical Report TR99-35* (1999).
- [Che20] Kelvin Chelli et al. “A Versatile 5D Light Field Capturing Array”. In: *NEM Summit 2020*. 2020.
- [Dan15] Donald G. Dansereau, Oscar Pizarro, and Stefan B. Williams. “Linear Volumetric Focus for Light Field Cameras”. In: *ACM Transactions on Graphics (TOG)* 34.2 (2015).

- [Eva15] Alex Evans. "Learning from failure: A Survey of Promising, Unconventional and Mostly Abandoned Renderers for 'Dreams PS4', a Geometrically Dense, Painterly UGC Game. SIGGRAPH, 2015.
- [Fly19] John Flynn et al. "Deepview: View synthesis with learned gradient descent". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2367–2376.
- [Gar21] Stephan J Garbin et al. "Fastnerf: High-fidelity neural rendering at 200fps". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 14346–14355.
- [Geo07] Georgi T Georgiev and James J Butler. "Long-term calibration monitoring of Spectralon diffusers BRDF in the air-ultraviolet". In: *Applied Optics* 46.32 (2007), pp. 7892–7899.
- [Her21] Thorsten Herfet et al. "Fristograms: Revealing and Exploiting Light Field Internals". In: *arXiv preprint arXiv:2107.10563* (2021).
- [Hua22] Xin Huang et al. "Hdr-nerf: High dynamic range neural radiance fields". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18398–18408.
- [Jan20] Joel Janai et al. "Computer vision for autonomous vehicles: Problems, datasets and state of the art". In: *Foundations and Trends® in Computer Graphics and Vision* 12.1–3 (2020), pp. 1–308.
- [Kop14] Sanjeev J. Koppal. "Lambertian Reflectance". In: *Computer Vision: A Reference Guide*. Ed. by Katsushi Ikeuchi. Boston, MA: Springer US, 2014, pp. 441–443. ISBN: 978-0-387-31439-6.
- [Kop21] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. "Robust consistent video depth estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1611–1621.
- [Lam03] Edmund Y Lam. "Image restoration in digital photography". In: *IEEE Transactions on Consumer Electronics* 49.2 (2003), pp. 269–274.
- [Le 19] Mikael Le Pendu, Christine Guillemot, and Aljosa Smolic. "A fourier disparity layer representation for light fields". In: *IEEE Transactions on Image Processing* 28.11 (2019), pp. 5740–5753.
- [Lev96] Marc Levoy and Pat Hanrahan. "Light field rendering". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996, pp. 31–42.
- [Lib19] Orly Liba et al. "Handheld mobile photography in very low light." In: *ACM Trans. Graph.* 38.6 (2019), pp. 164–1.
- [Luo20] Xuan Luo et al. "Consistent video depth estimation". In: *ACM Transactions on Graphics (ToG)* 39.4 (2020), pp. 71–1.
- [Mil19] Ben Mildenhall et al. "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–14.
- [Mil20] Ben Mildenhall et al. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *ECCV*. 2020.
- [Mil21] Ben Mildenhall et al. "NeRF in the Dark: High Dynamic Range View Synthesis from Noisy Raw Images". In: *arXiv* (2021).
- [MPE18] WG 11 MPEG-I. *MPEG-I Phase 1 Use Cases (v1.5)*. Standard. International Organization for Standardization, 2018.
- [Mül22] Thomas Müller et al. "Instant neural graphics primitives with a multiresolution hash encoding". In: *ACM Transactions on Graphics (ToG)* 41.4 (2022), pp. 1–15.
- [Par21] Keunhong Park et al. "Nerfies: Deformable neural radiance fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5865–5874.
- [Pum21] Albert Pumarola et al. "D-nerf: Neural radiance fields for dynamic scenes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 10318–10327.
- [Sab17] Neus Sabater et al. "Dataset and Pipeline for Multi-View Light-Field Video". In: *CVPR Workshops*. 2017.
- [Sam21] Jaroslaw Samelak et al. "Efficient Immersive Video Compression using Screen Content Coding". In: (2021).
- [Sch09] Heidrun Schaaf et al. "evolution of photography in maxillofacial surgery: from analog to 3D photography—an overview". In: *Clinical, cosmetic and investigational dentistry* (2009), pp. 39–45.
- [Sch16] Johannes Lutz Schönberger and Jan-Michael Frahm. "Structure-from-Motion Revisited". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

- [Sha17] Gaurav Sharma and Raja Bala. *Digital color imaging handbook*. CRC press, 2017.
- [Sha48] Claude E Shannon. “A mathematical theory of communication”. In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.
- [Wil05] Bennett Wilburn et al. “High performance imaging using large camera arrays”. In: *ACM SIGGRAPH 2005 Papers*. 2005, pp. 765–776.
- [Xia21] Wenqi Xian et al. “Space-time neural irradiance fields for free-viewpoint video”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9421–9431.
- [Yu21] Alex Yu et al. “pixelnerf: Neural radiance fields from one or few images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4578–4587.

Texture Spectral Similarity Criteria Comparison

Michal Havlíček

Institute of Information
Theory and Automation,
Czech Academy of
Sciences
Pod Vodarenskou vezi 4
182 08 Prague, Czechia
havlimi2@utia.cas.cz

Michal Haindl

Institute of Information
Theory and Automation,
Czech Academy of
Sciences
Pod Vodarenskou vezi 4
182 08 Prague, Czechia
haindl@utia.cas.cz
&
Faculty of Management,
University of Economics,
Jarošovská 1117,
Jindřichuv Hradec,
Czechia

ABSTRACT

Criteria capable of texture spectral similarity evaluation are presented and compared. From the fifteen evaluated criteria, only four criteria guarantee zero or minimal spectral ranking errors. Such criteria can support texture modeling algorithms by comparing the modeled texture with corresponding synthetic simulations. Another possible application is the development of texture retrieval, classification, or texture acquisition system. These criteria thoroughly test monotonicity and mutual correlation on specifically designed extensive monotonously degrading experiments.

Keywords

Texture Comparison, Texture Modeling, Texture Retrieval, Texture Classification, Texture Acquisition

1 INTRODUCTION

An automatic texture comparison represents a significant but not completely solved complex problem [Hai14]. Such a method would be advantageous to support texture model development where a comparison of the original acquired and to be modeled texture with synthesized or reconstructed ones would help with the optimal model parameter set. There are other possible applications, such as texture database retrieval or texture classification or segmentation, etc. Although there already exist approaches for these tasks, e.g., [Har73, Gal75, Law80, Wys82, Man96, Oja02, Hai06], etc., they do not rank textures according to their visual similarity. Moreover, most methods are limited to mono-spectral textures, a notable disadvantage as color is the most significant visual feature [Hav19].

The psycho-physical evaluations [Hai12], i.e., quality assessments performed by humans, currently represent the only reliable alternative. Methods of this type require both time-demanding experiment design setup and performing, rigorously defined and controlled conditions, and a representative collection of testers, i.e., a sufficient number of individuals, ideally from the general public, naive concerning the goal and design of the experiment. Therefore such experiments are highly impractical and generally demanding, and they cannot be performed on a daily base, on demand, or even in real-time. These experiments are also impracticable in the case of hyper-spectral textures, as not all spectra can be visualized simultaneously due to the limited trichromatic nature of the human perception system.

The criteria mentioned and compared in this paper are intended for the spectral texture composition comparison, i.e., for a specific subset of the general texture comparison problem. The textures are compared as independent sets of pixels where the pixel are treated as vectors of real vector space while the positions of the pixels in the textures are not considered. Texture spectral composition comparison deals with the appearance and amount of pixels that occur in only one of the compared textures and also with the ratio of occurrences of pixels appearing in both textures.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

The objectives of the study are as follows:

- To study the effectiveness of spectral similarity criteria for textural applications.
- To analyze their mutual substitutability.

The rest of the paper is organized as follows. Section 2 briefly reviews existing methods relevant to the texture spectral composition comparison. Section 3 outlines experiments used to compare individual methods presented in section 2. Section 3 presents and comments achieved results. Section 4 summarizes the paper with a discussion.

2 TEXTURE SPECTRAL SIMILARITY CRITERIA

In this section, we briefly survey existing texture spectral composition comparison methods. The straightforward way is to use an n-dimensional (n-D) histogram or local one [Yua15], approximating the spectral texture distribution.

Let A and B are the textures to be compared. We denote by a_ρ and b_ρ the ρ -th bin of the n-D histogram of the textures A and B , respectively. The range of the histogram multi-index $\rho = \rho_1, \rho_2, \dots, \rho_n$ depends on a space C , in which the texture is represented, e.g., in the case of the standard 24-bit red, green, and blue (RGB) color space, the range of all three components of the multi-index is an integer from 0 to 255.

The most intuitive way is to compute the n-D histogram block distance, also known as the Manhattan distance or the Minkowski distance:

$$\Delta_q H(A, B) = \left(\sum_{\rho \in C} |a_\rho - b_\rho|^q \right)^{1/q}, \quad (1)$$

with $q = 1$ (histogram difference), $q = 2$ (Euclidean distance of histograms), $0 < q < 1$ (fractional dissimilarity of histograms) representing the most used variants. A special case is the maximum distance also called Chebyshev distance or chessboard distance:

$$\Delta_\infty H(A, B) = \sum_{\rho \in C} \max\{|a_{\rho_1} - b_{\rho_1}|, \dots, |a_{\rho_n} - b_{\rho_n}|\}. \quad (2)$$

Several other possibilities exist for n-D histogram comparison, such as the histogram intersection [Swa91]:

$$\cap H(A, B) = 1 - \frac{\sum_{\rho \in C} \min\{a_\rho, b_\rho\}}{\sum_{\rho \in C} b_\rho}, \quad (3)$$

the squared chord [Kok03]:

$$d_{sc}(A, B) = \sum_{\rho \in C} (\sqrt{a_\rho} - \sqrt{b_\rho})^2, \quad (4)$$

and the Canberra metric [Kok03]:

$$d_{can}(A, B) = \sum_{\rho \in C} \frac{|a_\rho - b_\rho|}{a_\rho + b_\rho}, \quad (5)$$

where $C_0 = \{\rho : a_\rho + b_\rho \neq 0\} \subset C$.

The information-theoretic measures like the Kullback-Leibler divergence [Kul51] can also be used:

$$KL(A, B) = \sum_{\rho \in C_0} a_\rho \log \frac{a_\rho}{b_\rho}, \quad (6)$$

with $C^0 = \{\rho : a_\rho b_\rho \neq 0\} \subset C$, or the Jeffrey divergence:

$$J(A, B) = \sum_{\rho \in C^0} a_\rho \log \frac{2a_\rho}{a_\rho + b_\rho} + b_\rho \log \frac{2b_\rho}{a_\rho + b_\rho}, \quad (7)$$

can be also considered for n-D histogram comparison as well as a measure based on χ^2 statistic [Zha03]:

$$\chi^2(A, B) = \sum_{\rho \in C_0} \frac{2 \left(a_\rho - \frac{a_\rho + b_\rho}{2} \right)^2}{a_\rho + b_\rho}. \quad (8)$$

The generalized color moments (GCM) [Min98] can also be useful for texture spectral composition comparison problems. The original definition of the GCM of the $(p+q)$ -th order and the $(\alpha+\beta+\gamma)$ -th degree is:

$$\Delta GCM_{pq}^{\alpha\beta\gamma}(A, B) = \int \int_{\langle A \rangle} r_1^p r_2^q [Y_{r_1, r_2, 1}^A]^\alpha [Y_{r_1, r_2, 2}^A]^\beta [Y_{r_1, r_2, 3}^A]^\gamma dr_1 dr_2 - \int \int_{\langle B \rangle} r_1^p r_2^q [Y_{r_1, r_2, 1}^B]^\alpha [Y_{r_1, r_2, 2}^B]^\beta [Y_{r_1, r_2, 3}^B]^\gamma dr_1 dr_2, \quad (9)$$

where $[r_1, r_2] \in \langle A \rangle$ represents planar coordinates of the texture pixel Y_r^A , $Y_{r_1, r_2, i}^A$ denotes a pixel intensity in the i -th spectral channel of the texture A , similarly $Y_{r_1, r_2, i}^B$ where $[r_1, r_2] \in \langle B \rangle$. GCM can be easily re-defined for an arbitrary number of spectral channels. The terms r_1^p and r_2^q are meaningless in the case of texture spectral composition comparison, and therefore both are put equal to one, using GCMs for which $p = q = 0$ holds. Moreover, it has been observed that the best results are achieved if $\alpha = \beta = \gamma$, specifically using GCMs for $\alpha = \beta = \gamma < 4$ [Hav19].

Another possibility for texture spectral composition comparison represents cosine-function-based dissimilarity, which computes an angle between two vectors.



Figure 1: Textures used for experiments.

Both A, B must have the same number of pixels, a significant drawback of this criterion. This criterion is the only one mentioned in this article suffering from this. All intensity values of corresponding texture spectral channels of all pixels of the textures are arranged into vectors \vec{A} and \vec{B} and the difference is computed as [Zha03]:

$$d_{cos}(A, B) = \frac{\vec{A}^T \vec{B}}{|\vec{A}| |\vec{B}|} . \quad (10)$$

Various set-theoretic measures can be considered as criteria as well. Let sets \mathcal{A} and \mathcal{B} denotes the set of unique multi-dimensional vectors representing pixels occurring in the texture A and B , respectively. Criteria can be based on methods developed for comparing the similarity and diversity of the sample sets, such as the Jaccard index [Jac01]:

$$JI(A, B) = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|} , \quad (11)$$

or the Sørensen-Dice index [Dic45]:

$$SDI(A, B) = \frac{2 |\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A}| + |\mathcal{B}|} . \quad (12)$$

JI and SDI are equivalent in the sense that given a value for SDI, one can calculate the respective JI value and vice versa.

Alternative to the existing methods may be a modified criterion developed for the texture comparison as the spectral texture composition comparison is its exceptional case. It is possible to remove structure term from the structural similarity metric (SSIM) [Wan04] and define reduced SSIM [Hav16]:

$$rSSIM(A, B) = \frac{1}{\#\{r_3\}} \sum_{\forall r_3} \frac{2\mu_{A,r_3}\mu_{B,r_3}}{\mu_{A,r_3}^2 + \mu_{B,r_3}^2} \frac{2\sigma_{A,r_3}\sigma_{B,r_3}}{\sigma_{A,r_3}^2 + \sigma_{B,r_3}^2} , \quad (13)$$

where $\#\{r_3\}$ is the spectral index cardinality, i.e., the number of spectral channels, μ_{A,r_3} is the mean of r_3 -th spectral plane of A and σ_{A,r_3} is the standard deviation of r_3 -th spectral plane of A and similarly for μ_{B,r_3} and σ_{B,r_3} .

A very accurate method, the mean exhaustive minimum distance (MEMD), was introduced in [Hav19]. MEMD can be described as the following algorithm. For each pixel from A , the most similar pixel from B is found. This pixel from B can be identified as the most similar to an arbitrary one from A only once. The evaluation ends when all pixels from A have their counterparts in B or all pixels from B are identified as the most similar pixel for an arbitrary one from A . The similarity can

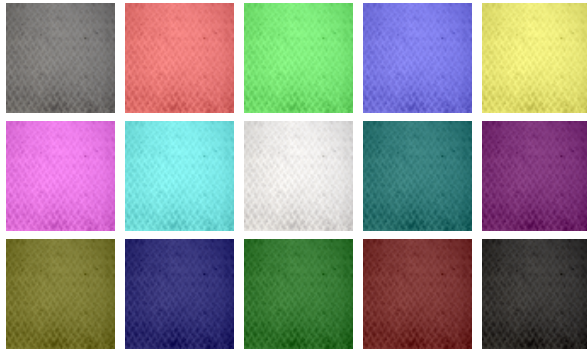


Figure 2: Example of tested texture (top-left) and its the most modified (final) versions obtained during fourteen individual experiments with it.

be expressed by arbitrary metric ρ . The best results were obtained using the maximum metric. The values of metrics are summed and then divided by the number of compared pixels which equals the minimum of the number of pixels in A and the number of pixels in B denoted as M , i.e.:

$$MEMD(A, B) = \quad (14)$$

$$\frac{1}{M} \sum_{(r_1, r_2) \in \langle A \rangle} \min_{(s_1, s_2) \in \langle U \rangle} \{ \rho(Y_{r_1, r_2, \bullet}^A, Y_{s_1, s_2, \bullet}^B) \},$$

where $Y_{r_1, r_2, \bullet}^A$ denotes pixel at $(r_1, r_2) \in \langle A \rangle$, similarly for $Y_{s_1, s_2, \bullet}^B$ but $(s_1, s_2) \in \langle U \rangle$, where $\langle U \rangle$ represents the set of the planar coordinates of the pixels from B not identified as the most similar pixel for the pixels from A evaluated before the pixel at (r_1, r_2) .

This criterion was optimized by applying a quicksort sorting algorithm on input data [Hav21]. It is also possible to decrease evaluating time by not including pixels with the exact location in both compared textures if the intensity values are the same in both textures in the corresponding spectral channels. This optimization is meaningful when both textures have the same size, and their difference is expected in the number of pixels, which is significantly smaller than the texture size. It is optional for the locations of such pixels to be known in advance.

3 COMPARISON

All criteria mentioned in the previous section have been extensively tested on precisely defined experiments. The basic idea was to gradually modify the original texture (Figure 1) to resemble the original texture steadily less. The criterion should be able to track these changes to rate the more modified versions of the original texture as less similar to the original. The evaluation error is the ratio of the number of such violations of the assumed monotony to the number of

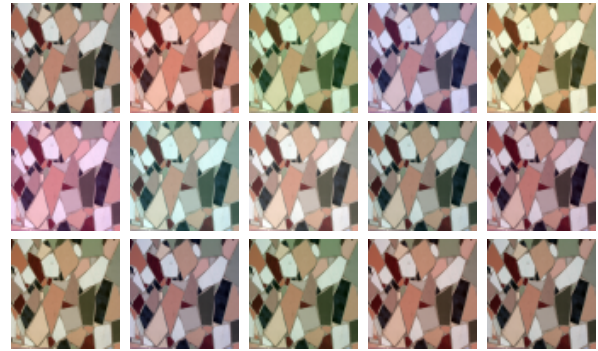


Figure 3: Example of tested texture (top-left) and its the most modified (final) versions obtained during fourteen individual experiments with it.

Criterion	Error [%]	Rank
$\Delta_1 H(\cdot)$	57.0	8
$\Delta_2 H(\cdot)$	57.1	9
$\Delta_{0.5} H(\cdot)$	56.7	7
$\cap H(\cdot)$	57.0	8
$d_{sc}(\cdot)$	56.3	4
$d_{can}(\cdot)$	56.6	6
$KL(\cdot)$	73.1	11
$J(\cdot)$	69.5	10
$\chi^2(\cdot)$	56.4	5
$\Delta GCM_{00}^{111}(\cdot)$	0.0	1
$d_{cos}(\cdot)$	1.1	2
$JI(\cdot)$	48.8	3
$SDI(\cdot)$	59.6	10
$rSSIM(\cdot)$	0.0	1
$MEMD(\cdot)$	0.0	1

Table 1: Average error over all experiments and all textures for individual criteria and corresponding ranks.

modified versions of the original texture. It should be possible to create modifications that are detectable by the criterion but imperceptible to the human observer. Criteria that can detect even such changes are another advantage over psychophysical experiments and also have possible practical use in areas where maximum accuracy higher than that achievable by a human observer is welcome [Lac22]. Based on these requirements, adjustments were proposed to add or subtract the minimum possible value to all intensity values in selected spectral channels for all texture pixels at once, e.g., Figures 2,3. So that in the case of RGB color space, it is possible to modify data in a single channel, in two channels at the same time, or in all channels at the same time resulting in 14 experiments. In the case of used RGB color space, the minimum possible value that can be added or subtracted equals one, and the adding or subtracting is stopped when maximum, i.e., 255, or minimum, i.e., 0, respectively, is reached

criterion	$\Delta_1 H$	$\Delta_2 H$	$\Delta_{0.5} H$	$\cap H$	d_{sc}	d_{can}	KL	J	χ^2	ΔGCM_{00}^{111}	d_{cos}	JI	SDI	$rSSIM$
$\Delta_1 H$														
$\Delta_2 H$	0,99													
$\Delta_{0.5} H$	0,99	0,96												
$\cap H$	1,00	0,99	0,99											
d_{sc}	0,99	0,97	1,00	0,99										
d_{can}	-0,99	0,96	1,00	0,99	1,00									
KL	-0,56	-0,44	-0,68	-0,56	-0,64	-0,66								
J	0,71	-0,61	-0,81	-0,71	-0,78	-0,79	0,97							
χ^2	1,00	0,98	0,99	1,00	1,00	1,00	-0,60	-0,75						
ΔGCM_{00}^{111}	-0,66	-0,60	-0,72	-0,66	-0,69	-0,72	0,71	0,73	-0,68					
d_{cos}	-0,72	-0,65	-0,78	-0,72	-0,75	-0,78	0,76	0,78	-0,73	0,99				
JI	-0,99	-1,00	-0,96	-0,99	-0,97	-0,96	0,44	0,61	-0,98	0,61	0,66			
SDI	-0,87	-0,82	-0,92	-0,87	-0,90	-0,92	0,78	0,85	-0,88	0,90	0,94	0,82		
$rSSIM$	-0,68	-0,61	-0,74	-0,68	-0,71	-0,74	0,74	0,76	-0,69	1,00	1,00	0,62	0,92	
$MEMD$	0,76	0,70	0,81	0,76	0,79	0,81	-0,75	-0,79	0,77	-0,99	-1,00	-0,71	-0,95	-0,99

Table 2: The color criteria Pearson correlation over all 161 materials.

for any intensity value of any pixel. This additional requirement is introduced as a prevention against data overflow or underflow, which could lead to a distortion of the results in the sense that increasing dissimilarity from the original texture would no longer be guaranteed. The number of textures generated by gradually modifying the original texture differs for each texture and depends on the values of the pixel intensities in the original texture. Examples of generated textures to compare with the original can be seen in Figures 2,3.

One hundred sixty-one color textures with resolution 64×64 saved as 24-bit RGB portable network graphics (PNG) files were used as the original textures covering a wide range of natural and artificial materials. Textures were obtained from accessible texture databases^{1,2}, and they are shown in Figure 1.

4 RESULTS

Input data used in the experiments described in the previous section led to 78 647 texture-to-texture comparisons for each tested criterion. Achieved results are presented in Table 1. There is an average error over all experiments, and all textures and corresponding ranks are presented for all tested criteria. It is clear from these results that although the tested criteria seem to be theoretically used for texture spectral composition comparison, they rather fail in this task. All histogram-based criteria and set-theoretic measure-based ones reach an error rate of around 50.0% and an even significantly higher error rate in the case of information-theoretic measure-based criteria. One of the reasons might be that our degradation experiments modify non-linearly

histograms, often in unpredictable manners, while individual pixels are distorted linearly. But many real-world image degradations can be approximated using linear pixel modifications. On the other hand, four criteria meet the requirements for a credible method for texture spectral composition comparison as their error rate is 1.1% (d_{cos}) or even 0.0% (ΔGCM_{00}^{111} , $rSSIM$ and $MEMD$). The target of our paper is not to compare textures. Thus we do not consider here any geometric transformations.

Table 2 illustrates Pearson correlation between all pairs of criteria. The best criteria ($MEMD$, $rSSIM$, ΔGCM_{00}^{111} , d_{cos}) are mutually highly correlated ($rSSIM \times \Delta GCM_{00}^{111}$, $rSSIM \times d_{cos}$, $MEMD \times \Delta GCM_{00}^{111}$, $MEMD \times d_{cos}$, $MEMD \times rSSIM$). Similarly, the histogram criteria ($\Delta_1 H(\cdot)$, $\Delta_2 H(\cdot)$, $\Delta_{0.5} H(\cdot)$, $\cap H(\cdot)$), the squared chord $d_{sc}(\cdot)$, and χ^2 are also correlated.

The highly correlated criteria are thus mutually interchangeable.

5 CONCLUSIONS

We properly tested criteria potentially useful for texture spectral composition comparison and demonstrated their suitability in a specially designed experiment. The texture spectral composition comparison represents a partial solution for assessing the textures' quality. Although the criteria do not consider the location of the pixels in the textures, they can help in numerous texture analysis or synthesis applications. The best three criteria - $MEMD$, generalized color moments, and our reduced structural similarity metric perform with zero spectral ranking errors, while the cosine criterion has a tiny error only. These criteria can be used mainly as a reliable, fully automatic alternative to psychophysical experiments, which are more

¹ texturer.com

² mayang.com

impractical due to their cost and strict demands on design setup, conditions control, human resources, and time. Additionally, psychophysical experiments are restricted to visualization of the maximum of 3-D data due to the limited trichromatic nature of human vision, while the MEMD criterion has no upper limit for possible spectral bands.

ACKNOWLEDGMENTS

The Czech Science Foundation project GAČR 19-12340S supported this research.

6 REFERENCES

- [Dic45] Dice, L.R. Measures of the amount of ecological association between species. *Ecology*, 26, (3), pp.297-302, 1945.
- [Gal75] Galloway, M.M. Texture analysis using gray level run lengths. *Comput. Graph. Image Process.*, 4, (2), pp.172-179, 1975.
- [Hai12] Haindl, M., Filip, J. Visual texture. in Singh, S., Kang, S.B. (Eds.): *Advances in computer vision and pattern recognition*, Springer-Verlag London, London, 2012.
- [Hai14] Haindl, M., Kudělka, M. Texture fidelity benchmark. in *Computational Intelligence for Multimedia Understanding (IWCIM)*, 2014 International Workshop on, IEEE Computer Society CPS, Los Alamitos, pp.1-5, 2014.
- [Hai06] Haindl, M., Mikeš, S. Unsupervised texture segmentation using multispectral modelling approach. *Proc. 18th Int. Conf. Pattern Recognition ICPR 2006*, vol. II, pp.203-206, 2006.
- [Har73] Haralick, R.M., Shanmugam, K., Dinstein, I. Textural features for image classification. *IEEE Trans. Syst. Man Cybern.*, SMC-3, (6), pp.610-621, 1973.
- [Hav16] Havlíček, M., Haindl, M. Texture spectral similarity criteria. *Proceedings of the 4th CIE Expert Symposium on Colour and Visual Appearance*, Commission Internationale de L'Eclairage CIE Central Bureau, Vienna, pp.147-154, 2016.
- [Hav19] Havlíček, M., Haindl, M. Texture spectral similarity criteria. *IET Image Processing*, 13(11), pp.1998-2007, 2019.
- [Hav21] Havlíček, M., Haindl, M. Optimized texture spectral similarity criteria. In *Advances in Computational Collective Intelligence: 13th International Conference, ICCCI 2021, Kallithea, Rhodes, Greece, September 29-October 1, 2021, Proceedings 13*, Springer International Publishing, pp.644-655, 2021.
- [Jac01] Jaccard, P. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, Basel, Switzerland, 1901.
- [Kok03] Kokare, M., Chatterji, B., Biswas, P. Comparison of similarity metrics for texture image retrieval. In: *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*. Vol. 2. IEEE, pp.571-575, 2003.
- [Kul51] Kullback, S., Leibler, R.A. On information and sufficiency. *Annals of Mathematical Statistics*. 22 (1), pp. 79-86, doi:10.1214/aoms/1177729694, JSTOR 2236703. MR 0039968, 1951.
- [Lac22] Lachaud, G., Conde-Cespedes, P., Trocan, M. Patch Selection for Melanoma Classification. In *Computational Collective Intelligence: 14th International Conference, ICCCI 2022, Hammamet, Tunisia, September 28-30, 2022, Proceedings*, Cham: Springer International Publishing, 2022.
- [Law80] Laws, K.I. Rapid texture identification. *Proc. SPIE Conf. Image Processing for Missile Guidance*, San Diego, USA, pp.376-380, 1980.
- [Man96] Manjunath, B.S., Ma, W.Y. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18, (8), pp.837-842, 1996.
- [Min98] Mindru, F., Moons, T., Gool, L. V. Color-based moment invariants for viewpoint and illumination independent recognition of planar color patterns. In: *Illumination Independent Recognition of Planar Color Patterns*, *Proceedings ICAPR'98*. pp.113-122, 1998.
- [Oja02] Ojala, T., Pietikäinen, M., Mäenpää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24, (7), pp.971-987, 2002.
- [Swa91] Swain, M. J., Ballard, D. H. Color indexing. *International Journal of Computer Vision* 7 (1), pp.11-32, 1991.
- [Wan04] Wang, Z., Bovik, A.C., Sheikh, H.R., et al. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13, (4), pp.600-612, 2004.
- [Wys82] Wyszecki, G., Stiles, W.S. *Color science*, vol. 8, Wiley, New York, 1982.
- [Yua15] Yuan, J., Wang, D., Cheriadat, A.M. Factorization-based texture segmentation. *IEEE Trans. Image Process.*, 24, (11), pp.3488-3497, 2015.

- [Zha03] Zhang, D., Lu, G. Evaluation of similarity measurement for image retrieval. In: Neural Networks and Signal Processing, Proceedings of the 2003 International Conference on. Vol. 2. IEEE, pp.928-931, 2003.

Designing a Lightweight Edge-Guided Convolutional Neural Network for Segmenting Mirrors and Reflective Surfaces

Mark Edward M. Gonzales

De La Salle University
Taft Avenue, Malate
Manila 1004, Philippines

mark_gonzales@dlsu.edu.ph

Lorene C. Uy

De La Salle University
Taft Avenue, Malate
Manila 1004, Philippines

lorene_c_uy@dlsu.edu.ph

Joel P. Ilao

De La Salle University
Taft Avenue, Malate
Manila 1004, Philippines

joel.ilao@dlsu.edu.ph

ABSTRACT

The detection of mirrors is a challenging task due to their lack of a distinctive appearance and the visual similarity of reflections with their surroundings. While existing systems have achieved some success in mirror segmentation, the design of lightweight models remains unexplored, and datasets are mostly limited to clear mirrors in indoor scenes. In this paper, we propose a new dataset consisting of 454 images of outdoor mirrors and reflective surfaces. We also present a lightweight edge-guided convolutional neural network based on PMDNet. Our model uses EfficientNetV2-Medium as its backbone and employs parallel convolutional layers and a lightweight convolutional block attention module to capture both low-level and high-level features for edge extraction. It registered F_β scores of 0.8483, 0.8117, and 0.8388 on the Mirror Segmentation Dataset (MSD), Progressive Mirror Detection (PMD) dataset, and our proposed dataset, respectively. Applying filter pruning via geometric median resulted in F_β scores of 0.8498, 0.7902, and 0.8456, respectively, performing competitively with the state-of-the-art PMDNet but with $78.20\times$ fewer floating-point operations per second and $238.16\times$ fewer parameters. The code and dataset are available at <https://github.com/memgonzales/mirror-segmentation>.

Keywords

Mirror segmentation, object detection, convolutional neural network (CNN), CNN filter pruning

1 INTRODUCTION

Despite the ubiquitous presence of mirrors and reflective surfaces in everyday scenes — from indoor rooms to outdoor buildings — existing computer vision systems have difficulty detecting them due to their lack of a consistent distinguishing appearance and the visual similarity of reflections with their surroundings [Par21]. This results in complications in tasks such as robot navigation [And18] and three-dimensional scene reconstruction [Zha18], where approaches to accommodate the presence of mirrors entail having to augment visual information from cameras with cues from specialized hardware, including ultrasonic sensors and dedicated illumination devices [Tin16].

Mirrors and reflective surfaces also pose potential hazards to autonomous driving and driver assistance systems that rely on stereo vision since they can cause glare spots, irregularly distorted reflections, and infinite

reflections [Zen17]. These challenges are pronounced given the presence of safety mirrors in road and parking space junctions, as well as large reflective glass surfaces in the façades of several high-rise buildings. Hence, developing systems that can reliably recognize and localize them is critical to autonomous navigation.

While general object detection and segmentation frameworks have achieved success in various applications [He17, Zha17], they are unable to satisfactorily distinguish reflections from the actual objects. Consequently, directly applying them to mirror detection has yielded subpar results, as the reflections also tend to get segmented [Yan19]. Meanwhile, salient object detection techniques may not necessarily tag mirrors as salient [Yan19, Lin20a].

In this regard, the segmentation of mirrors and reflective surfaces posits itself as a challenging task that necessitates tailored approaches. Early works focused on exploiting contrasts and relationships between the contents inside and outside the mirror [Yan19, Lin20a]. Recently, depth [Mei21], semantic association with surrounding objects [Gua22], and visual chirality [Tan22] have also been explored to enrich the set of cues.

However, despite their success, designing lightweight mirror segmentation models remains an unexplored direction. Most systems have over 100 million param-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

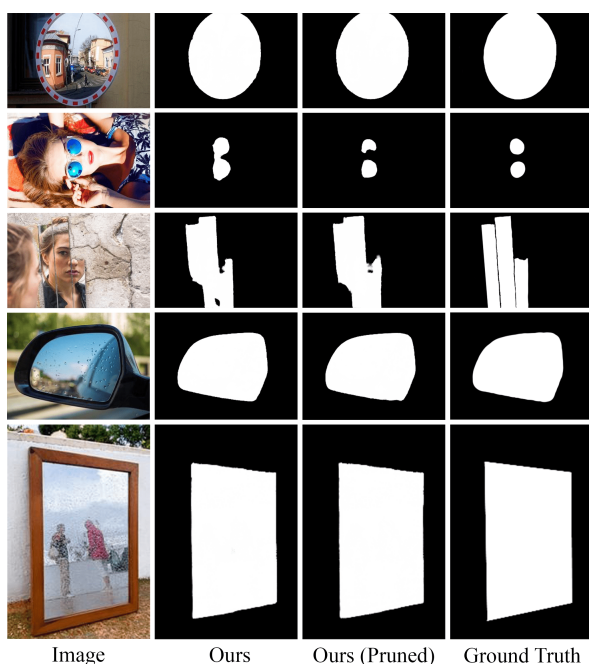


Figure 1: Existing datasets consist mostly of clear indoor mirrors. Our proposed dataset focuses on outdoor mirrors and reflective surfaces of varying shapes and sizes (first column). Our edge-guided CNN and its pruned version perform competitively with the state-of-the-art. This pruned version is also lightweight and can be deployed to resource-constrained devices.

ters, with MirrorNet [Yan19], PMDNet [Lin20a], and SANet [Gua22] having 121.77, 147.66, and 105.84 million parameters, respectively. Existing datasets are also mostly limited to clear mirrors in indoor scenes; outdoor mirrors and reflective surfaces (e.g., tinted car windows and building façades) are not well represented. These may be prohibitive to the integration of models into resource-constrained devices, such as drones and autonomous navigation vehicles.

In an attempt to address these gaps, our study seeks to contribute the following:

- We propose a dataset of outdoor mirrors and reflective surfaces with 454 images and their corresponding ground-truth masks.
- We modified the architecture of PMDNet [Lin20a] and extensively tested different feature extraction backbones and edge-related modules to guide the segmentation.
- We pruned our best-performing edge-guided convolutional neural network, resulting in a lightweight model with 1.52 billion floating-point operations per second (FLOPS) and 0.62 million parameters. It performs competitively with the state-of-the-art PMDNet but with $78.20\times$ fewer FLOPS and $238.16\times$ fewer parameters.

2 RELATED WORKS

Early attempts to detect and segment mirrors require the assistance of specialized hardware [Whe18] or user interaction [Cha17]. The first model to perform the task given solely an RGB image input is MirrorNet [Yan19]. Using ResNeXt-101 [Xie17] as its multi-scale feature extraction backbone, content discontinuities inside and outside the mirror are captured via a dedicated contextual contrasted feature extraction module.

PMDNet [Lin20a] extends this by considering not only discontinuities but also similarities between the reflection and the surroundings via a dedicated module connected to the side-outputs of a ResNeXt-101 backbone. Moreover, an edge detection and fusion module captures both high-level and low-level features from the feature maps generated by the backbone. However, MirrorNet and PMDNet may have some difficulty handling cases where there are insufficient correlational features or contextual contrast, such as when the reflection occupies most of the image.

Recent studies have also investigated the integration of various cues. Adopting ResNet-50 [He16] as its backbone, PDNet [Mei21] captures not only RGB features but also depth. Aside from the limitations posed by the need for specialized hardware to capture depth, objects such as doorways may confuse its depth-aware module.

The scene-aware SANet [Gua22] capitalizes on semantic associations, i.e., the observed placement of mirrors together with certain objects for functional purposes. Since this approach relies on annotations, low-quality labels may affect performance. Annotated datasets may also be expensive to construct and may thus not be readily available for most real-world use cases.

VCNet [Tan22] frames visual chirality [Lin20b], the change in image statistics upon reflection, as a commutative residual. Similar to MirrorNet and PMDNet, it utilizes a ResNeXt-101 backbone. While its use of a visual chirality cue allows its edge detection module to learn features other than the conventional geometric properties, it has difficulty excluding small occluding objects and handling boundaries with complex shapes.

Our work builds on insights from these previous works and explores another direction by focusing on the construction of a lightweight model that is capable of performing competitively with the state-of-the-art. We also demonstrate the effectiveness of using EfficientNet [Tan19] as a promising and less computationally expensive alternative to the usual ResNeXt backbone used in existing mirror detection and segmentation models.

3 OUTDOOR MIRRORS AND REFLECTIVE SURFACES DATASET

Following previous works [Yan19, Lin20a, Mei21, Gua22, Tan22], we used two publicly available mirror datasets in our study: MSD [Yan19] and PMD

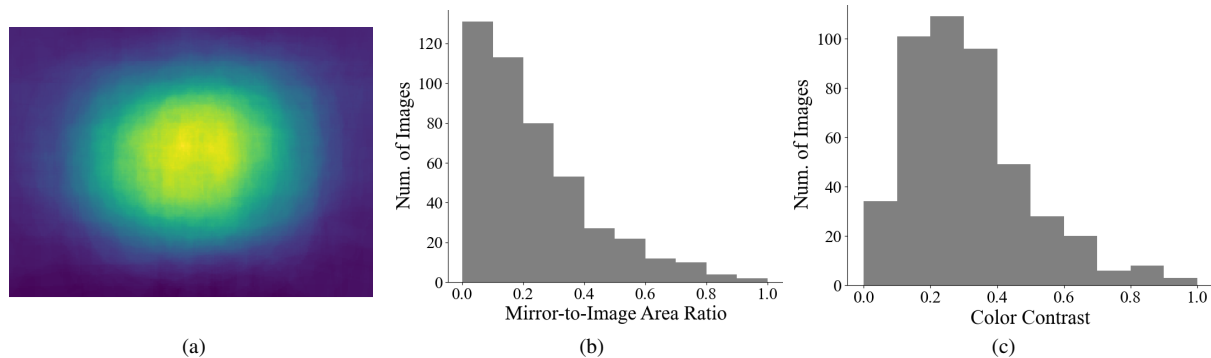


Figure 2: Dataset Statistics. (a) Distribution of the mirror location, with yellow corresponding to higher frequencies and blue corresponding to lower frequencies. (b) Mirror-to-image area ratio. (c) Color contrast between the mirror and the surrounding area, as measured by taking the χ^2 distance between their RGB histograms, following [Yan19].

[Lin20a]. MSD consists of 4018 images; however, most are zoomed-in images of indoor scenes that exhibit high similarity. PMD aggregates 6016 images from multiple datasets including ADE20K [Zho17] and NYUD-V2 [Sil12]. Although the images in PMD are more varied than those in MSD, outdoor mirrors and reflective surfaces remain underrepresented.

To help address this limitation, we propose the De La Salle University – Outdoor Mirrors and Reflective Surfaces (DLSU-OMRS) dataset. The images were scraped from Shutterstock using the key phrases *outdoor mirror* and *street mirror* and manually filtered to remove duplicates and heavily manipulated photos. Ground-truth masks were produced through manual segmentation. The DLSU-OMRS dataset contains 454 images, with an average structural similarity index of 28.67%. As characterized in Figure 2 and Table 1, most mirrors are located near the center and occupy up to 20% of the image. The color contrast [Yan19] of most images is also below 40%, which suggests that the contents inside the mirrors are visually similar to their surroundings, making our dataset more challenging.

4 MODEL CONSTRUCTION

4.1 Model Architecture

Using PMDNet as the base model (Figure 3a), we introduced two modifications in an attempt to improve performance and lower computational costs.

First, we explored seven feature extraction backbones that were pretrained on ImageNet [Den09]: ResNet-50 [He16], Xception-65 [Cho17], VoVNet-39 [Lee19], MobileNetV3 [How19], EfficientNetLite4 [Tan19], EfficientNet-Edge-Large (pruned following the lottery ticket hypothesis) [Tan19], and EfficientNetV2-Medium [Tan19]. These were selected in light of their application in object segmentation [Cha22, Lin22].

Second, we modified PMDNet’s edge detection and fusion module. While PMDNet extracts low-level edge

	Num. of Images
One Mirror	338
Multiple Mirrors	116
	Num. of Mirrors
By Shape	
Triangle	4
Quadrilateral	258
Polygonal (≥ 5 straight edges)	9
Round/Elliptical	160
Irregular	355
By Presence of Occlusion	
Present	192
Not Present	594

Table 1: Mirror Shape and Occlusion Statistics. For images with multiple mirrors, each mirror is categorized separately by shape and by the presence of an occluding object. In total, our DLSU-OMRS dataset has 454 images and 786 mirrors within those images.

features by connecting the side-output of the lowest-level backbone to a sequence of three convolutional layers (Figure 3b), our proposed design (Figure 3c) connects it to a boundary extraction module with four parallel convolutional layers of varying kernel sizes and dilation rates, adapted from GDNNet [Mei22]; suppose this module’s output is denoted by f_{low} .

To extract high-level edge features, our design shares PMDNet’s approach of feeding the highest-level relational contextual contrasted local module’s output to a convolutional block attention module [Woo18], a lightweight module that infers spatial and channel attention maps; suppose its output is denoted by f_{high} .

The intermediate output maps f_{low} and f_{high} are then concatenated and passed to an edge prediction block. Our edge prediction block expands that of PMDNet, changing it from a single 3×3 convolutional layer to a 1×1 convolutional layer (with batch normalization and ReLU) connected to a 3×3 convolutional layer.

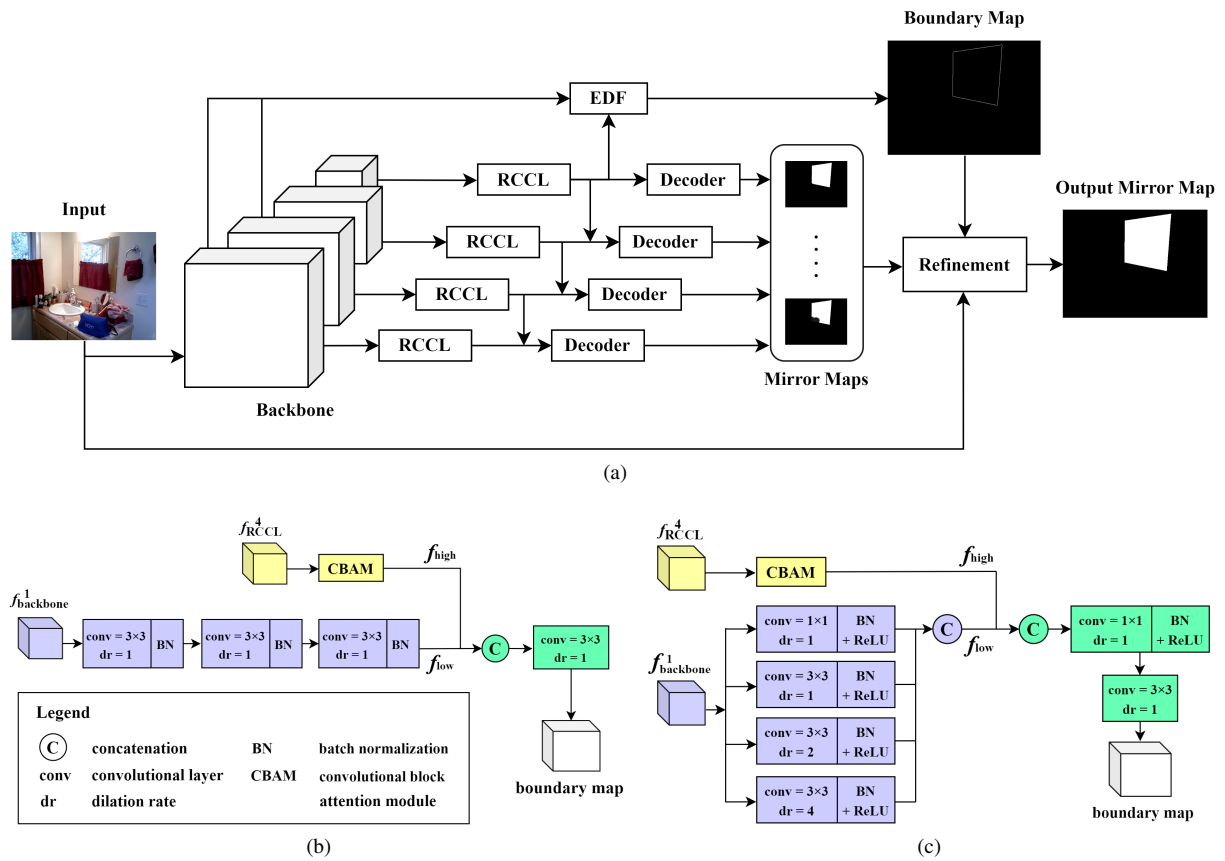


Figure 3: Model Architecture. (a) Overall architecture (the diagram is adapted from [Lin20a]). Its main components are the relational contextual contrasted local (RCCL) module, which is designed to extract contrasts and similarities inside and outside the mirror, and the edge extraction and fusion (EDF) module. (b) Original EDF module of PMDNet. (c) Our proposed modification to the EDF module. The blue blocks in (b) and (c) are for low-level edge feature extraction; $f_{backbone}^1$ is the side-output of the lowest-level backbone, and f_{low} is the low-level edge feature map. The yellow blocks are for high-level edge feature extraction; f_{RCCL}^4 is the output of the highest-level RCCL module, and f_{high} is the high-level edge feature map. The green blocks combine and process f_{low} and f_{high} for edge prediction.

These aforementioned convolutional layers both have a dilation rate of 1. Our modified architecture (Figure 3c) aims to exploit richer edge semantics without adding significant overhead to the model's complexity.

4.2 Model Training

Our models were built using PyTorch and trained on the training partition of the split PMD dataset, which consists of 5096 images. The input images were then resized to 352×352 and augmented through random horizontal flipping and jittering the brightness, contrast, saturation, and hue by a random value in the interval $[0.9, 1.1]$. They were normalized following the mean and standard deviation of the images in ImageNet [Den09]. The batch size was set to 10.

The learning rate was initialized to 1×10^{-3} and updated via a polynomial strategy with 0.9 as the power. The loss function was minimized using stochastic gradient descent with a weight decay of 5×10^{-4} and

momentum of 0.9. The models were trained for 150 epochs, with the exception of those with ResNet (200 epochs) and EfficientNet (140 epochs) backbones.

4.3 Loss Function

We combined three loss functions to supervise the training of our model. First, intersection-over-union (IoU) loss was used for the multi-scale mirror maps (i.e., excluding the final mirror map). Second, a Laplacian-based loss [Zha19] for emphasizing edges was used for the boundary map. Third, an additive loss that combines the weighted IoU and the weighted binary cross-entropy (BCE) loss proposed by [Wei20] was used for the final (output) mirror map.

Our choice of loss functions differs from the usual approach in existing mirror segmentation models [Yan19, Lin20a, Mei21, Gua22, Tan22], which mostly employ Lovász-Softmax [Ber18] for the mirror maps and BCE loss for the boundary map.

A drawback of Lovász-Softmax is its high computational cost, as noted in our initial experiments and in related studies [Alo19]. Our use of IoU loss for the multi-scale mirror maps is more efficient, albeit generally outperformed by Lovász-Softmax. To compensate for this while maintaining efficiency, we employed an additive loss that combines weighted IoU and BCE for the final mirror map. Unlike ordinary IoU and BCE loss, which focus only on individual pixels, their weighted variants draw the model to a larger receptive field [Wei20].

In addition, our use of a Laplacian-based loss function tailored for emphasizing edges is an alternative strategy to BCE, which is sensitive to imbalanced edge/non-edge distribution [Den18], a problem that is more pronounced since our edge extraction module is concerned only with the edges of the mirrors.

To formalize, let $\mathcal{L}_{\text{mirror}}(\hat{M}_i, M)$ denote the IoU loss between the i^{th} predicted mirror map \hat{M}_i and the ground truth M ; $\mathcal{L}_{\text{edge}}(\hat{E}, E)$, the Laplacian-based loss between the predicted boundary map \hat{E} and the ground truth E ; and $\mathcal{L}_{\text{output}}(\hat{M}, M)$, the additive loss between the predicted output mirror map \hat{M} and the ground truth M . Note that the ground-truth boundary maps were obtained by applying Canny edge detection [Can86] on the ground-truth mirror maps.

Our final loss function \mathcal{L} is given by Equation 1.

$$\mathcal{L} = \sum_{i=1}^4 w_{\text{mirror}} \cdot \mathcal{L}_{\text{mirror}}(\hat{M}_i, M) + w_{\text{edge}} \cdot \mathcal{L}_{\text{edge}}(\hat{E}, E) + w_{\text{output}} \cdot \mathcal{L}_{\text{output}}(\hat{M}, M) \quad (1)$$

The weighting coefficients w_{mirror} (for $i = 1$ to 4), w_{edge} , and w_{output} were set to 1, 5, and 2, respectively, following [Lin20a]. These values were empirically found to yield the best performance from a parameter space of $\{(1, 1, 1), (1, 2, 2), (1, 5, 2), (1, 5, 5), (1, 5, 7)\}$.

4.4 Model Compression

To further decrease its complexity, we subjected our best-performing model to filter pruning via geometric median (FPGM), a one-shot pruning technique that reduces redundant filters by leveraging the geometric median as a data centrality estimator to capture the mutual information shared by filters in the same layer [He19]. FPGM has also been applied in previous studies on object detection and segmentation [Hao22]. In our work, we applied FPGM on the convolutional and linear layers at a sparsity level of 10%.

After pruning, we performed retraining for 20 epochs to recover lost accuracy; to this end, we adopted a learning rate rewinding policy [Ren20], which uses the original learning rate schedule to retrain unpruned weights from their final values.

4.5 Model Evaluation

We evaluated the performance of our built models on MSD, the test partition of the split PMD dataset, and our proposed DLSU-OMRS dataset, which contain 955, 571, and 454 images, respectively.

We employed two evaluation metrics: maximum F-measure (F_β) and mean absolute error (MAE). Given the ground truth $Y(\cdot, \cdot)$, the predicted output $\hat{Y}(\cdot, \cdot)$, and an image of width w and height h , the formal definitions of these measures are given in Equations 2 and 3; β^2 was set to 0.3, as suggested by [Ach09].

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \quad (2)$$

$$\text{MAE} = \frac{1}{w \cdot h} \sum_{x=1}^w \sum_{y=1}^h |\hat{Y}(x, y) - Y(x, y)| \quad (3)$$

Moreover, the number of floating-point operations per second (FLOPS) and the number of parameters were identified to measure our models' complexity.

5 RESULTS AND ANALYSIS

5.1 Model Performance

Table 2 compares the performance of our models with two relevant state-of-the-art systems. VST [Liu21] is a transformer-based salient object detection model that can handle scenarios with similar foreground and background, as is the case for most images with mirrors. PMDNet is the base model of our work.

Our model that uses an EfficientNetV2-Medium backbone and employs our compound loss function and edge extraction and prediction module (second to last row of Table 2) registered the top performance across both metrics on the PMD dataset, as well as the lowest MAE on MSD. It performed competitively with PMDNet, achieving a slight edge on MSD and PMD. While it was slightly outperformed on DLSU-OMRS, our model has the advantage of having $4.79 \times$ fewer FLOPS and $2.77 \times$ fewer parameters.

The pruned version of this model (last row of Table 2) also performed competitively with PMDNet and registered the highest F_β on both MSD and DLSU-OMRS, slightly outperforming the said baseline by 0.0148 and 0.0033 points, respectively. It also achieved the second-lowest MAE on both of these datasets. Among our models, this pruned version has the least computational complexity, clocking in $78.20 \times$ fewer FLOPS and $238.16 \times$ fewer parameters compared to PMDNet.

On another note, although our model with an EfficientNet-Lite backbone was not able to outperform PMDNet, its F_β scores across all three benchmark datasets were consistently within 0.06 points of the

Model	Computational Complexity		MSD		PMD		DLSU-OMRS	
	GFLOPS ↓	# of Params ↓	F_β ↑	MAE ↓	F_β ↑	MAE ↓	F_β ↑	MAE ↓
VST [Liu21]	46.36	44.48M	0.4290	0.2739	0.1317	0.261	0.5730	0.2274
PMDNet [Lin20a]	118.86	147.66M	0.8350	0.0816	<u>0.8011</u>	<u>0.0324</u>	<u>0.8423</u>	0.0878
Ours (Compound Loss)								
ResNet-50	105.47	129.04M	0.7548	0.1119	0.7650	0.0403	0.7874	0.1011
Ours (Compound Loss + Edge Extraction)								
ResNet-50	116.46	130.12M	0.7695	0.1098	0.7524	0.0409	0.8042	0.1025
Xception-65	75.28	129.12M	0.7800	0.0973	0.7566	0.0401	0.7643	0.1164
VoVNet-39	98.25	61.90M	0.7014	0.1196	0.7578	0.0412	0.7868	0.1088
MobileNetV3	<u>6.61</u>	20.76M	0.7515	0.1153	0.7508	0.0427	0.8256	0.1006
EfficientNet-Lite	6.99	15.54M	0.7909	0.1027	0.7769	0.0387	0.8178	0.1048
EfficientNet-Edge-Large (Pruned)	17.02	<u>10.42M</u>	0.7682	0.1082	0.7831	0.0349	0.8035	0.1044
EfficientNetV2-Medium	24.79	53.35M	<u>0.8483</u>	0.0800	0.8117	0.0313	0.8388	0.1032
Ours (Compound Loss + Edge Extraction + FPGM Pruning)								
EfficientNetV2-Medium	1.52	0.62M	0.8498	<u>0.0813</u>	0.7902	0.0364	0.8456	<u>0.0955</u>

Table 2: Performance of the Models. The row labels for our models denote the backbone. Higher F_β and lower MAE correspond to better performance. The best scores are given in bold; the second-best scores are underlined.

highest scores. Moreover, it has $17.00\times$ fewer FLOPS and $9.50\times$ fewer parameters compared to PMDNet. These results suggest the applicability of the EfficientNet family of networks as a promising and less computationally expensive alternative to the ResNeXt backbone used in existing mirror segmentation models.

Figure 4 provides a qualitative comparison of how the different models handle some challenging cases.

5.2 Performance of the Pruned Model

To quantify the extent to which pruning can be applied without overly compromising the model's performance, we applied FPGM pruning to the best-performing unpruned model at different sparsity levels and retrained the pruned model for 20 epochs following a learning rate rewinding policy.

As seen in Tables 3 and 4, raising the sparsity from 10% to 20% decreased the F_β score by around 0.02 to 0.04 points; further increasing it to 40% already resulted in a significant drop of around 0.16 to 0.22 points. A visual example is provided in Figure 5.



Figure 5: Visual Example of Performance Under Different Sparsity Levels. In this image taken from our proposed dataset, the performance of the pruned model noticeably degrades at 30% sparsity and above, as it already fails to properly distinguish the hung face mask from the mirror.

Sparsity Level	MSD	PMD	DLSU-OMRS
40%	0.6267	0.6006	0.6876
30%	0.7695	0.7566	0.7963
20%	0.8073	0.7795	0.8211
10%	0.8498	0.7902	0.8456
Unpruned	0.8483	0.8117	0.8388

Table 3: F_β Under Different Sparsity Levels

Sparsity Level	MSD	PMD	DLSU-OMRS
40%	0.4633	0.4790	0.1485
30%	0.0970	0.0410	0.1039
20%	0.0905	0.0352	0.0940
10%	0.0813	0.0364	0.0955
Unpruned	0.0800	0.0313	0.1032

Table 4: MAE Under Different Sparsity Levels

	MSD	PMD	DLSU-OMRS
Unpruned	0.8483	0.8117	0.8388
Not Retrained	0.8505	0.7858	0.8432
Retrained	0.8498	0.7902	0.8456

Table 5: F_β of Pruned Model (Sparsity = 10%) Before and After Retraining

	MSD	PMD	DLSU-OMRS
Unpruned	0.0800	0.0313	0.1032
Not Retrained	0.4185	0.4585	0.4407
Retrained	0.0813	0.0364	0.0955

Table 6: MAE of Pruned Model (Sparsity = 10%) Before and After Retraining

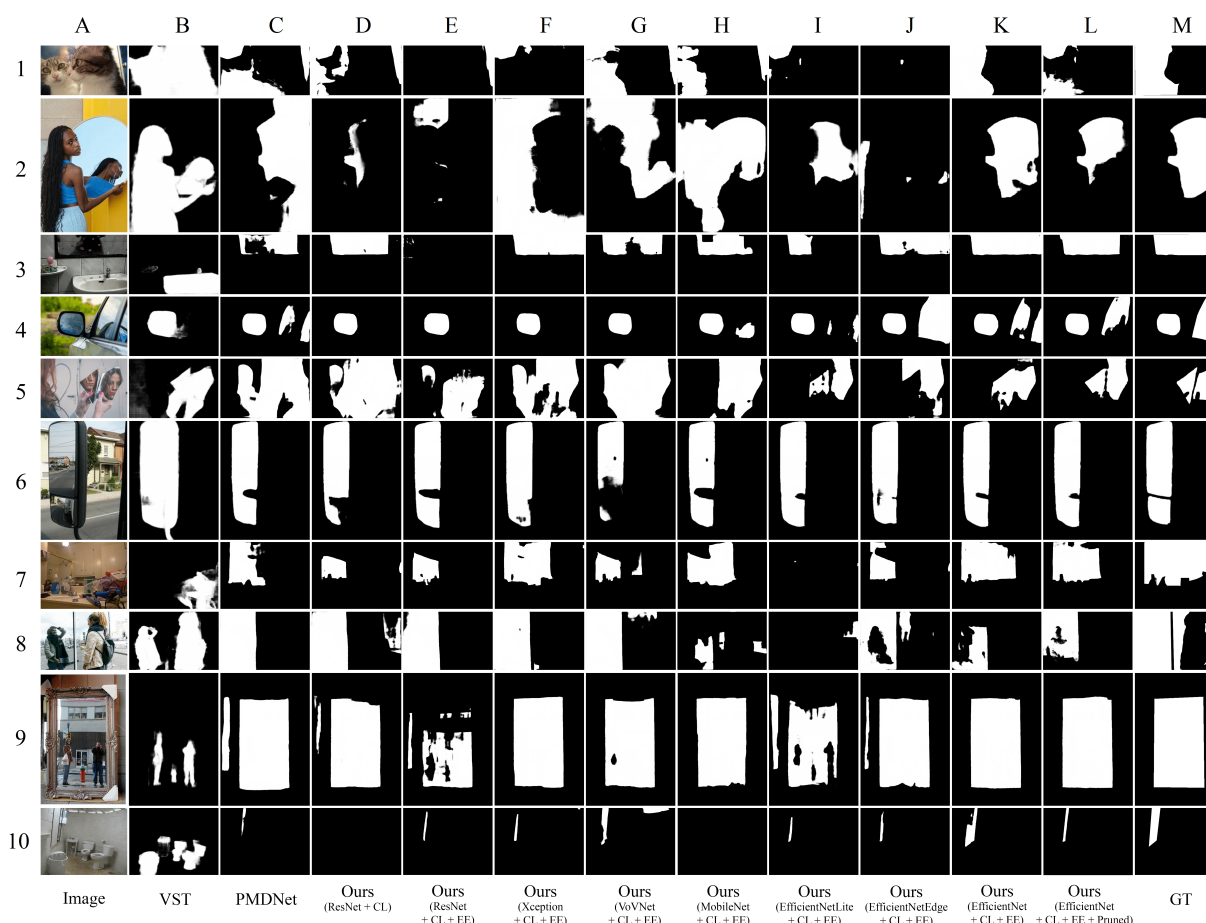


Figure 4: Qualitative Comparison on Challenging Cases. *CL* and *EE* indicate that the model uses our proposed *compound loss* and *edge extraction and prediction module*, respectively. *GT* pertains to the *ground truth*. Salient object detection models (column B) may not necessarily tag mirrors as salient. Our best-performing model (column K) can handle some cases that may be challenging even for a state-of-the-art model (column C). These include images where (i) the object occludes the mirror and, alongside its reflection, occupies a large portion of the image (rows 1 and 2), (ii) the reflection has a similar color to the mirror's frame (row 3), and (iii) multiple mirrors and reflective surfaces are present (row 4). Our pruned version (column L) was able to segment irregularly shaped mirror shards (row 5), although, in general, it seems to have some difficulty handling cases where mirrors are separated by only a thin divider (row 6) and where the object and reflection occupy the majority of the image (rows 1 and 2). Although our best-performing model and its pruned version captured the largest fraction of the ground-truth mask in row 7, it remains challenging to handle cases where the contextual features inside and outside the mirror appear continuous (row 8).

Tables 5 and 6 report the performance after pruning the model at 10% sparsity but prior to retraining. Although the F_β score was comparable, there was a significant increase in MAE prior to retraining. This increased MAE can be attributed to the resulting output maps emphasizing the mirrors but failing to completely mask out the surroundings, as seen in Figure 6.



Figure 6: Visual Example of Performance of Pruned Model Before and After Retraining

5.3 Model Component Analysis

To demonstrate the contribution of our proposed edge extraction and prediction module, we conducted ablation experiments on our unpruned model (Tables 7 and 8). On MSD and PMD, incorporating our module outperformed not including any edge semantics-related module and utilizing PMDNet's original edge detection and fusion module. On DLSU-OMRS, using PMDNet's original module resulted in the highest performance, albeit only by 0.0001 F_β and 0.0114 MAE points. Visual examples are given in Figure 7.

To investigate the effects of our choice of loss functions, we also measured the performance of our best-

performing model if simple BCE and IoU loss functions were used to supervise the training of the boundary and final mirror maps, respectively. As seen in Tables 9 and 10, our proposed loss function resulted in the best F_β and MAE scores on MSD and the highest F_β on PMD. A visual example is also provided in Figure 8.

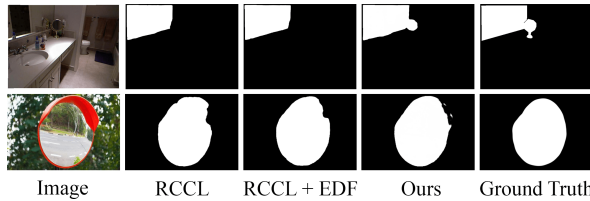


Figure 7: Visual Example of Performance of Ablated Models. The use of our edge extraction and prediction module helps in capturing boundaries of small objects that may otherwise be missed (first row). However, in certain cases, it may also result in the inclusion of noise in the predicted mask (second row).

	MSD	PMD	DLSU-OMRS
RCCL	0.8052	0.7957	0.8300
RCCL + EDF	0.8224	0.8001	0.8389
Ours	0.8483	0.8117	0.8388

Table 7: F_β After Ablation. *RCCL* and *EDF* refer to PMDNet's relational contextual contrasted local module and edge detection and fusion module. Our model modifies the EDF module (Section 4.1).

	MSD	PMD	DLSU-OMRS
RCCL	0.0957	0.0332	0.0956
RCCL + EDF	0.0949	0.0335	0.0918
Ours	0.0800	0.0313	0.1032

Table 8: MAE After Ablation

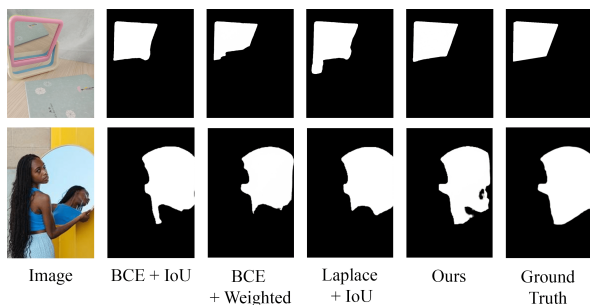


Figure 8: Visual Example of Performance Under Different Loss Functions. Using our compound loss function resulted in the most accurate mirror map in the image in the first row. Although its use in the image in the second row increased sensitivity to boundaries proximate to the reflection's chest area, the overall contour of the ground-truth mask was better captured.

Loss	MSD	PMD	DLSU-OMRS
BCE + IoU	0.8352	0.8038	0.8314
BCE + Weighted	0.8163	0.8073	0.8470
Laplace + IoU	0.8148	0.7989	0.8553
Ours	0.8483	0.8117	0.8388

Table 9: F_β Under Different Loss Functions. *Ours* refers to our use of a Laplacian-based loss function for the boundary map and an additive loss function combining weighted IoU and BCE loss for the final mirror map (Section 4.3).

Loss	MSD	PMD	DLSU-OMRS
BCE + IoU	0.0949	0.0320	0.0969
BCE + Weighted	0.0967	0.0319	0.0995
Laplace + IoU	0.0950	0.0302	0.0881
Ours	0.0800	0.0313	0.1032

Table 10: MAE Under Different Loss Functions

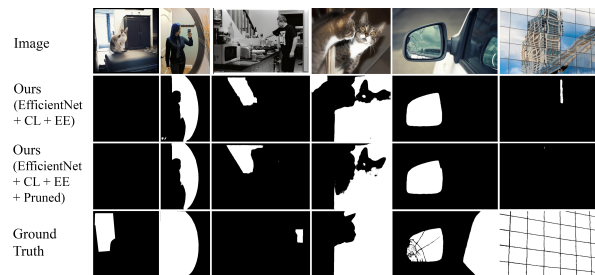


Figure 9: Failure Cases. *CL* and *EE* indicate that the model uses our proposed *compound loss* and *edge extraction and prediction module*, respectively. Some failure cases, such as the fourth image, may be confusing even for human observers. Moreover, fine details such as cracks (second to last image) are generally not preserved, although the mirror's overall contour is correctly captured.

5.4 Failure Cases

Figure 8 shows the limitations of our model. Since our model exploits contextual discontinuities and similarities, it has some difficulty handling cases where the contextual features inside and outside the mirror appear continuous (first image) or where the available contextual features are inadequate due to the mirror occupying the entire image (last image).

Sharp discontinuities within the mirror (second image) may also result in the reflection being treated as part of the predicted mask's boundary. Some transparent glass objects may be falsely flagged as mirrors, whereas small mirrors in the background (third image) and heavily tinted reflective surfaces (fifth image) may be challenging to recognize.

6 CONCLUSION

In this study, we propose DLSU-OMRS, a dataset of 454 images of outdoor mirrors and reflective surfaces, which are not well represented in existing mirror datasets. We also modified the architecture of PMDNet and extensively tested different feature extraction backbones and edge-related modules to guide the segmentation. Our best-performing model uses EfficientNetV2-Medium as its backbone and employs an edge detection module consisting of parallel convolutional layers and a lightweight convolutional block attention module to capture both low-level and high-level edge semantics.

Our model performs competitively with the state-of-the-art PMDNet, registering F_β scores of 0.8483, 0.8117, and 0.8388 on MSD, PMD, and our proposed dataset, respectively. Compressing this model by pruning via geometric median resulted in F_β scores of 0.8498, 0.7902, and 0.8456, respectively, maintaining competitive performance but with $78.20\times$ fewer FLOPS and $238.16\times$ fewer parameters.

Future directions include addressing the discussed limitations of our work and extending our approach to further realize the applicability of mirror detection and segmentation models to resource-constrained devices, such as those for autonomous navigation (e.g., drones).

7 ACKNOWLEDGMENT

We thank Mr. Gregory G. Cu and Mr. Fritz Kevin S. Flores for providing us with access to the University's computing resources. We also thank Dr. Macario O. Cordel, II and Dr. Ann Franchesca B. Laguna for their feedback on the initial manuscript.

8 REFERENCES

- [Ach09] Achanta, R., Hemami, S., Estrada, F., and Susstrunk, S. Frequency-tuned salient region detection. 2009 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1597-1604, 2009.
- [Alo19] Alonso, I., Yuval, M., Eyal, G., Treibitz, T., and Murillo, A.C. CoralSeg: Learning coral segmentation from sparse annotations. *Journal of Field Robotics*, 36, 8, pp. 1456-1477, 2019.
- [And18] Anderson, P. et al. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3674-3683, 2018.
- [Ber18] Berman, M., Triki, A.R., and Blaschko, M. The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [Can86] Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, 6, pp. 679-698, 1986.
- [Cha17] Chang, A. et al. Matterport3D: Learning from RGB-D Data in indoor environments. 2017 International Conference on 3D Vision (3DV), pp. 667-676, 2017.
- [Cha22] Chahal, E.S., Patel, A., Gupta, A., Purwar, A., and Dhanalekshmi, G. Unet based Xception model for prostate cancer segmentation from MRI images. *Multimedia Tools and Applications*, 81, 26, pp. 37333-37349, 2022.
- [Cho17] Chollet, F. Xception: Deep learning with depthwise separable convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1800-1807, 2017.
- [Den09] Deng, J. et al. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248-255, 2009.
- [Den18] Deng, R., Shen, C., Liu, S., Wang, H., and Liu, X. Learning to predict crisp boundaries. *Computer Vision - ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI*, pp. 570-586, 2018.
- [Gua22] Guan, H., Lin, J., and Lau, R.W.H. Learning semantic associations for mirror detection. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5931-5940, 2022.
- [Hao22] Hao, Z., Wang, Z., Bai, D., and Tong, X. Surface defect segmentation algorithm of steel plate based on geometric median filter pruning. *Frontiers in Bioengineering and Biotechnology*, 10, pp. 945248, 2022.
- [He16] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.
- [He17] He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980-2988, 2017.
- [He19] He, Y., Liu, P., Wang, Z., Hu, Z., and Yang, Y. Filter pruning via geometric median for deep convolutional neural networks acceleration. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4335-4344, 2019.
- [How19] Howard, A. et al. Searching for MobileNetV3. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1314-1324, 2019.

- [Hua17] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K.Q. Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261-2269, 2017.
- [Kra11] Krähenbühl, P., and Koltun, V. Efficient inference in fully connected CRFs with Gaussian edge potentials. *Advances in Neural Information Processing Systems*, 24, 2011.
- [Lee19] Lee, Y., Hwang, J.W., Lee, S., Bae, Y., and Park, J. An energy and GPU-computation efficient backbone network for real-time object detection. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 752-760, 2019.
- [Lin20a] Lin, J., Wang, G., and Lau, R.H. Progressive mirror detection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3694-3702, 2020.
- [Lin20b] Lin, Z., Sun, J., Davis, A., and Snavely, N. Visual chirality. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12292-12300, 2020.
- [Lin22] Lin, J. et al. Efficient heterogeneous video segmentation at the edge. *Sixth Workshop on Computer Vision for AR/VR (CV4ARVR)*, 2022.
- [Liu21] Liu, N., Zhang, N., and Wan, K., Shao, L., and Han, J. Visual saliency transformer. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4722-4732, 2021.
- [Mei21] Mei, H. et al. Depth-aware mirror segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3044-3053, 2021.
- [Mei22] Mei, H. et al. Large-field contextual feature learning for glass detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 01, pp. 1-17, 2022.
- [Par21] Park, D., and Park, Y.H. Identifying reflected images from object detector in indoor environment utilizing depth information. *IEEE Robotics and Automation Letters*, 6, 2, pp. 635-642, 2021.
- [Ren20] Renda, A., Frankle, J., and Carbin, M. Comparing rewinding and fine-tuning in neural network pruning. *International Conference on Learning Representations*, 2020.
- [Sil12] Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. Indoor segmentation and support inference from RGBD images. *European Conference on Computer Vision (ECCV)*, 2012.
- [Tan19] Tan, M., and Le, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th International Conference on Machine Learning*, 97, pp. 6105-6114, 2019.
- [Tan22] Tan, X. et al. Mirror detection with the visual chirality cue. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-13, 2022.
- [Tin16] Tin, S.K., Ye, J., Nezamabadi, M., and Chen, C. 3D reconstruction of mirror-type objects using efficient ray coding. 2016 IEEE International Conference on Computational Photography (ICCP), pp. 1-11, 2016.
- [Wei20] Wei, J., Wang, S., and Huang, Q. F³Net: Fusion, feedback and focus for salient object detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34, pp. 12321-12328, 2020.
- [Whe18] Whelan, T. et al. Reconstructing scenes with mirror and glass surfaces. *ACM Trans. Graph.*, 37, 4, 2018.
- [Woo18] Woo, S., Park, J., Lee, J., Lee, J., and Kweon, I.S. CBAM: Convolutional block attention module. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3-19, 2018.
- [Xie17] Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5987-5995, 2017.
- [Yan19] Yang, X. et al. Where is my mirror? *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8809-8818, 2019.
- [Zen17] Zende, O., Honauer, K., Murschitz, M., Humenberger, M., and Domínguez, G. Analyzing computer vision data - the good, the bad and the ugly. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6670-6680, 2017.
- [Zha17] Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid scene parsing network. 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6230-6239, 2017.
- [Zha18] Zhang, Y., Ye, M., Manocha, D., and Yang, R. 3D reconstruction in the presence of glass and mirrors by acoustic and visual fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40, 8, pp. 1785-1798, 2018.
- [Zha19] Zhao, T., and Wu, X. Pyramid feature attention network for saliency detection. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3080-3089, 2019.
- [Zho17] Zhou, B. et al. Scene parsing through ADE20K dataset. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Monte Carlo Based Real-Time Shape Analysis in Volumes

Krishna Gurijala

Stony Brook University
New York, USA

kgurijala@cs.stonybrook.edu

Lei Wang

Stony Brook University
New York, USA

leiwang1@cs.stonybrook.edu

Arie Kaufman

Stony Brook University
New York, USA

ari@cs.stonybrook.edu

ABSTRACT

We introduce a Monte Carlo based real-time diffusion process for shape-based analysis in volumetric data. The diffusion process is carried out by using tiny massless particles termed shapetons, which are used to capture the shape information. Initially, these shapetons are randomly distributed inside the voxels of the volume data. The shapetons are then diffused in a Monte Carlo fashion to obtain the shape information. The direction of propagation for the shapetons is monitored by the Volume Gradient Operator (VGO). This operator is known for successfully capturing the shape information and thus the shape information is well captured by the shapeton diffusion method. All the shapetons are diffused simultaneously and all the results can be monitored in real-time. We demonstrate several important applications of our approach including colon cancer detection and design of shape-based transfer functions. We also present supporting results for the applications and show that this method works well for volumes. We show that our approach can robustly extract shape-based features and thus forms the basis for improved classification and exploration of features based on shape.

Keywords

Shapeton diffusion, shape analysis, Monte Carlo, colon cancer detection, transfer-function.

1 INTRODUCTION

Much research has been undertaken to incorporate information for volume data analysis from various parameters such as voxel intensity, gradient, curvature, and size. However, incorporating shape information for volume analysis still remains a challenge. This is not the scenario in the case of manifolds, where diffusion based techniques have become popular for manifold shape analysis. A successful attempt has been made by Gurijala et al. [GWK12] in using the diffusion based method for shape-based volume analysis, wherein a modified form of heat diffusion, called cumulative heat diffusion (CHD), was introduced. Despite good results, this method cannot be adopted for real-time analysis due to the high computational cost. Precisely, the computational complexity of the heat diffusion process for discrete surface meshes is of the order $t \times n^2$, where n is the number of voxels and t is the number of time steps. In addition, the heat diffusion is carried out only between voxels and 1-ring neighboring voxels per time step and hence the number of time steps required to capture the shape information increases with the increasing number

of voxels. In other words, the rate of heat flow is influenced by the resolution of the data. As a result, the diffusion based methods suffer from the problem of long running times. In order to address these challenges, in this paper, we introduce a Monte Carlo based shape analysis method for volumes which not only obtains efficient results but also provides a means of real-time shape analysis, by re-defining the diffusion process using a new set of particles, which we call shapetons. In addition, a new definition of time step is introduced.

This paper makes the following contributions. We introduce a new diffusion based shape analysis method using new particles, called shapetons. The shapetons are tiny massless particles which are diffused across the voxels of a volume, in random directions, for a pre-defined distance per time step, to determine the local shape information. This is the first time the diffusion particles (in our case, the shapetons) are diffused across the voxels separated by some distance, rather than just between the adjacent voxels. Our method is independent of the size of the volume; it only depends on the number of shapetons. This independence on the resolution (size) of the data is another important contribution of the paper. In addition, using probabilistic methods for shape analysis is in itself a contribution. All the shapetons can be diffused simultaneously and independent of each other. As a result, our method can run in parallel for all the shapetons. We use the GPU for implementation and the convergence of the shapetons to a stable value can be monitored in real time, thereby fa-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

cilitating real-time shape analysis. To the best of our knowledge, this is the first time volume analysis based on shape with real-time monitoring of the result is being carried out, thereby achieving orders of magnitude improvement in the computational cost.

The remainder of the paper is organized as follows. Section 2 provides background information and reviews the related literature. Section 3 describes the algorithm with a detailed description of the shapeton diffusion process. We discuss how our shapeton diffusion method can robustly extract the features based on their shape information in the volume data. The influence of different parameters on the result is analyzed in Section 4. Section 5 discusses applications of our method in colon cancer detection and transfer function design and Section 6 presents the results of our method. Finally, we draw some concluding remarks along with the future work in Section 7.

2 RELATED WORK

Shape has been previously used for volume classification. Sato et al. [YSABNSK00] have proposed a volume classification based on shape where they detect pre-defined shapes such as edge lines and blobs by measuring the multi-scale responses to 3D filters. Skeleton based approaches were extensively used to study shapes and for shape based volume visualization. Hilaga et al. [HSKK01] have used skeletons for shape matching and volume visualization. Pizer et al. [PGJA03] have proposed a framework of stable medial representation for segmentation of objects, registration and statistical 3D shape analysis. Several other attempts using skeletons for shape-based volume classification were conducted by Correa et al. [CS05] and Reniers et al. [RJT08]. Motivated by these ideas, Praßni et al. [PRMH10] have presented a shape-based transfer function using the curve-skeleton of the volumetric structure. However, in all these works, the shape has been pre-defined such as blobs, surfaces and tubes. In contrast, we enforce no shape restrictions. All similar shapes, irrespective of orientation and scaling are recognised and at the same time distinguished from other shapes.

In volumes, the diffusion methods have been majorly used in the form of photon diffusion in the volume rendering pipeline [Jen96]. Apart from this, diffusion based models have also been used to visualize fire [SF95], air pollution [Wan13], and rendering depth of field effects [KB07]. None of these diffusion based approaches have been used for shape analysis. Diffusion based methods have been used extensively for shape analysis in manifolds [ASC11, BK10, OMMG10, SOG09, VBCG10]. However, these methods cannot be directly extended to volumes for shape analysis due to the huge computational cost. The only attempt to perform volume

analysis based on shape was made very recently by Gurijala et al. [GWK12] who introduced a cumulative heat diffusion approach. Despite the novelty, the method still has a large computational cost and the shape analysis cannot be monitored in real-time. Using our shapeton diffusion approach, we are able to not only perform shape-based volume analysis but also monitor the analysis in real-time.

Monte Carlo methods are not new to volume graphics and visualization [AK90, BSS94, PM93]. They have been largely used for photorealistic rendering (photon mapping) [DEJ⁺99, Jen96, JC98] and ray tracing (rendering volumetric caustics and shadows) [JLD99, LW96, PKK00]. Unfortunately, most of these methods have high computational cost. To solve this, several variations of Monte Carlo photon diffusion approximation methods have been proposed for various rendering applications [DJ05, JMLH01, Sta95]. All these Monte Carlo based photon diffusion methods are a combination of a diffusion model and Monte Carlo methods ala our technique. A GPU-based Monte Carlo volume rendering approach including scattering, ambient occlusion has been proposed by Salama [Sal07]. However, none of these methods focus on shape analysis in volumes. Ours is the first time a Monte Carlo based method using GPU has been developed for shape based volume analysis, thereby facilitating a real-time monitoring of the shape information.

3 ALGORITHM

The shapeton diffusion process efficiently captures the shape information in volumes and in addition facilitates a real time monitoring of this information. The diffusion particles, the shapetons, are able to capture the majority of the shape information and hence the name shapetons. Initially, these shapetons are randomly distributed inside the data. The primary idea of our approach is that each shapeton is diffused based on the local shape information in a probabilistic manner. The probability that a shapeton moves in a particular direction is based on how much the region in that direction contributes to the shape information. In continuous space, generally the shape information around each shapeton can be represented in the form of an uneven distribution. This is because the local shape information around the shapeton is not uniform (varies depending on the data). The area of this shape distribution would give a measure of the shape information obtained around the shapeton. A random number is used to select a fraction of the area. This fractional area indicates the shape information obtained in that direction and in turn the probability for the shapeton to move in that direction. In other words, the probability of shapeton diffusion is based on the ratio of the area of a sub-region to the area of the total shape distribution. The difference between our method and previous diffusion based

methods such as the cumulative heat diffusion (CHD) is that, ours is a particle-based (shapetons) diffusion process while the latter is not. During the diffusion process the shapetons are moved inside the volume, across the voxels, for a pre-defined distance in each time step. As a result, the shapetons have the freedom to move anywhere inside the volume and not just between the 1-ring neighboring voxels. Therefore, the rate of shapeton diffusion is not affected by the resolution of the data and is independent of the size of the data. In the remainder of the paper, please note that all the pre-defined distance values are chosen by considering a $[0, 1]$ normalized space of the volume data. Hence, the distance value will always lie in the interval $[0, 1]$. Initially, all

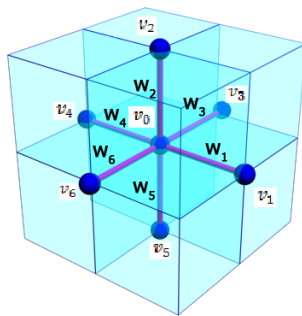


Figure 1: $v_1, v_2, v_3, v_4, v_5, v_6$ are the 1-ring neighboring voxels of the source voxel v_0 and w_1, w_2, w_3, w_4, w_5 and w_6 are the corresponding edge weights respectively.

the shapetons are randomly distributed inside the voxels. The initial distribution of the shapetons does not influence the final result. Only the steady state result is considered for shape analysis. The steady state result will smooth out the differences and is not affected by the initialization of the shapetons. We will discuss in detail about the steady state later. The shapetons are diffused inside the volume based on the local shape information. In order to describe the direction along which the shapetons travel in each time step, two angles are used, namely the longitudinal angle and the latitudinal angle. We now describe the elaborate process of shapeton diffusion in detail. In general, any voxel is surrounded by six adjacent voxels in a volume. Thus, for any shapeton s inside a voxel (say v_0), there are six adjacent voxels (say v_1, v_2, v_3, v_4, v_5 and v_6).

The edge weights w_1, w_2, w_3, w_4, w_5 and w_6 between the voxel v_0 and its adjacent voxels, as shown in the Figure 1, are determined using the VGO [GWK12], defined by Equation 1. This VGO captures the local shape information of the volume. There is a parameter p in the VGO definition that influences the final result. We discuss the effect of the parameter p in Section 4.5. For $i \in \{1, 2, 3, 4, 5, 6\}$:

$$w_i = VGO(v_0, v_i) = \Delta(v_0, v_i) + F_v(v_0, v_i) \quad (1)$$

where Δ is the Laplace-Beltrami Operator (LBO) and F_v is a data-driven operator:

$$F_v(v_0, v_i) = 1 - p \cdot h_g(v_0, v_i) \quad (2)$$

where h_g is the half gradient and p is a user defined value. The half gradient h_g of the voxel v_0 is given by:

$$h_g(v_0, v_i) = \left| \frac{I(v_i) - I(v_0)}{res} \right| \quad (3)$$

where I gives the intensity of the corresponding voxel, res is the size of the voxel which accounts for the distance between the two voxels under consideration.

We use these six edge weights to create a shape distribution diagram around the shapeton, as shown in Figure 2. This shape distribution accounts for the shape information around the voxel v_0 (the shapeton is inside this voxel) and is used to determine the direction of shapeton diffusion in a probabilistic manner. The six weights form eight regions where each region represents an octant of a sphere. We call this octant of the sphere octavusphere (derived from Latin). Therefore, the six weights form eight octavuspherical regions where sets of three weights form a single octavuspherical region, as shown in Figure 2. In spherical coordinates we normally need two angles (say θ and ϕ) to describe the direction of shapeton propagation. The angle θ is measured with respect to the x -axis on the $x-y$ plane and the angle ϕ is measured with respect to the y -axis on the $y-z$ plane. In geographical terms, we refer to the angle θ as the longitudinal angle and the angle ϕ as the latitudinal angle.

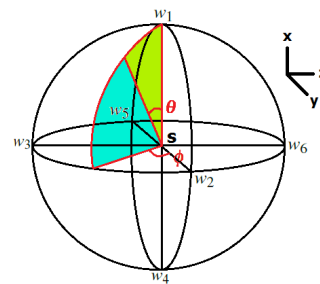


Figure 2: The shape distribution around the shapeton s shown using the edge weights. The probabilistically estimated angles ϕ and θ define the direction of the shapeton propagation.

Since we have to determine two angles probabilistically, namely θ (longitude) and ϕ (latitude), two random numbers are drawn, one for each of them. We do it in a step-by-step manner. First, the value of the angle ϕ is determined by employing the first random number. Fixing this value of ϕ , the value of the angle θ is then estimated by employing the second random number. In a given octavuspherical region both ϕ and θ vary between 0 and $\frac{\pi}{2}$. The probability of the shapeton

diffusion should take into account the shape information around it, which is indicated by the volume of the octavuspherical region enclosed by the edge weights. In other words, the probability of the shapeton to move in a certain octavusphere is based on the ratios of the volumes of the octavuspheres to the whole volume. By employing the first random number over the volumes of the octavuspherical regions, a particular octavusphere region is selected and the corresponding value of ϕ is estimated. This angle ϕ splits the selected octavuspherical region into two sub-regions, which are separated by a sector shown by the green and blue regions in Figure 2. By employing a second random number over the area of this sector the final value of θ is estimated. More details about the steps involved in calculating the longitude and latitude for shapeton propagation are provided in Appendix 4.

Now that we have evaluated both ϕ and θ , we have the final direction for the shapeton to move. Once the direction of propagation for the shapeton is determined, the shapeton is moved in that direction for a pre-defined distance. This accounts for one time step of the shapeton. This process is performed for all the shapetons independently and simultaneously. After each time step, all the steps described above are repeated to calculate the new direction for the shapetons to diffuse. After each time step, the number of shapetons inside each voxel is summed up to get the accumulated density of shapetons. For example, let $s_{t-1}(i)$ be the accumulated shapeton density on a vertex i before the t_{th} time step and c_t be the number of shapetons that move onto that vertex from its neighboring vertices during the t_{th} time step. Then, the new accumulated shapeton density $s_t(i)$ on that vertex is:

$$s_t(i) = s_{t-1}(i) + c_t \quad (4)$$

The value of c_t will either be positive or zero and never negative since we only consider the number of shapetons that accumulate in each of the voxels after each time step and not the number of shapetons that diffuse away from the voxels. Note that no new shapetons are added at any stage of the algorithm and the number of shapetons used for diffusion is always constant. Only the shapetons diffuse inside the volume and based on which voxel each of the shapetons are present after every time step, the corresponding accumulated shapeton density of those voxels are updated. The accumulated number of shapetons in each voxel indicates the probability of the shapetons to appear at that location. For all the voxels corresponding to objects of similar shape, the shapetons have a similar probability to visit them. Hence, the number of shapetons within each voxel would be the same for all voxels corresponding to objects of similar shape. The diffusion of shapetons in volumes is influenced by the VGO

which incorporates the local shape information. Thus, the shapetons capture the shape information along their path of diffusion and the accumulated number of the shapetons inside the voxels quantifies the shape information obtained.

4 ANALYSIS

The shapeton diffusion process is an efficient method in classifying different objects based on their shape. The shape information is obtained irrespective of the size and deformation of the objects. However, the amount of shape information obtained is influenced by a number of parameters such as the number of shapetons, the value of the pre-defined distance, and the value of p . In the remainder of the paper, for all the results, the rendering is based on the accumulated number of shapetons in the voxels for a given number of time steps and the colors are assigned such that a higher shapeton count is shown in red and the color changes from red to blue with the decrease in the shapeton count.

4.1 Steady State

Like any Monte Carlo method, the probability of the shapetons to take a particular path increases as we increase the number of shapetons and hence the rate of accumulation of shapetons at a particular feature increases. Thus, the shape information is obtained much faster in terms of the number of iterations with the increase in the number of shapetons. If the number of shapetons is reduced, it takes more iterations to capture a specific feature, which otherwise would have taken fewer iterations using more shapetons. However, there is a tradeoff. Though the number of iterations decreases, the time taken for each iteration (time step) increases with the increase in the number of shapetons.

We say that the shapeton diffusion process has reached a steady state if the rate of change of the accumulated shapeton density on all the voxels is uniform. For this, we check if the rate of change of the accumulated shapeton density on all the voxels after every time step ($\Delta t = 1$) is below a threshold value as follows:

$$\Delta s(t) = \sum_{i \in V} (s_t(i) - s_{t-1}(i))^2 \leq \epsilon \quad (5)$$

where $\Delta s(t)$ denotes the rate of change in the accumulated shapeton density for all the voxels after t time steps, V denotes the number of voxels in the volume, $s_t(i)$ and $s_{t-1}(i)$ are the accumulated shapeton densities on voxel i after t and $t - 1$ time steps respectively and ϵ is the threshold value. In all our datasets, we choose the threshold value to be 0.05. This threshold value chosen is not an accurate estimation and is chosen experimentally by observing the shapeton diffusion process on several datasets. As future work, we plan on finding a way to provide a more accurate estimate of the

threshold value that will be dependent on the dataset. We check if the condition in Equation 5 is satisfied continuously in at least 90% of the last 50 time steps. The 10% leverage is given to account for some unexpected changes caused due to the probabilistic movement of the shapetons. The number of time steps after which all these requirements are satisfied is chosen to be the point where a steady state is reached.

4.2 Distance Value

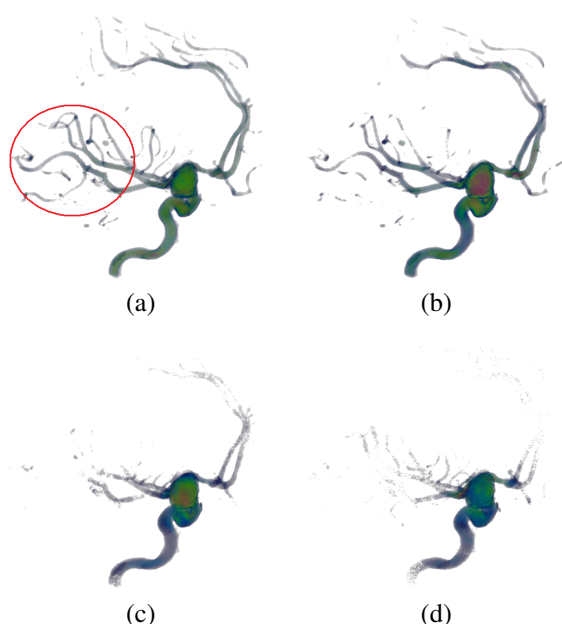


Figure 3: Effect of the different distance values on the aneurysm volume data using 400 time steps. Smaller features such as the narrow blood vessels (shown in the red circle) are captured using small distance values of 0.001 in (a) and 0.005 in (b) which are absent when larger distance values of 0.01 in (c) and 0.05 in (d) are used.

When a shapeton travels a pre-defined distance (defined by the user), it is said to complete one time step or iteration of the diffusion process. This distance value also affects the diffusion process of the shapetons and the shape information captured. When we use a large distance value, the shapetons travel a larger distance in one time step. Thus, if we increase the distance value the diffusion process converges faster in terms of the number of time steps when compared to a lower distance value. The smaller distance values cause the shapetons to move slowly, thereby resulting in more time steps needed to capture the global shape. However, the catch here is that we cannot obtain the local features using a large distance value because most of the shapetons will travel over the smaller features missing them completely. Smaller distance values are useful in obtaining and analyzing local features. Therefore, the distance value is an indication of the shape information obtained at different scales of the data. Intricate local

shape details are obtained by using a smaller distance value, while global shape information is obtained using a higher distance value (with a smaller number of time steps). It is not that the smaller distance value is unable to capture the global shape information, it is just that it takes more time steps to obtain the global shape information using a smaller distance value. On the contrary, a higher distance value is unable to obtain the local features despite using more time steps.

Figure 3 shows the results of the shapeton diffusion on the aneurysm dataset using different distance values for the same number of 400 time steps. Figure 3(a) shows the result for a distance value of 0.001; Figure 3(b) shows it for a distance value of 0.005; Figure 3(c) shows it for a distance value of 0.01, and Figure 3(d) shows the result for a distance value of 0.05. For small distance values even the smaller features such as the narrow blood vessels (shown in the red circle) are captured. As the distance value is increased, only the relatively larger features such as the aneurysm blob are captured by the shapetons. The smaller features such as the narrow vessels are missing in Figures 3 (c) and (d), where a higher distance value is used.

4.3 Shape Classification

We now show that our shapeton diffusion method is indeed successful in classifying different shapes. The shapetons are diffused based on the VGO in volumes, which captures the shape information. Hence, the probability of a shapeton to go in a particular path is influenced by the shape information. The accumulated number of shapetons per voxel in a shape such as a cube would be different from a shape such as a sphere since both of them have different shape and thus bear different probabilities for the shapetons to capture them.

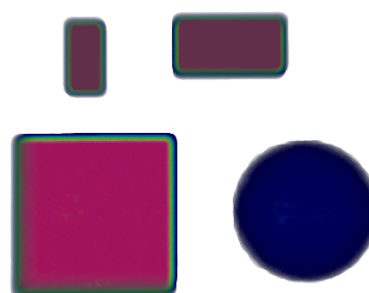


Figure 4: Shape classification capability of the shapeton diffusion approach shown using a synthetic data consisting of a cube, two cuboids of different size and orientation and a sphere.

We use a synthetic data consisting of a cube, two cuboids of different size and orientation and a sphere to confirm this. Figure 4 shows the result of using the shapeton diffusion method on the synthetic data. We consider a large number of 2500 time steps to make

sure that a stable state is reached. In each time step, the shapeton was moved by a distance of 0.01. We can clearly see from Figure 4 that all the shapes have been identified and distinguished successfully (shown by the different colors). The colors are assigned based on the number of shapetons accumulated. The shapeton propagation is based on the shape information (VGO) and hence the number of shapetons accumulated per voxel is the same in similar shaped objects. This fact can be observed in Figure 4 where both cuboids have the same color. In addition, the color of the cube is almost similar to that of cuboids indicating that they have almost similar shape. The cube and the sphere have also been classified as different shapes, thus asserting that the shapeton diffusion method serves as a powerful tool in finding objects with similar shape and distinguishing them from objects with other shape. An important observation that can be made from Figure 4 is that both cuboids have been identified as similar shape irrespective of their size and orientation. This further confirms that our method can recognize different shapes independent of their size and orientation.

4.4 Invariance to Deformations

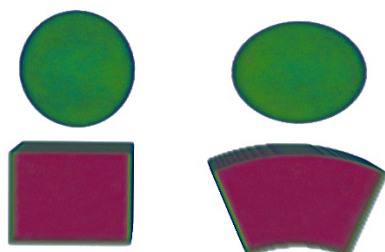


Figure 5: Objects of similar shape identified successfully irrespective of their deformations.

The shapeton diffusion method displays some lucrative properties such as invariance to deformations. We generated a synthetic data to establish this property. The synthetic data consists of a cuboid, a deformed cuboid, a sphere and a deformed sphere. Figure 5 shows the result of the shapeton diffusion on this synthetic data. Again the diffusion process is carried out for a large number of time steps to ensure a stable state is reached and all the objects in the volume data are obtained.

You can observe that although the cuboid has been deformed, the number of shapetons accumulated per voxel in both the cuboid and its deformed version are the same and hence both have similar color. Likewise, the sphere and its deformed version have similar color. The sphere and the cuboid have also been distinguished from each other. We used 1000 shapetons for 2500 time steps to obtain the results. The results in Figure 5 show that the shapeton diffusion method is successful in identifying objects of similar shape though they have been

deformed, thus proving that it is invariant to deformation.

4.5 Effect of p

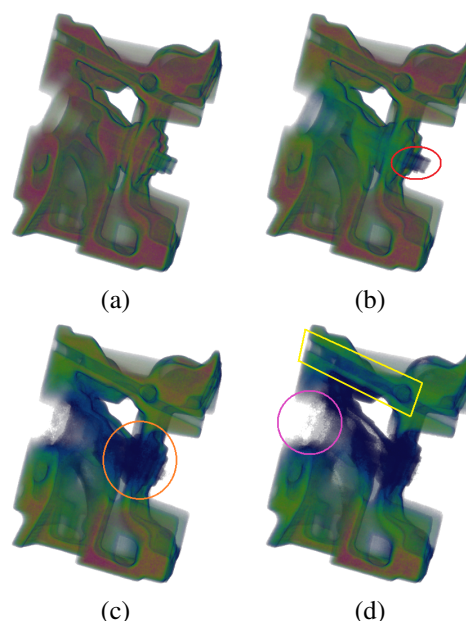


Figure 6: Comparison of choosing different values for p . (a), (b), (c) and (d) are the results obtained by choosing $p = 4, 9, 15$ and 20 , respectively, on the engine dataset for 1300 time steps. Internal parts such as the pipe (shown in the red ellipse), the outer rim around the pipe (shown in the orange circle) and the beam (shown in the yellow box) are captured in (b), (c) and (d), respectively. For large values of p in (d) some of the global shape information is missing (shown in the pink circle).

The direction of shapeton propagation is guided by the VGO. VGO has a parameter p which influences the result obtained. p is a user defined parameter that is used to decide the boundaries of the objects in a given volume data. The clarity of the boundary determines how clearly the different shapes are identified. The parameter p gives the user extra flexibility in deciding the object boundaries. A large p value would enhance the local shape differences within an object and hence result in more sub-objects. Therefore, by increasing the value of p the local internal objects within an object can be obtained. However, we tend to lose some of the global shape information for larger values of p . Thus, the final results obtained might vary both locally and globally for different values of p based on how well the objects are distinguished and how sharp the features are. All these effects of p are shown experimentally using the engine data in Figure 6.

Figure 6 shows the result of choosing different values of p on the engine dataset. Figures 6(a), (b), (c) and (d) show the result when $p = 4, 9, 15$ and 20 , respectively for 1300 time steps with a pre-defined distance

of 0.05. We can observe that different parts of the engine are captured by using different values of p . In Figure 6(b) where $p = 9$, the internal pipe (shown in the red ellipse) is separated which was not when $p = 4$ in Figure 6(a). Similarly, when $p = 15$ in Figure 6(c) the outer rim around the pipe (shown in the orange circle) is captured. Finally, when $p = 20$ in Figure 6(d) the beam of the engine (shown in the yellow box) is captured. We can see that by increasing the value of p more internal parts of the engine are captured as the local shape differences between these parts are enhanced. However, some of the global shape information is missing (shown in the pink circle) in Figure 6(d). This is because a high value of p divides the same object into much smaller sub-parts and because the pre-defined distance used was relatively high, these smaller sub-parts are not captured. The same is the reason why the internal pipe from Figure 6(b) is missing in Figures 6(c) and (d). In this way, different parts of the engine based on their shape can be obtained and analyzed using different values of p . This facilitates a better analysis and understanding of the data. As the convergence of shapetons can be monitored in real time, even though the value of p is changed, the new result can be obtained very fast. Thus, based on what features the user wishes to focus on and what features the user wants to analyze, different values of p can be selected.

5 APPLICATIONS

5.1 Transfer Function Design in Volumes

The shapetons accumulate all the shape information over different time steps while diffusing inside the volume. This information can be used to design a shape-based transfer function. The user can assign different colors and opacities to the final accumulated shapeton count, which forms a 1-D transfer function based on the shape information.

Figure 7 shows a volume rendered image of a CT chest dataset with a transfer function designed using the shape information obtained by our shapeton diffusion method. We were able to classify different parts of the data, such as the rib bones (shown in red), the sternum (shown in dark green), the clavicle bones (shown in magenta), the soapula (shown in fluorescent green), and small bones of the spinal cord (shown in blue) based on the shape information. Figure 7 shows all the segmented parts of the CT chest data by using our transfer function. All the ribs have similar curved shape and hence have been classified as the same shape indicated by the same color. Even the small but important part named xiphoid (greyish blue shown in the black circle), which is present at the tip of the sternum has been classified by the shape-based transfer function. The number of shapetons used was 65000 with a distance value of 0.05. The diffusion process was carried out for 1600 time steps, for a total time of 3.10 sec.

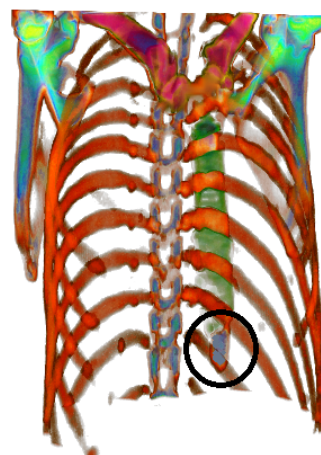


Figure 7: Volume rendering with the shape-based transfer function on the CT chest dataset. The rib bones (red), the sternum (dark green), the clavicle bones (magenta) and the soapula (fluorescent green) are obtained. The small bones of the spinal cord (blue), xiphoid (greyish blue in the black circle) - a small part present at the tip of the sternum are also classified.

5.2 Colon Cancer Detection

Colorectal cancer is the second leading cause of cancer related deaths in United States. Polyps are the precursors of colorectal cancer. Polyps are small protrusions of the tissue that grow out of the walls of the colon. Early detection and removal of these polyps is important for preventing colon cancer. We used our shapeton diffusion approach to detect the polyps on the colon surface, obtained from a CT scan of the patient's abdomen for virtual colonoscopy (VC) [HMK⁺97].

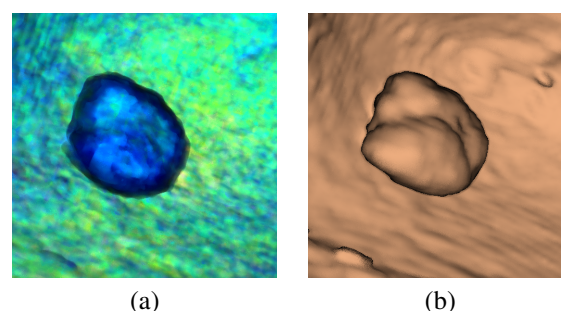


Figure 8: Polyp detection inside the colon using the shapeton diffusion method. (a) Polyp (shown in blue) detected using our approach; (b) Volume rendering of the corresponding location inside the colon confirming the presence of the polyp.

We used real volumetric colon data from VC to show the effectiveness of the shapeton diffusion process in polyp detection. The volumetric colon is electronically cleansed CT data. Figure 8 shows the result of the polyp detection using our shapeton diffusion method on the real colon data. Figure 8(a) shows the result obtained by our method and Figure 8(b) shows the volume ren-

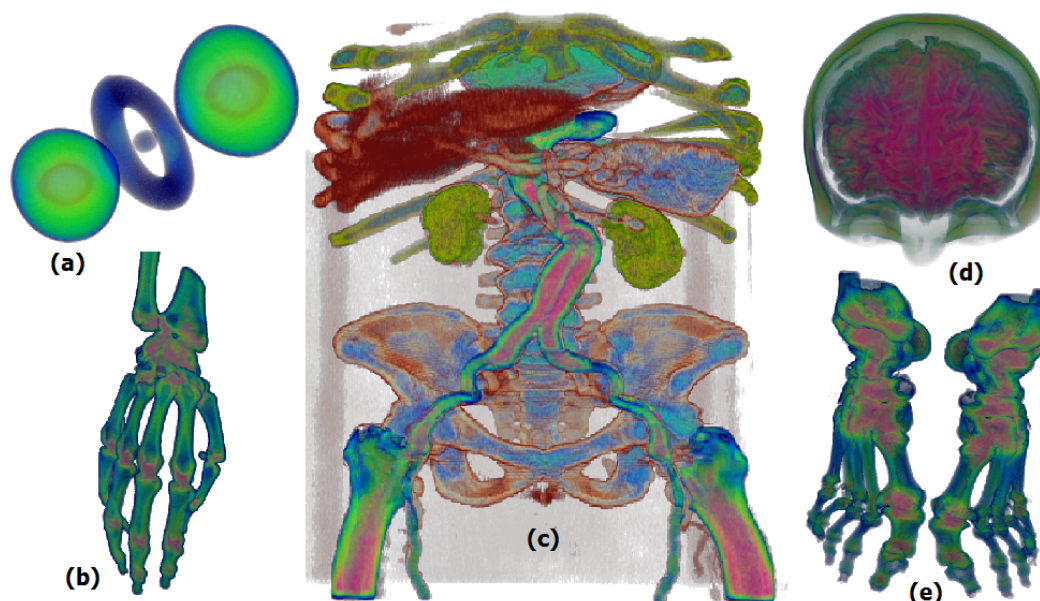


Figure 9: Classifying objects based on their shape using our shapeton diffusion approach on (a) hydrogen atom, (b) visible female hand, (c) CT abdomen, (d) MRI head, and (e) visible female feet volumetric datasets.

dering result of the corresponding location of the polyp inside the colon. Since polyps have a blob-like shape, different from the shape of the colon walls, we were able to successfully detect the polyps using our method. Figure 8(a) shows one such polyp (shown in blue) detected. We confirmed the position of the polyp by examining the corresponding location inside the colon volume data. This result can be seen in Figure 8(b). It took just 4000 time steps using 65000 shapetons to achieve this result. The p value was chosen to be 15. The reason to choose a high value for p is to get a clear boundary of the polyps. Since a smaller scale is needed for the polyp detection, a low distance value of 0.005 was chosen. The total time taken was 5.44 sec.

6 RESULTS

We used several datasets to demonstrate the efficiency of our method. Figures 9 (a)-(e) show the object classification capability of our approach based on the shape information for hydrogen atom, visible female hand, CT abdomen, MRI brain and visible female feet volumetric datasets, respectively. In Figure 9(a) both the orbitals of similar shape are clearly distinguished from the nucleus (center) and the orbit (around the nucleus) in a hydrogen atom as indicated by different colors. In Figure 9(c), the shape-based volume exploration of the CT abdomen reveals various organs such as the kidneys, liver, pancreas, and vital parts such as the aortic vessel, spinal cord and pelvic bones using 260000 shapetons and a pre-defined distance of 0.01. All the internal organs have different shapes and by virtue of our method, they have been identified successfully. Furthermore, it has just taken only 2.13 sec using 2600 time steps to obtain this result. In Figure 9(d), we are able to separate

the brain from the cranium and eye sockets in the MRI head data, using the shape-based transfer function designed by our approach. We used 65000 shapetons for a pre-defined distance value of 0.01 and 4160 time steps which accounted for a total time of 2.82 sec. Figures 9 (b) and (e) show that the bones and the joints between the bones are identified in the visible female hand and feet data, respectively. While we used 65000 shapetons and a pre-defined distance of 0.01 in both the cases, the number of time steps were 460 and 420 with a total time of 0.34 sec and 0.27 sec for the visible female hand and feet, respectively.

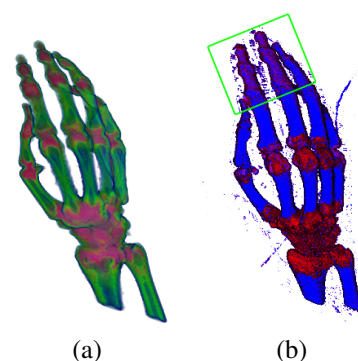


Figure 10: Visual comparison of the results obtained for the visible female hand dataset using (a) Our shapeton diffusion method and (b) The cumulative heat diffusion method.

We compared our approach with the cumulative heat diffusion (CHD) approach [GWK12], in terms of the running time per iteration and the number of time steps required to obtain visually similar or even better results. Table in Appendix 5 shows the comparison re-

sults using different volume datasets. For the sake of completion, we also provide a visual comparison of the results obtained by using our method with that of the results obtained using the CHD method using the visible female hand dataset (see Figure 10). Figure 10(a) shows the results obtained using the shapeton diffusion approach, while Figure 10(b) shows the result obtained by using the CHD method. In Figure 10(b), 1000 time steps were considered while in Figure 10(a) only 460 time steps were considered for a distance value of 0.01. Since we wanted to capture the local features, a smaller distance value was used. The same p value of 10 was used in both the cases. We can clearly observe that visually better results were obtained using our method compared to the CHD method. We can also see that a better distinction of shapes was obtained using our method even in very local regions, as indicated by the region in the green box in Figure 10(b). The joints have been clearly distinguished from the hand bones. Furthermore, the result was obtained in much less time compared to the CHD approach, further emphasizing the superiority of our method.

7 CONCLUSION AND FUTURE WORK

The main contribution of this paper is the real time shape analysis method in volumes using a Monte Carlo approach. Tiny massless particles, called shapetons, are diffused based on the VGO in a Monte Carlo manner. In addition, a new definition for the time step using a pre-defined distance is introduced. Unlike the conventional diffusion based methods, this method is independent of the size and resolution of the data. The final accumulated shapeton count after each time step would capture the shape information and helps in analyzing the data based on shape. The diffusion process can be monitored in real time and this facilitates a real time shape analysis of different features until a convergence state is reached. Furthermore, we discuss the properties of our method by presenting results using simple as well as complex datasets. Important applications of our method to colon cancer detection and transfer-function design have also been discussed, along with supporting results.

The results obtained using our method are influenced by many parameters such as the number of shapetons, the distance value, the number of time steps t , and the value of p . As discussed earlier, there is an optimum value for the number of shapetons used after which the time taken to obtain the results increases even though the number of shapetons is increased. Similarly, the time step t and the value of p have optimum values to obtain the best results based on the dataset used. As part of our future work, we plan to focus on finding a way to automatically decide the optimum values for all the parameters in order to obtain the best results.

8 REFERENCES

- [AK90] James Arvo and David Kirk. Particle transport and image synthesis. *SIGGRAPH Computer Graphics*, 24(4):63–66, September 1990.
- [ASC11] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. *ICCV Workshops*, pages 1626–1633, 2011.
- [BK10] Michael M. Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. *CVPR*, pages 1704–1711, 2010.
- [BSS94] P. Blasi, B. L. Saec, and C. Schlick. An importance driven monte-carlo solution to the global illumination problem. *Proceedings of the Eurographics Workshop on Rendering*, pages 173–183, 1994.
- [CS05] Carlos D. Correa and Deborah Silver. Dataset traversal with motion-controlled transfer functions. *IEEE Visualization*, pages 359 – 366, October 2005.
- [DEJ⁺99] Julie Dorsey, Alan Edelman, Henrik Wann Jensen, Justin Legakis, and Hans K hling Pedersen. Modeling and rendering of weathered stone. *SIGGRAPH*, pages 225–234, 1999.
- [DJ05] Craig Donner and Henrik Wann Jensen. Light diffusion in multi-layered translucent materials. *ACM Transactions on Graphics*, 24(3):1032–1039, July 2005.
- [GWK12] Krishna Chaitanya Gurijala, Lei Wang, and Arie E. Kaufman. Cumulative heat diffusion using volume gradient operator for volume analysis. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2069–2077, Dec 2012.
- [HMK⁺97] Lichan Hong, Shigeru Muraki, Arie E. Kaufman, Dirk Bartz, and Taosong He. Virtual voyage: interactive navigation in the human colon. *SIGGRAPH*, pages 27–34, 1997.
- [HSKK01] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiya L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. *SIGGRAPH*, pages 203–212, 2001.

- [JC98] Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. *SIGGRAPH*, pages 311–320, 1998.
- [Jen96] Henrik Wann Jensen. Global illumination using photon maps. *Proceedings of the Eurographics Workshop on Rendering Techniques*, pages 21–30, 1996.
- [JLD99] Henrik Wann Jensen, Justin Legakis, and Julie Dorsey. Rendering of wet materials. *Proceedings of the Eurographics Workshop on Rendering Techniques*, pages 273–282, 1999.
- [JMLH01] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. *SIGGRAPH*, pages 511–518, 2001.
- [KB07] Todd J. Kosloff and Brian A. Barsky. An algorithm for rendering generalized depth of field effects based on simulated heat diffusion. *Proceedings of the International Conference on Computational Science and its Applications*, pages 1124–1140, 2007.
- [LW96] Eric P. Lafortune and Yves D. Willems. Rendering participating media with bidirectional path tracing. *Proceedings of the Eurographics Workshop on Rendering Techniques '96*, pages 91–100, 1996.
- [OMMG10] Maks Ovsjanikov, Quentin Mérigot, Facundo Mémoli, and Leonidas J. Guibas. One point isometric matching with the heat kernel. *Computer Graphics Forum*, 29(5):1555–1564, 2010.
- [PGJA03] Stephen M. Pizer, Guido Gerig, Sarang C. Joshi, and Stephen R. Aylward. Multiscale medial shape-based analysis of image objects. *Proceedings of the IEEE*, 91(10):1670–1679, 2003.
- [PKK00] Mark Pauly, Thomas Kolliig, and Alexander Keller. Metropolis light transport for participating media. *Proceedings of the Eurographics Workshop on Rendering Techniques*, pages 11–22, 2000.
- [PM93] S. N. Pattanaik and S. P. Mudur. Computation of global illumination in a participating medium by Monte Carlo simulation. *The Journal of Visualization and Computer Animation*, 4(3):133–152, September 1993.
- [PRMH10] Jörg-Stefan Praßni, Timo Ropinski, Jörg Mensmann, and Klaus H. Hinrichs. Shape-based transfer functions for volume visualization. *IEEE Pacific Visualization Symposium*, pages 9–16, Mar 2010.
- [RJT08] Dennie Reniers, Andrei Jalba, and Alexandru Telea. Robust classification and analysis of anatomical surfaces using 3D skeletons. *VCBM*, pages 61–68, 2008.
- [Sal07] Christof Rezk Salama. GPU-based Monte Carlo volume raycasting. *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 411–414, 2007.
- [SF95] Jos Stam and Eugene Fiume. Depicting fire and other gaseous phenomena using diffusion processes. *SIGGRAPH*, pages 129–136, 1995.
- [SOG09] Jian Sun, Maks Ovsjanikov, and Leonidas J. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009.
- [Sta95] Jos Stam. Multiple scattering as a diffusion process. *Proceedings of the Eurographics Workshop on Rendering Techniques*, pages 41–50, 1995.
- [VBCG10] Amir Vaxman, Mirela Ben-Chen, and Craig Gotsman. A multi-resolution approach to heat kernels on discrete surfaces. *ACM Transactions on Graphics*, 29:121:1–121:10, July 2010.
- [Wan13] Lei Wang. Research of air pollution dispersion visualization based on GPU and volume rendering. *Proceedings of the 2nd International Conference On Systems Engineering and Modeling (ICSEM)*, 2013.
- [YSABNSK00] C.-F. Westin, Y. Sato, S. Nakajima, A. Bhalerao, S. Tamura, N. Shiraga, and R. Kikinis. Tissue classification based on 3D local intensity structure for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):160–180, 2000.

Synthetic-Real Domain Adaptation for Probabilistic Pose Estimation

Omar Del-Tejo-Catala
Instituto Tecnologico de
Informatica
Camino de Vera, S/N
Spain 46022, Valencia
odeltejo@iti.es

Javier Perez
Instituto Tecnologico de
Informatica
Camino de Vera, S/N
Spain 46022, Valencia
javierperez@iti.es

Jose-Luis Guardiola
Universidad Politecnica
de Valencia
Camino de Vera, S/N
Spain 46022, Valencia
joguagar@iti.es

Alberto J. Perez
Universidad Politecnica
de Valencia
Camino de Vera, S/N
Spain 46022, Valencia
aperez@disca.upv.es

Juan-Carlos
Perez-Cortes
Universidad Politecnica
de Valencia
Camino de Vera, S/N
Spain 46022, Valencia
jcperez@iti.es

ABSTRACT

Real samples are costly to acquire in many real-world problems. Thus, employing synthetic samples is usually the primary solution to train models that require large amounts of data. However, the difference between synthetically generated and real images, called domain gap, is the most significant hindrance to this solution, as it affects the model's generalization capacity. Domain adaptation techniques are crucial to train models using synthetic samples. Thus, this article explores different domain adaptation techniques to perform pose estimation from a probabilistic multiview perspective. Probabilistic multiview pose estimation solves the problem of object symmetries, where a single view of an object might not be able to determine the 6D pose of an object, and it must consider its prediction as a distribution of possible candidates. GANs are currently state-of-the-art in domain adaptation. In particular, this paper explores CUT and CycleGAN, which have unique training losses that address the problem of domain adaptation from different perspectives. This work evaluates a patch-wise variation of the CycleGAN to keep local information in the same place. The datasets explored are a cylinder and a sphere extracted from a Kaggle challenge with perspective-wise symmetries, although they holistically have unique 6D poses. One of the main findings is that probabilistic pose estimation, trained with synthetic samples, cannot be solved without addressing domain gap between synthetic and real samples. CUT outperforms CycleGAN in feature adaptation, although it is less robust than CycleGAN in keeping keypoints intact after translation, leading to pose prediction errors for some objects. Moreover, this paper found that training the models using synthetic-to-real images and evaluating them with real images improves the model's accuracy for datasets without complex features. This approach is more suitable for industrial applications to reduce inference overhead.

Keywords

Pose Estimation, CycleGAN, Image-to-image, Graph Neural Networks, UNet, Domain adaptation, CUT, Symmetry Robust Pose Estimation

1 INTRODUCTION

Obtaining a large number of real samples to train a model is often expensive, making it infeasible for

many industrial applications. To address this issue, researchers have proposed training models with synthetic samples, which are cheaper and easier to produce on a large scale [Sve21a, Che21a, Sha22a]. However, synthetic samples cannot simulate every nuance in the real samples domain, resulting in a domain gap. Techniques such as domain adaptation and domain randomization have been proposed to address this issue. This paper uses domain adaptation techniques to reduce the domain gap between synthetic and real samples.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Pose estimation has many practical applications, including robot grasping, virtual reality, and robot localization. Various contexts have been proposed to evaluate different working scenarios, such as pose estimation in cluttered scenes [Buk20a, Ste13a, YuX17a], relative camera pose estimation [Che21b], and pose refinement [YiL20a]. This article focuses on multiview probabilistic uncluttered pose estimation of perspective-symmetric objects, as presented in [Oma21a].

The data capturing environment for this work revolves around a quality inspection system [Jua18a], which comprises several cameras distributed along a sphere that captures objects under inspection in a controlled environment. The objects are free-falling, and the cameras capture them without occluded faces. However, the 6D pose is uncontrolled, and some keypoints are not visible from all perspectives. Hence, the algorithm's first stage must account for auto-occlusions and partially informative perspectives, which might only allow the algorithm to narrow down the range of possible solutions. Therefore, each image prediction should be an estimation of the true 6D pose, and the final model's prediction is computed by combining multiple views.

This paper evaluates how different state-of-the-art domain adaptation techniques affect probabilistic pose estimation, which is also compared against the baseline, i.e., not handling domain gap. Objects under inspection might have perspective-relative symmetry axes, although, considering the object as a whole, they might not be symmetric (see Figure 1). Therefore, the probabilistic pose estimation model must be able to model and detect those axes internally. The domain adaption algorithm must not interfere with that model's capacity.

Furthermore, this article compares pose estimation performed separately in the synthetic and real domains. Thus, it presents the results in several scenarios:

- Case 0: Training the pose estimation model with synthetic images and evaluating with real images.
- Case 1: Training the pose estimation model with domain-translated synthetic images, i.e., synthetic-to-real images, and evaluating with real images.
- Case 2: Training with synthetic images and testing with domain-translated real images, i.e., real-to-synthetic images.

The work is organized as follows. Section 2 presents the pose estimation's state-of-the-art and several methods to address the domain gap. Section 3 presents the datasets evaluated along with the algorithms used, their structure, and their losses. Section 4 presents the results achieved by the different domain adaption algorithms for the pose estimation tasks related to the

datasets. Section 4.3 compares and discusses the results achieved with previous work in pose estimation and domain adaptation. Section 5 summarizes the articles' findings.

2 RECENT SOLUTIONS

2.1 Pose Estimation

Pose estimation is a well-established field that has been extensively studied [Wad17a, Tom18a, Buk20a]. Recent approaches include per-pixel pose inference, bounding box detection, prediction refinements, and direct pose regression. However, many of these models do not effectively handle symmetries. Some models manually address symmetries during training [Wad17a, Buk20a], while others account for them during metric computation, such as ADD and ADDS [Tom18a, Bug18a], used in the LINEMOD dataset [Ste13a]. However, none of these approaches automatically model an object's symmetries. Therefore, we selected the work in [Oma21a], which proposes a probabilistic multiview approach for pose estimation. This approach enables the combination of multiple hypotheses based on uncertainty by modeling the probability distribution of the object's rotation, which may be uncertain from certain viewpoints.

2.2 Addressing domain gap

Domain bias [Csu2017] is a problem that models often encounter when trained on a single narrow-range dataset, limiting their generalization capabilities. Inconsistencies arise when models are trained with synthetic images and evaluated with real ones.

To address the difference between synthetic and real images, the literature primarily focuses on two approaches: domain randomization and domain adaptation.

Domain randomization [Sve21a, Tob18a, Tob17a] involves modifying synthetic generation hyperparameters to increase the diversity of synthetic image generation. In the context of pose estimation, such hyperparameters may include the position of the cameras, object textures, lighting, or scene context. Although this technique attempts to emulate the possible variability in real scenarios, it does not leverage the information from real samples and may not address crucial variability that could impact the model's generalization capability. For example, [Sve21a] generates multiple scenarios with cat and dog models in Unity, with different camera positions, object textures, and occlusion configurations. It evaluates the model using the public Kaggle Cats-Dogs dataset. [Tob17a] employs synthetic training with various textures and occlusions to address the task of object localization for robotic manipulation.

Domain adaptation [Che21a, Sha22a, Jac21a] minimizes the domain gap by decreasing the difference

between both domains' images or features. Usually, this is addressed using image-to-image transformation between domains, mainly using unsupervised techniques like autoencoders or GANs. Such approaches include employing CycleGANs [Che21b] or Contrastive Unpaired Translation (CUT) [Tae20a].

In [Jac21a], pose estimation and domain adaptation are addressed from a single-perspective non-symmetric approach. It asserts that CUT outperforms CycleGAN for domain adaptation in pose estimation. Domain adaptation techniques must keep keypoints along with the object symmetries intact.

Similar to [Jac21a], this paper evaluates different domain adaptation techniques, although it focuses on different symmetric objects. Thus, a slight variation of the CycleGANs [Che21b] and the CUT algorithm [Tae20a] are employed to reduce the domain gap in probabilistic pose estimation.

3 PROPOSED SOLUTION

3.1 Dataset

This paper's dataset expands some of the datasets presented in [Oma21a] and published in Kaggle [ITI21a]. We selected the cylinder, which has a perspective symmetry but a single valid object-level prediction, and the sphere, whose perspectives are mostly uninformative, except for the ones containing the "T".

The cylindrical object's bases contain a carved triangle on one side and a square on the other. The square and the triangle are aligned to create the object's reference point, albeit no camera can see both simultaneously. Therefore, the pose of the cylinder can only be calculated by combining the predictions from the cameras that were able to see both polygons separately. As presented in Figure 1, any camera that captures some polygon can predict the X-axis orientation. Nonetheless, for the Y-axis, the cameras that captured the triangle should return three equally likely Y-axis predictions. In comparison, the cameras that captured the square should return four equally likely Y-axis predictions. Finally, the combination of all separate Y-axis predictions must return a single likeliest Y-axis.

The sphere's only informative keypoint is the carved "T" on its surface. Most views are unable to see the "T" due to auto-occlusions. Those views only provide the information that they cannot see the "T". Thus, the prediction should be an equally like distribution over the subspace of 3D rotations that yield perspectives without the "T". Nonetheless, any perspective that completely captures the keypoint should return a unique 6D pose prediction.

The corresponding 3D CAD files were employed to generate real objects using a 3D printer. The files describe a cylinder with 50 millimeters of height and a

diameter of 25 mm and a sphere of 30mm of diameter. Like any other generation process, this procedure is prone to generate singularities in the object's texture that may disclose the 6D pose instead of using the intended keypoints. Pose estimation algorithms can then overfit these singularities leading to errors in a real environment. Moreover, even the texture and keypoints present in the object's ideal mesh have some variability in the real generated objects. A comparison between ideal and real object captures can be seen in figure 1. For instance, the printed sphere has a stain on one side that may affect the pose estimation algorithm (see Figure 1e). This object provides a more complex non-uniform texture to evaluate the domain adaptation algorithm.

The printed objects were captured in an industrial quality inspection system, presented in [Jua18a], that comprises a multicamera environment. Capturing objects inside this system leads to sixteen camera images per capture. As the pose estimation algorithm presented in [Oma21a] leverages multiview images, their individual predictions are joined to a unique 6D pose prediction in testing phase.

The images from the printed objects contain the object in many different poses, as the objects are captured free falling inside a controlled environment [Jua18a]. Thus, the groundtruth pose for each capture must be manually labeled. Using OpenCV's CVAT labeling tool, some reference keypoints were selected from multiple views. These keypoints' correspondence was used to compute de 6D pose of each launch.

To train the models, this article synthetically simulates the real environment to capture the 3D CAD files, similar to other domain randomization authors [Sve21a]. The real-world objects are captured in a 16-camera sphere-distributed environment with diffuse white lighting; thus, the backgrounds are entirely white. Some sample images can be seen in Figure 1.

This dataset was made publicly available in Kaggle [ITI23a]¹.

3.2 Probabilistic pose estimation algorithm

This algorithm takes an image and some cropping-related information, i.e., pixel coordinate of the bounding box center, and a scaling factor applied to resize the image to 128×128 resolution. Then, the algorithm predicts the object's translation and a probability distribution for the rotation. This probability distribution is a discretization of the rotation's $SO(3)$ group \mathcal{R}^9 space, which is compressed up to \mathcal{R}^6 . In other words, it predicts a discretization of the rotation matrices' X and Y

¹ <https://www.kaggle.com/ds/2947388>

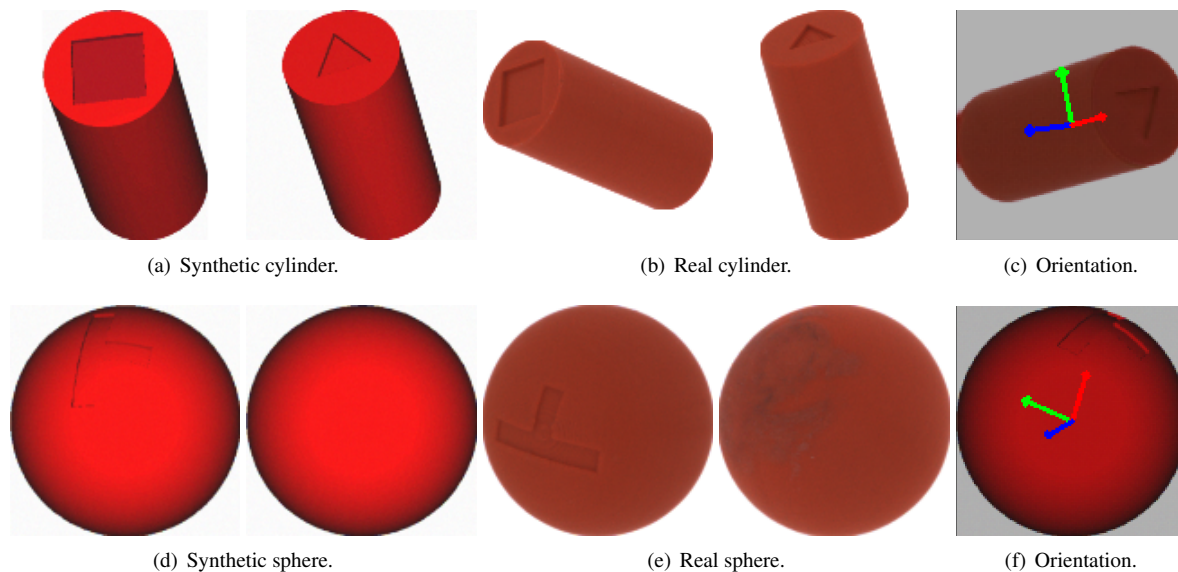


Figure 1: Train and test capture comparison to show the domain gap. The cylinder's X-axis corresponds to the object's longitudinal axis. The Y-axis points from the polygons' centers to the center of the only side of the polygon that aligns with some opposite polygon's side. The sphere's X-axis is the vector from the object's centroid to the "T". The Y-axis corresponds to the vertical orientation of the "T". In (c) and (f), red, green and blue axes correspond to X, Y and Z axes, respectively.

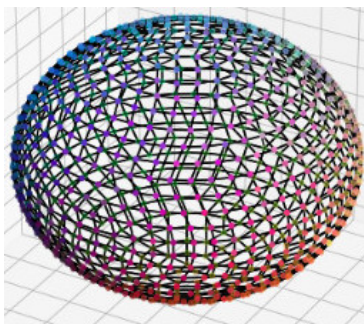


Figure 2: Discretization of the rotations' axis space [Oma21a].

axes components. As described in [Oma21a], this can be done without information loss due to the rotation matrices orthonormality.

These axes, belonging to the \mathcal{R}^3 unitary sphere, are discretized sampling N equidistant points. Hence, the algorithm's output is a tensor of $N \times 2$ for each image. The space discretization and how each point is connected as a graph can be seen in figure 2. The predictions are combined for all the images in a capture. Finally, this distribution is queried for the likeliest positions for the X and Y axes; thus, the Z axes can be inferred. More detail of the algorithm can be seen in [Oma21a].

The predictions can be unwrapped for visualization purposes. This projection consists of extracting the azimuth and elevation of each \mathcal{R}^3 point (i.e., a rotation matrix axis) and printing them in an image, being the intensity of each pixel related to the axis likelihood. We

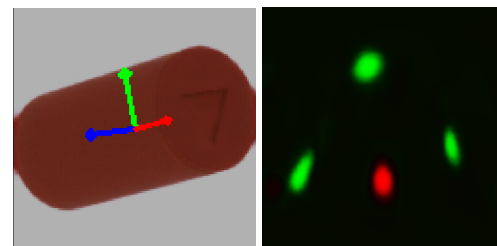


Figure 3: Prediction when the triangle is visible. The predicted distributions, i.e., one for the X-axis and one for the Y-axis, are projected to 2D space for visualization purposes. Red color corresponds to the X-axis activations and green to the Y-axis activations. It shows a single activation for the X-axis, as the direction of the longitudinal axis can be determined, and three activations for the symmetric triangle rotations.

can visualize each axis in different color channels, i.e., red for X-axes and green for Y-axes. Figure 3 shows the three possible Y-axis clustered activations that occur when predicting using the triangle due to its symmetries. Three equally likely rotations can be predicted using the likeliest axis of each green y-axis cluster and the likeliest axis of the red X-axis cluster. All those three equally likely activations are related to an identical image representation.

3.3 CycleGAN

To perform cross-domain evaluation, we must address the problem of the domain gap between training and testing sets. This paper employs CycleGANs as they force the image transformation to keep the information

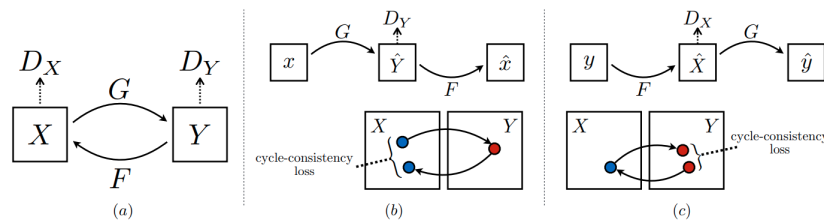


Figure 4: Cycle consistency from CycleGAN extracted from the original paper [Jun17a]. a) shows the structure of the network, X and Y being the domains. b) shows the cycle consistency loss for images from the X domain. c) shows the cycle consistency loss for the images from the Y domain.

when transforming the domain to be able to reconstruct the original image.

CycleGANs are divided into two different transformations, each with their corresponding generator G and discriminator D . Having two different domains d_A and d_B , we call $G_{A \rightarrow B}$ to the generator that transforms from domain d_A to d_B , $G_{B \rightarrow A}$ to the one that transforms domain d_B to d_A , D_A to the domain d_A discriminator, and D_B to the domain d_B discriminator.

3.3.1 Losses

CycleGANs have two losses: a cycle consistency loss and a discrimination confidence loss. Cycle consistency loss corresponds to the loss of projecting an image from one domain to the other and reprojecting it back to the original domain. This is, being X_A images from domain d_A and X_B images of domain d_B :

$$\hat{X}_A = G_{B \rightarrow A}(G_{A \rightarrow B}(X_A)) \quad (1)$$

$$\hat{X}_B = G_{A \rightarrow B}(G_{B \rightarrow A}(X_B)) \quad (2)$$

$$\mathcal{L}_{GCC} = \|\hat{X}_A - X_A\|_1 + \|\hat{X}_B - X_B\|_1 \quad (3)$$

A visual representation of the process can be seen in Figure 4.

Confidence loss corresponds to the traditional GAN loss originating from the discrimination of synthetic and real images. It is different for the generator \mathcal{L}_{GC} and the discriminator \mathcal{L}_{DC} . Thus, these losses are described as follows:

$$\mathcal{L}_{GC} = \|D_B(G_{A \rightarrow B}(X_A))\|_2 + \|D_A(G_{B \rightarrow A}(X_B))\|_2 \quad (4)$$

$$\begin{aligned} \mathcal{L}_{DC} = & \|1 - D_B(G_{A \rightarrow B}(X_A))\|_2 + \\ & \|1 - D_A(G_{B \rightarrow A}(X_B))\|_2 + \\ & \|D_A(X_A)\|_2 + \|D_B(X_B)\|_2 \end{aligned} \quad (5)$$

The generator networks are updated with the sum of \mathcal{L}_{GC} and \mathcal{L}_{GCC} , whilst the discriminators are updated with \mathcal{L}_{DC} .

3.3.2 Generators

As other authors have previously proposed [Tae20a, Dmi22a], this article uses UNet-like [Ola15a] structure for the generative network. A pretrained VGG16 performs as the network's encoder, extracting features at four different resolutions. Those features are then processed and upsampled by a decoder network, similar to a traditional UNet. The only trainable part of the network is the decoder.

Additionally, slight variations must be considered to keep the keypoints in place as best as possible. Therefore, patches of 32×32 were employed to force the network to store the information in an enclosed space. The image is reconstructed from its patches after each transformation. This process can be seen in Figure 5.

3.3.3 Discriminators

The discriminator also employs a pretrained VGG16 network that returns the features at two different resolutions: $32 \times 32 \times 256$ and $16 \times 16 \times 512$. At each resolution, the discriminator extracts trainable features and predicts their probability of being generated. Those probability maps are resized to the largest resolution and averaged.

It is a common practice to add noise to the discriminator input images to avoid overfitting. Thus, 0.1 intensity uniform noise was applied as a preprocess to the discriminator images.

In summary, discriminators take noisy $128 \times 128 \times 3$ images and return a 32×32 tensor with the probability of an image region being generated.

3.4 Contrastive Unpaired Translation

Contrastive Unpaired Translation (CUT) [Tae20a] is a domain adaptation technique that learns a transformation between two domains. It is designed without CycleGAN's cycle consistency loss and without reprojecting to the original domain. Therefore, it is faster than CycleGAN and is reported to achieve better results [Tae20a]. It relies on mutual information maximization using PatchNCE loss. This loss minimizes the distance between equivalent image patches in both domains, i.e., comparing the original patch with its translated version (positive) while maximizing the distance

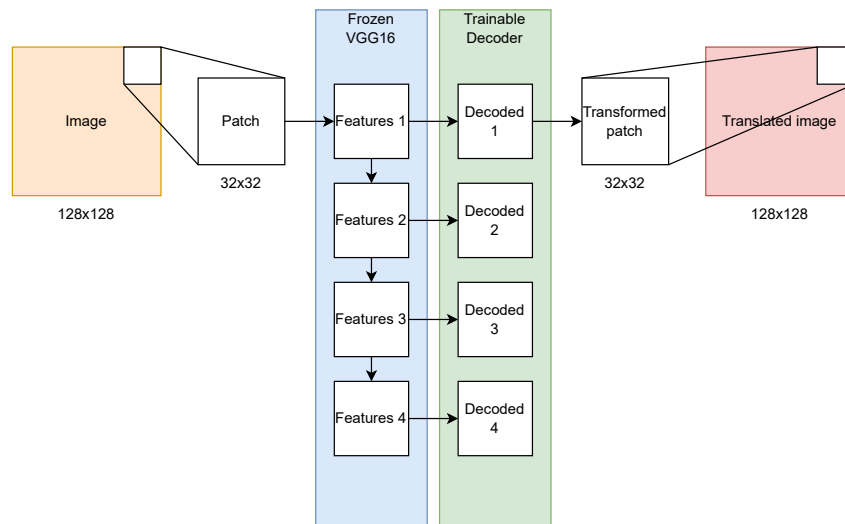


Figure 5: Image domain transformation process.

with the remaining image patches (negatives). This paper also studies the inclusion of negatives from other images but concludes that it hampers training.

4 RESULTS

4.1 Experimental setup

This article's experiments were designed using TensorFlow 2.10. Domain adaptation techniques were trained using 3000 synthetic images and 160 real images. Random rotations were applied to both domain images for data augmentation. The CUT model was trained using the code provided by the authors in Github¹. It was trained for 400 epochs, removing the default random cropping and scaling and adding random rotations. Thus, the original $128 \times 128 \times 3$ image resolution is kept and there is some data augmentation. CycleGAN was trained by adding 0.1 standard deviation gaussian noise to the upsampling decoder inside synthetic-to-real generator, and data was augmented with random rotations.

The pose estimation models were trained using 256000 synthetic images, validated using 1600 synthetic images and tested using 880 real images. During training, random noise, dropout and L2 norm were added to increase generalization.

4.2 Case results

Three cases are proposed to test the two different domain transformations scenarios:

- Case 0 - no domain transformation. The model was trained with synthetic images, data augmentation, and regularization added to the model.

- Case 1 - training with real samples. Training images are translated to the real domain. The last four training checkpoints were used to include some variability in the domain transformation using CUT models. Therefore, the same synthetic image results in different real images. CycleGAN model has some embedded randomness while projecting from synthetic to real domains.
- Case 2 - evaluating with synthetic samples. Real images are projected to the synthetic domain to remove production singularities from the object's texture.

Some domain transformation samples can be seen in Figure 6. On the one hand, as seen from a visual inspection, when projecting from the real to the synthetic domain (Case 2), images still have some artifacts that do not belong to the synthetic domain. These include some modifications in the object's keypoints (Figure 6e) for CUT and random dirt (Figure 6f) for CycleGAN. On the other hand, both domain adaptation techniques perform correctly for this task when projecting from synthetic to real domain (Case 1). Moreover, as CycleGAN has intrinsic randomness, it learned to add some artifacts (see the square inside in Figure 6c).

Despite our best efforts, we could not train a valid CUT domain adaptation model for the sphere dataset. A sphere image without a "T" is still valid for the discriminator. Thus, these models converge to a state where all images projected do not keep the "T" keypoint.

For a quantitative comparison, Frechet-Inception-Distance (FID) [Mar17a] was used to measure which domain adaptation algorithm performed best. This score measures the distance between the original and domain-shifted image distributions. The score's magnitude depends on the dataset; thus, different dataset's FID scores cannot be compared.

¹ <https://github.com/taesungp/contrastive-unpaired-translation/>. Commit: 7 Jun 2022.

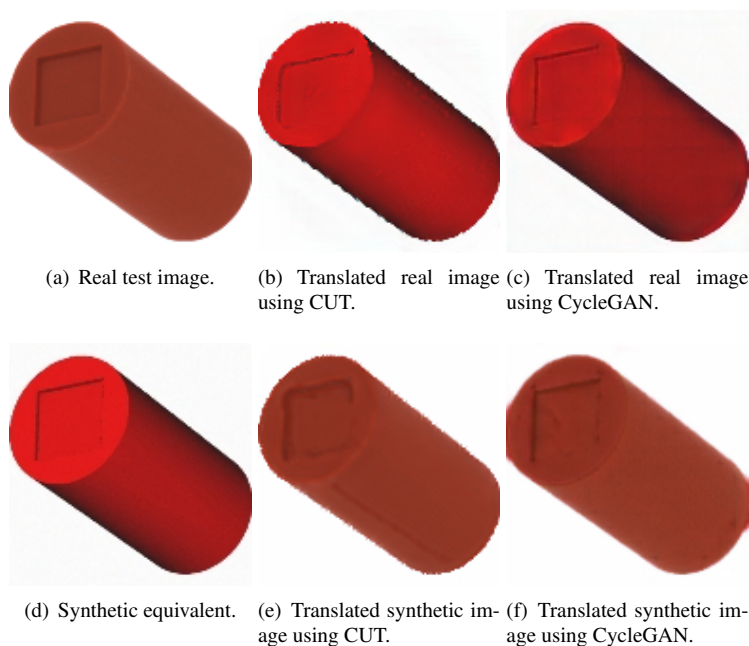


Figure 6: Results for cylinder domain transformation.

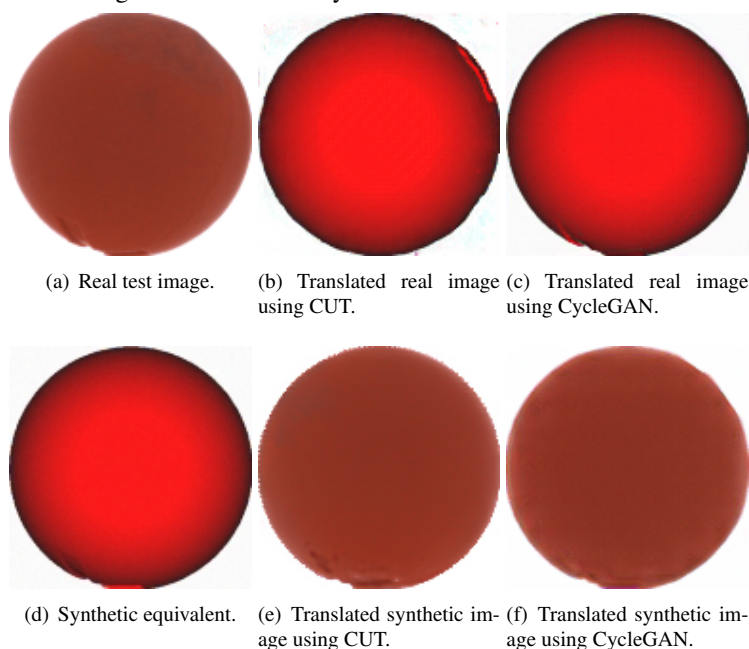


Figure 7: Results for cylinder domain transformation.

As seen for the cylinder case in Table 1, CycleGAN performs better than CUT in Case 1, but CUT outperforms CycleGAN in Case 2. This difference is probably because CycleGAN has higher variability in the real domain due to its randomness. However, as stated above, CycleGAN has proven to be more robust than CUT for the sphere.

As seen in Table 2, training the network directly with real images improves the model's accuracy for the cylinder dataset. However, the sphere dataset results show an improvement when the pose estimation model

is trained with synthetic images. A more in-depth analysis of the results is performed in the following section.

4.3 Discussion

As stated above, one of the main findings of this work is that training the model with synthetic-to-real images or using synthetic images provides different results depending on the object under inspection.

Unlike having to translate incoming images dynamically, the synthetic-to-real training approach has the

Case	Domain adaptation	FID	
		Cylinder	Sphere
Case 1	CycleGAN	49.33	47.25
Case 1	CUT	53.13	96.41*
Case 2	CycleGAN	82.45	43.45
Case 2	CUT	64.77	108.37*

Table 1: Frechet-Inception-Distance (FID) between different domain adaptation methods. (*) marks the experiments where the domain adaptation did not yield visually acceptable results.

Validation					
Case	Domain adaptation	Rotation loss (degrees)		Translation loss (mm)	
		Cylinder	Sphere	Cylinder	Sphere
Case 0 & 2	None	0.9 ± 0.05	2.0 ± 0.09	0.7 ± 0.03	1.3 ± 0.06
Case 1	CUT	1.4 ± 0.08	$1.5 \pm 0.08^*$	0.7 ± 0.03	1.3 ± 0.06
Case 1	CycleGAN	1.0 ± 0.06	1.2 ± 0.06	0.7 ± 0.03	1.3 ± 0.06
Test					
Case 0	None	111.9 ± 7.23	130.9 ± 4.31	2.6 ± 0.14	3.8 ± 0.63
Case 1	CUT	1.9 ± 0.11	$158.3 \pm 4.57^*$	2.7 ± 0.14	3.8 ± 0.63
Case 1	CycleGAN	5.2 ± 2.03	10.9 ± 3.05	2.6 ± 0.14	3.8 ± 0.63
Case 2	CUT	23.0 ± 5.45	$123.1 \pm 5.12^*$	2.7 ± 0.14	3.8 ± 0.63
Case 2	CycleGAN	23.7 ± 5.82	5.7 ± 0.39	2.7 ± 0.14	3.8 ± 0.63

Table 2: Validation and testing phase pose errors for each case and domain adaptation method. (*) marks the experiments where the domain adaptation did not yield visually acceptable results. Case 0 and case 2 share validation results because their validation set is synthetic, although test sets are different, i.e., synthetic and real images, respectively.

added benefit that no overhead is added after deploying the model; all overhead devoted to domain adaptation is performed during training. Contradicting the hypotheses presented by [Sha22a], at least for some objects without high variability textures, these findings show that training using synthetic-to-real images might be helpful to emphasize features that are more subtle in the synthetic domain by translating them to the real domain. The synthetic image generation algorithm might have a limited capacity for generating hyper-realistic object captures with precise shadows. Thus, an algorithm that projects those synthetic images to a domain with more perceptible features will improve the model's performance.

However, for cases where the textures are more complex, for instance, high variability textures such as wood or textures with anomalies, working only on the synthetic domain improves the pose estimation's results. This improvement arises because domain adaptation and pose estimation techniques trained on the real domain would require to be fit to a more challenging texture distribution. The synthetic domain provides a more accessible working environment for both models.

Furthermore, as can be seen in Table 2 from the comparison of Case 0 with any other case, applying some domain adaptation method significantly improves the model's generalization capacity. In most cases, the domain gap between synthetic and real samples cannot

be solved by adding traditional regularization to the model.

CycleGAN, due to its patch-wise domain transformation to the original domain, keeps the object's keypoints in place better than CUT. This difference leads to a more robust domain adaptation algorithm for different objects. However, using CUT improves the model's synthetic-real generalization capacity for some objects, which is reflected in the model's cylinder test accuracy. Not only is it CUT a lighter algorithm, as it only performs one-way domain transformation, but it also achieves a better transformation, though less generic. Thus, as found in [Tae20a], CUT's PatchNCE loss outperforms CycleGAN's cycle consistency in performing domain transformations to similar features, but the fact that those features do not require reprojection to the original domain leads to information loss, which is highly detrimental for pose estimation.

5 CONCLUSION

Considering the results, the main conclusion is that domain gap should be addressed somehow to train a model with synthetic samples to solve the pose estimation task effectively, being CycleGAN and CUT valid for this purpose. From the baseline, where the model could not estimate the object rotation, adding some domain adaptation technique (either in case 1 or case 2 scenarios) reduced the rotation error below 6° for both datasets. Additionally, to address the domain gap effectively, this work found that training in the synthetic

or real domains depends mainly on the object texture's complexity. However, performing domain adaptation in training phase, i.e., training with real images, reduces computational costs after deployment. Computational costs in industrial applications are crucial, and adding computational costs to the offline training phase has a lower impact than domain transforming every incoming image to infer the 6D pose.

Moreover, this paper demonstrates that CycleGAN largely preserves the keypoint's positions so that a pose estimation algorithm can be trained to a reasonable accuracy. This fact is especially remarkable in this dataset due to the scarcity of keypoints from which to extract rotational information. CUT performs a better feature transformation, as it outperforms CycleGAN in the cylinder dataset. However, it might not prevent the loss or modification of keypoints during domain transformation, which leads to poor results for the sphere dataset.

As further research, including objects with anomalous textures to the domain adaptation and pose estimation algorithms, might provide valuable insight to improve the algorithms' robustness.

5.1 Acknowledgements

This work was funded with grants from the Generalitat València with grant number IMAMCA/2023/11, the European Regional Development Fund (ERDF) with grant number IMDEEA/2022/46, and CDTI's CELIA project with grant number CER-20211022.

6 REFERENCES

- [Csu2017] Gabriela Csurka. A comprehensive survey on domain adaptation for visual applications. *Advances in Computer Vision and Pattern Recognition*, 1-35, 2017.
- [Ste13a] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *Computer Vision – ACCV 2012*, pages 548–562. Springer Berlin Heidelberg, 2013.
- [Ola15a] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.
- [Jun17a] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [Tob17a] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.
- [Wad17a] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 1530–1538, nov 2017.
- [YuX17a] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes *Robotics: Science and Systems (RSS)*, 2018.
- [Bug18a] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 292–301, nov 2018.
- [Jua18a] Juan-Carlos Perez-Cortes, Alberto Perez, Sergio Saez-Barona, Jose-Luis Guardiola, Ismael Salvador Igual, and Sergio Sáez. A system for in-line 3d inspection without hidden surfaces. *Sensors*, 18:2993, 09 2018.
- [Tob18a] Josh Tobin, Lukas Biewald, Rocky Duan, Marcin Andrychowicz, Ankur Handa, Vikash Kumar, Bob McGrew, Alex Ray, Jonas Schneider, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Domain randomization and generative models for robotic grasping. *IEEE International Conference on Intelligent Robots and Systems*, pages 3482–3489, 2018.
- [Tom18a] Tomáš Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Caner Sahin, Fabian Manhardt, Federico Tombari, Tae Kyun Kim, Jiří Matas, and Carsten Rother. BOP: Benchmark for 6D object pose estimation *Computer Vision – ECCV*, 2018.
- [Buk20a] Yannick Bukschat and Marcus Vetter. EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach *Preprint*, nov 2020.

- [YiL20a] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation. *International Journal of Computer Vision*, 128(3):657–678, mar 2020.
- [YiL20a] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation. *International Journal of Computer Vision*, 128(3):657–678, mar 2020.
- [Tae20a] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 2020.
- [Che21a] Chen Li and Gim Hee Lee. From synthetic to real: Unsupervised domain adaptation for animal pose estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1482–1491, 2021.
- [Che21b] Chenhao Yang, Yuyi Liu, and Andreas Zell. Relative camera pose estimation using synthetic data with domain adaptation via cycle-consistent adversarial networks. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 102, 8 2021.
- [ITI21a] ITI-Instituto Tecnológico de Informatica. Pose estimation *Kaggle Dataset*, 2021.
- [Jac21a] Jack Langerman, Ziming Qiu, Gábor Sörös, Dávid Sebők, Yao Wang, and Howard Huang. Domain adaptation of networks for camera pose estimation: Learning camera pose estimation without pose labels *Preprint*, 2021.
- [Oma21a] Omar Del-Tejo-Catala, Jose-Luis Guardiola, Javier Perez, David Millan Escrivá, Alberto J. Perez, and Juan-Carlos Perez-Cortes. Probabilistic pose estimation from multiple hypotheses, *Preprint*, 2021.
- [Sve21a] Svetozar Zarko Valtchev and Jianhong Wu. Domain randomization for neural network classification. *Journal of Big Data*, 8:94, 12 2021.
- [Dmi22a] Dmitrii Torbunov, Yi Huang, Haiwang Yu, Jin Huang, Shinjae Yoo, Meifeng Lin, Brett Viren, and Yihui Ren. Uvcgan: Unet vision transformer cycle-consistent gan for unpaired image-to-image translation *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.
- [Sha22a] Sota Shoman, Tomohiro Mashita, Alexander Plopski, Photchara Ratsamee, and Yuki Urishi. Real-to-synthetic feature transform for illumination invariant camera localization. *IEEE Computer Graphics and Applications*, 42:47–55, 2022.
- [ITI23a] ITI-Instituto Tecnológico de Informatica. Pose estimation with domain adaptation *Kaggle Dataset*, 2023.
- [Mar17a] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium *Advances in Neural Information Processing Systems*, 30, 2017.

Error-Robust Indoor Augmented Reality Navigation: Evaluation Criteria and a New Approach

Oliver Scheibert
TH Köln
Steinmüllerallee 1
51643, Gummersbach, Germany
scheibert.oliver@gmail.com

Jannis Möller Steve Grogorick Martin Eisemann
TU Braunschweig
Mühlenpfordtstr. 23
38106, Braunschweig, Germany
{moeller,grogorick,eisemann}@cg.cs.tu-bs.de

ABSTRACT

Tracking errors severely impact the effectiveness of augmented reality display techniques for indoor navigation. In this work we take a look at the sources of error and accuracy of existing tracking technologies. We derive important design criteria for robust display techniques and present objective criteria that can be used to evaluate indoor navigation techniques without or in preparation of quantitative user studies. Based on these criteria we propose a new error tolerant display technique called Bending Words, where words move along the navigation path guiding the user. Bending Words outranks the other evaluated display techniques in many of the tested criteria and provides a robust, error-tolerant alternative to established augmented reality indoor navigation display techniques.

Keywords

Augmented Reality; Robust Indoor Navigation; Display Techniques; Perceived accuracy; Tracking errors;

1 INTRODUCTION

Augmented Reality (AR) device tracking systems are not perfect and errors can accumulate over time enforcing cognitive compensation of the user [MMC00]. In this work we compare the robustness of AR display techniques (interfaces) for indoor navigation to provide useful navigation information even in the presence of tracking errors.

AR provides location-aware user experience by overlaying spatially registered, digital information on a screen for real-time interaction with the physical and virtual environments [BCL15]. An important application scenario is pedestrian navigation. With recent advances in user tracking technologies and sufficient processing power of modern smartphones, the more challenging indoor navigation has become feasible [MKH⁺12].

However, the technology is still in its infancy and reliability for an extended amount of time has not been achieved [YNA⁺17, FPS⁺20]. McIntyre et al. stated that the problem of accumulation errors within a tracking system will not be solved in the near future [MCJ02]. And 20 years later he is still right. Relying on such a faulty system results in digital objects being far off their

supposed position. One way to tackle this problem is to increase the user's awareness of the tracking imperfections using different display techniques [PDCK13], e.g., by using 3D arrows that can change in color and shape, or similar feedback. Error visualization generally improves AR navigation systems but it is challenging to design suitable visualizations [PDCK13]. Even worse the system might not be aware of the tracking errors, lulling the user into false security.

In order to reduce the *impact* of tracking errors on indoor navigation instead of only making users aware of it, we contribute the following: We first define the problem of uncertain tracking errors in the context of tracking systems (section 2). Next, we classify AR navigation display techniques (section 3), and investigate typical representatives. We present objective criteria to evaluate AR display techniques with regard to error-robustness (section 4) and evaluate the representatives (section 5). In section 6, we propose our own display technique Bending Words and discuss (section 7) its advantages.

2 RELATED WORK

Pedestrian navigation services have gained attention for several years now [MRBT03, RC17]. They evolved from 2D paper maps to digital maps on mobile devices to location-based turn-by-turn instructions [Kim10]. Modern positioning systems enable AR systems to guide the user in real-time. These systems require a very high accuracy to correctly display information and avoid confusion and misguidance of the user. Yet, there is still no generally accepted solution for localization systems [MKH⁺12], e.g., Adler *et al.* who analyzed

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

and categorized 183 indoor localization techniques published between 2010 and 2014 [ASWK15].

A user generally expects accuracy to be perfect or at least good enough to rely on the displayed information which is often a mistake and may lead to navigation failures. Visualization techniques that adapt to the amount of tracking errors have the power to decrease the impact of inaccurate tracking solutions [PDCK13]. To understand why tracking solutions are always imperfect it is important to take a look at the underlying types and sources of error.

Localization

Localization is the process of tracking a user's position or more precisely the position of the device used for navigation. Applications for localization include pedestrian navigation, robotics, dynamic personalized pricing, product placement, advertisement, fleet management or intelligent spaces [YNA⁺17, LLY⁺15].

Localization systems rely on physical properties of signals, e.g. speed of light, or other measurable forces, such as earth's magnetic field. Sensor and information fusing may improve the overall performance, e.g. Wi-Fi and magnetic signals [SBS⁺15] or incorporating a priori knowledge, such as a map of the environment, to make the localization systems more robust and accurate [HFH04].

Despite constant improvements in localization techniques, a perfect solution seems almost impossible. Development times for indoor navigation systems are often several years and currently might not result in a widely accepted solution which has the high precision required to accurately display AR content within the camera feed [LLY⁺15, MSTSP⁺21, GFW21].

The cognitive load posed on the user of indoor navigation systems correlates with the accuracy of localization. This effect is especially prevalent in AR applications where the wrong positioning of visual media within the camera feed becomes distracting at best or misleading at worst [MCJ02].

Generally, two types of error exist: Rotational error, the angular difference between the direction to the assumed position of the next waypoint and the direction to the actual next waypoint; and translational error, the distance between the current position and the assumed position of the tracking device (Figure 1).

Tracking Systems

Different tracking systems have different sources and degrees of errors. Signal-based localization techniques (Wi-Fi, GPS, Bluetooth, etc.) are more prone to errors caused by changes in the environment. Cluttered

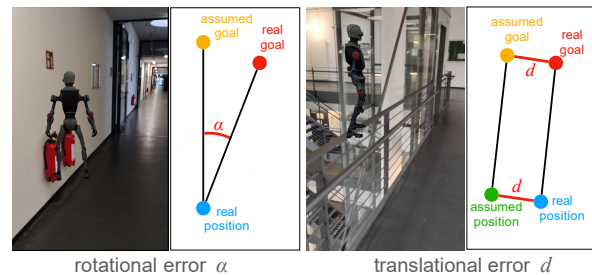


Figure 1: Types of error: **Rotational errors** (left) result in misleading direction information as the tracking system's **assumed goal** and the **real goal** differ. **Translational errors** (right) result in irritating positioning of the augmented content due to disparity between the **assumed position** and the **real position** as well as **assumed goal** and **real goal** of the tracking device.

Tracking System	Accuracy
Ultra-wideband Positioning [YNA ⁺ 17]	1 cm - 0.3 m
Wi-Fi Based Positioning [YS15]	1 - 10 m
Magnetic Positioning [SBS ⁺ 15]	1 - 8 m
Global Positioning System [LGD ⁺ 15]	1 - 10 m
Bluetooth [RLJ ⁺ 15]	0.5 - 10 m
Vision-Based Positioning [KJ08]	resolution dependent
Pedestrian Dead Reckoning [KH15]	distance dependent

Table 1: Accuracy of user tracking systems

environments introduce multipath, non-line-of-sight, and shadowing artifacts that affect either the arrival time, angle, or strength of a signal reaching the sensor [RC17, YNA⁺17]. Even the human body affects tracking accuracy [APM⁺16].

In Table 1 we provide an overview of several tracking techniques commonly found in pedestrian navigation systems and list typical accuracy ranges.

AR applications require a high precision tracking in order to display the virtual content correctly. This makes vision-based positioning systems currently the only alternative. These systems mostly rely on feature point tracking with one or more cameras and/or depth information [KJ08]. Due to their computational demands vision-based techniques are often coupled with faster techniques [BEP15]. While lighting conditions, surface properties (reflecting/refracting), and occlusion may negatively impact tracking precision [DRMS07], accuracy of vision-based positioning systems is theoretically only limited by image resolution.

Other exotic positioning systems utilize sound, light beacons, FM radio, or RADAR, to localize a user within a building but are rarely used in practice [YNA⁺17].

3 AR NAVIGATION DISPLAY TECHNIQUES

Commonly all AR indoor navigation display techniques require the following input:

1. The path from the current position towards the goal, usually provided as a list of 3D coordinates

2. The geometry of the building, including walls, doors, stairs, or elevators.
3. The user's pose that represents their position and rotation within the building.

Using this information the user gets feedback where to head next in order to reach their goal.

Besides the technical accuracy of these tracking systems, the way navigation information is presented has a strong impact on how accuracy of the navigation system is perceived by the user [PDCK13]. Misaligned or constantly jittering visual elements are distracting at best or misleading at worst [MCJ02]. Adapting how these instructions are delivered strongly increases performance of users when navigating through unknown environments [RC17]. But instead of interface design, the focus in indoor navigation applications is mostly put on their localization techniques.

The user's expectation of the system influences how they perceive it. If a technique is usually perceived as very accurate, based on past experiences, an inaccurate tracking will be overly distracting [MBMH01]. Whereas a less common technique which makes an imperfect tracking state less obvious could more easily prepare the user for the actual experience when using the system [MMC00].

Extending on Pankratz *et al.* [PDCK13] we distinguish three categories of AR navigation display techniques:

1. *Discrete information* which shows navigation hints as one or a series of next steps;
2. *Guiding information* which shows only the direction towards the next waypoint;
3. *Context information* which shows also the area around the user in an exocentric view.

Within these we found several representative techniques, see Table 2. Examples for the display techniques emphasized within the table are shown in Figure 2.

Discrete Information

Discrete Information display techniques provide information about the next steps along the path at any time. A common example is *Lines on the Ground* that lead towards the goal. Instead of providing a continuous path, *waypoint markers* display only the next corner. These techniques are prone to tracking errors as they do not provide much context information to allow the user to compensate for the error. A typical example are non-aligned waypoints due to rotational error. A line on the ground makes it sometimes hard to guess where the systems wants the user to go if the line is off it's intended direction (Figure 2, top left).

Discrete Information	Lines on the Ground Waypoint Marker Bending Words
Guiding Information	Guiding Arrow Shining Light Digital Avatar Haptic Feedback
Context Information	Paper Map World in Miniature

Table 2: Display technique examples. We evaluate the representatives most resilient against tracking errors (marked in bold) in section 5, and present Bending Words in (section 6).

Guiding Information

Guiding Information display techniques provide a series of guiding step-by-step information that are always limited to the next waypoint. This reduces cognitive load of the user and thereby positively affects their performance [WLPO94]. One contributing factor is the egocentric point of view that these display techniques provide [SCP95].

An example of such a technique is a *Guiding Arrow* that is positioned at the user and points towards the next waypoint [LMM16], or a *Shining Light* cone [MEN15]. This has the advantage of being perceived as less accurate than an arrow, which can dictate the users' expectations beforehand. The last approach in this category is the *Digital Avatar* that guides the user towards their goal by walking ahead of them [PDCK13].

A somewhat different approach to Guiding Information display techniques is the design of interfaces for visually impaired people [ZYZH19]. In this approach information is mapped to auditory or haptic cues (*Haptic Feedback*) to guide the user towards the next waypoint using different rates of tone frequencies or vibrations [ZYZH19].

Context Information

Context Information display techniques provide the user with additional information about their surroundings. This includes the traditional paper map. Users are shown the context in which they are positioned instead of direct navigation hints. By providing landmarks or other identifiable information about their surroundings, users are able to orient themselves using these techniques even without displaying their approximate location [BP13]. The goal of providing contextual information is to understand the connections between places within an environment and eventually help the user form their own route memory. This comes at the cost of an increased cognitive load while using the navigation system [RC17].

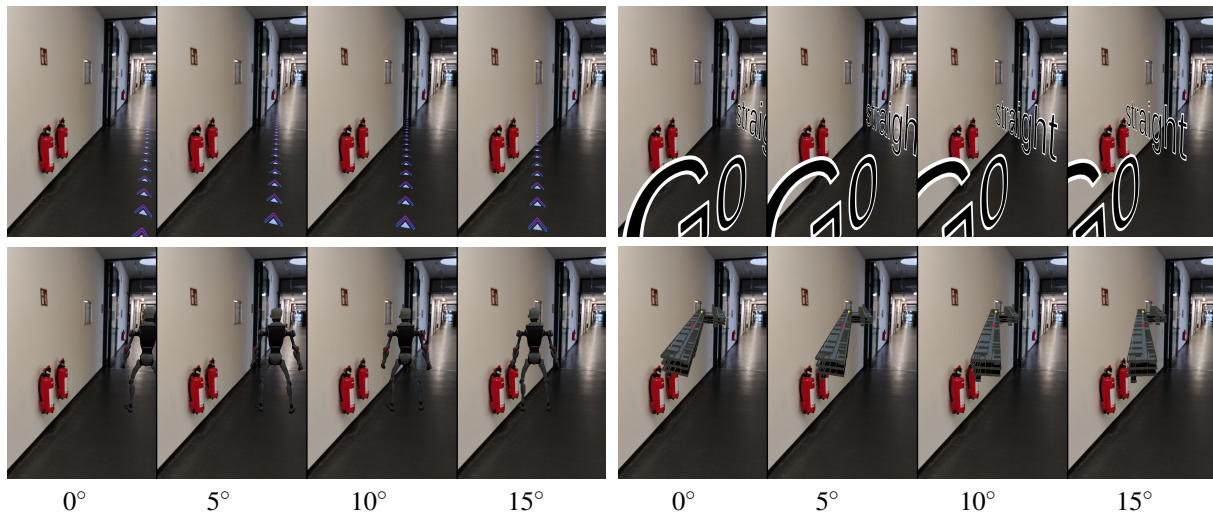


Figure 2: Simulated rotational error, increased in 5° steps from left to right for each display technique. Top left: Lines on the ground, top right: Bending Words (section 6), bottom left: Digital Avatar, bottom right: World in Miniature. As a non-visual technique, Haptic Feedback is omitted here.

A classic example is *World in Miniature (WIM)*. It essentially is a bird-eye view onto a small model of the building. Self-localizing is made easier by aligning the orientation of the miniature with the orientation of the real building and showing a marker for the position of the user within that model [HFH04, SCP95].

4 OBJECTIVE EVALUATION CRITERIA

In the following we describe how to evaluate navigation techniques based on objective criteria. We gathered these from multiple previous works to provide a comprehensive approach to rank display techniques within the context of creating error-robust indoor augmented reality navigation. They are not intended to replace systematic user studies but may be useful for preliminary investigations beforehand or to complement small-sized user studies in times where elaborate user studies are difficult to conduct, e.g. during a pandemic. The criteria are not metrics but instead provide guidelines to compare two or more display techniques against each other argumentatively. A score-based comparison was considered, as it can show variations between methods better, but due to the lack of calculable measures in some criteria it was decided against. Assigning each criterion a weight can be done and should be based on the individual requirements and target group.

We partition the aggregated list of criteria into two main categories: visibility (how to present information, subsection 4.1) and interaction (how to promote interaction, subsection 4.2). Each criterion within these categories was chosen based on the relevance to not only AR interfaces in general, but also to diminish the problem of uncertainty in tracking. Both technical aspects and human factors, including familiarity [AZLK12], are covered.

4.1 Visibility

Visibility describes elements that influence the ability of a user to perceive the navigation information and process it. This includes the visibility of elements, how often and how much of the information is located within the screen, and how effective it is in guiding the user.

Deviation Range: The deviation range refers to the range in which a display technique is still perceived as conveying the right information even though tracking errors exist [MKD⁺14]. The more precise the alignment between the desired position within the world and the position as shown on the display, the lower the chance of user failures such as taking a wrong turn or following a wrong path [MSS11]. It was shown that target detection performance decreases notably from precise (< 7.5° rotational error) to partially degraded (< 22.5°) to poor (< 45°) given the increase in error [YWMB01]. Möller et al. also showed that users "perceived [wrongly estimated orientation] more negatively than a wrongly estimated location" [MKH⁺12]. Thus, tolerance against angular tracking error is an important evaluation factor, which we call the Deviation Range criteria. The display techniques can be evaluated and compared using the above mentioned thresholds in a qualitative manner or by comparing the overlap between the displayed content and the next goal on the display.

Path Information Visibility: To help prevent navigation errors, instructions have to be comprehensively distributed across the path and must be visible at appropriate points in time [RC17, SK00]. An overview of the upcoming tasks and especially the ability to easily see the next target location can also increase the performance of users [MSS11]. To summarize these effects, we devised this criterion where one needs to compare

the amount of displayed path information for each technique.

4.2 Interaction

The way a display technique promotes interaction with the user, influences the user experience, but also impacts the tracking system itself. In the following we will explain the different criteria related to user interaction and their relevancy.

Device Orientation: The device orientation is mainly relevant for non-head-mounted vision-based tracking devices such as smartphones. The natural way for a user to hold these devices is at a 45° angle towards the ground [MKD⁺14]. This limits the number of feature points visible to the camera which negatively impacts the performance of a vision-based tracking system. A display technique should therefore enforce an upright position of the tracking device in a subtle and non-disturbing manner, which increases robustness.

Instant Feedback: Reaction time to changes in the tracking information of a display technique is an important factor for overall performance and user experience [RC17, SG04]. A technique, which updates the displayed information e.g. only at the next waypoint or at too large of a translational or rotational error can lead to deteriorated performance as the user might overshoot the target. Displaying lengthy animations are problematic for the same reasons.

Environmental Awareness: The awareness of the surrounding environment is an important factor when using digital navigation aids in general [BP13, MOP⁺09]. It decreases the requirement to use a navigation aid over time as the user becomes familiar with their surrounding and it also improves safety to avoid dangerous situations and obstacles. During times where the positioning system accumulates too much error to display reliable information, users can continue their navigation in the right direction using their acquired spatial knowledge [KAZ04]. A display technique can increase environmental awareness by using landmarks as part of the localization method or by making the environment stand out more. Techniques that cover too much of the screen or require the user to constantly look at it decrease environmental awareness. Environmental awareness using different display techniques is not measurable but needs to be compared argumentatively.

Multimodality Count: Multimodality Count is a measure counting the number of natural sensory receptors being utilized to convey information [GLB05]. In other words, how many senses does the display technique address? The main modalities relevant to current AR applications and the most researched within that context are: visual, auditory, and haptic modalities [Liv05, KSS20]. It is important to choose the right combination and number of modalities for the task at hand: Using more than

one creates a more natural interaction with the system that grants more flexibility in a mobile situation such as during navigation [Gri09]. It can also increase the application's accessibility by allowing a user to freely choose their preferred modality [KSS20]. And it can reduce navigation errors of users, especially in reduced visibility conditions or if one modality conveys ambiguous information, therefore increasing effectiveness of the user-computer system [CFBM13].

Familiarity: A familiar display technique can help users build trust in the navigation information [AZLK12]. Existing navigation applications have introduced a set of display techniques that are widely accepted and understood, such as arrows or lines. Using these known forms can help users understand the system's intention when being guided. New, unfamiliar, technologies can sometimes lead to inadequate user experience for users with no previous knowledge of it and increase the cognitive load [ASB18]. This criterion is not directly measurable and depends on previous user experiences, though we expect differences to be mostly cultural. Therefore, a ranking using the familiarity criterion must be based on argumentative comparisons.

5 CRITERIA APPLICATION

To apply the presented evaluation criteria, we chose at least one representative from each navigation visualization category. From the Discrete Information category we chose lines on the ground as they generally provide more information than waypoint markers. From the Guiding Information category we chose the Haptic Feedback [ZYZH19], as the only non-visual navigation technique; and the Digital Avatar [PDCK13], as Guiding Arrow and Shining Light are conceptually only specialized instantiations of the Digital Avatar. As Paper Map is a non-digital navigation technique we opted for World-in-Miniature [SCP95] from the Context Information category.

In the following, we briefly describe the implementations of the techniques. Lines on the ground are implemented as stripes of arrows, that lead from the user along the path (Figure 2, top left). The walking direction is indicated by the arrow directions. The entire path is visible as occlusions from the environment are not taken into account. World in Miniature (WIM) shows the surrounding area as a small model within the view (Figure 2, bottom right). The model, including the user's tracked position depicted as a red dot, is fixed in front of the user, while its orientation is constantly aligned with the tracked orientation of the real building for an improved user performance [WLPO94]. Besides model and user position, the only additional information is the very next step of the navigation path, displayed as a yellow dot. The Digital Avatar (Figure 2, bottom left), is a humanoid robot entity that guides the user towards

the next waypoint along the path. It walks in front of them and waits when the user is not moving. Haptic Feedback links angular difference between current viewing direction and next waypoint to vibration frequency of the tracking device. If the tracking device is pointed towards the correct direction (the next waypoint is inside the green region), it vibrates at the highest rate. If the next waypoint is in the outer thirds of the screen it vibrates with a medium frequency. If it is to the left or right of the screen the slowest vibration is applied to make the user aware that the system is still running.

5.1 Results

We obtain the **deviation range** ranking by measuring the range of rotational error that a display technique can be exposed to until it no longer overlaps with the position of the next waypoint. We verified the results qualitatively by simulating the rotational error in steps of 2.5° (Figure 2 showing every 2nd step).

Lines on the ground shows perceptible mismatch already at a small rotational error, the Digital Avatar and Haptic Feedback display accurate navigation information within a 5° rotational error. The value for Haptic Feedback has been obtained by measuring the range in which the vibration rates change. Besides the quantitative value of the deviation range we also include unique characteristics of the display techniques. E.g., even though the deviation range of WIM is only 5° it can still be used to identify the next steps by comparing the landmarks within the model with landmarks of the surroundings.

Concerning the **path information visibility** Haptic Feedback provides the fewest information as it only roughly points towards the next waypoint. Digital Avatar gives hints on where the waypoint is as a user approaches it. Because of the lack of occlusion, Lines on the ground can show more than just the path to the next waypoint, although information further down the path becomes more and more disassociated to the environment. WIM performs best, as it displays the complete environment as a map and can potentially show the complete path.

Most techniques that rely on displaying spatially registered information, such as lines on the ground and Digital Avatar, enforce an upright **device orientation** of 90° which is beneficial for feature tracking. With Lines on the ground the user has a slight tendency to look down and follow the arrows. Haptic Feedback is designed to aid visually impaired people who don't use visual cues and therefore tend to hold the device at a more natural angle of 45° . The camera then tracks mostly the floor which has fewer features. Linkage of the displayed model in WIM with the orientation of the tracking device [WLPO94] makes a rotation of smaller degrees more likely to improve visibility of path information, which has an inverse effect on the device orientation criterion. WIM also never gives incentives to point the camera in the direction of travel, making it rank worst.

Instant feedback All techniques except Digital Avatar update the navigation information or a deviation from the planned path in real-time. The Digital Avatar is restricted to human speed to not break the immersion. Note that this is strongly implementation dependant and should not be generalized.

Environmental awareness describes the trade-off between providing information from the navigation technique and environmental information to avoid obstacles and to become familiar with the environment over time. The WIM model shows the best support by providing context information in form of a model of the surroundings. Lines on the ground provides some but lesser context information in the form of directional changes at junctions and corners. Haptic Feedback was ranked third due to its minimum amount of information. We consider the digital avatar to be the least awareness-friendly technique, as it not only provides very little path information and occludes a large area of the screen, but also inherently enforces the user to focus on the avatar instead of the actual path.

The **multimodality count** refers to the number of natural senses addressed. All techniques solely rely on visual information except haptic feedback, which vibrates the device and even generates acoustic feedback thereby. Again, this is strongly implementation dependent.

The **familiarity** of each technique is based on how often elements of it are found within other everyday navigation situations. Digital Avatar implements the common situation where a user has to follow someone through an unknown environment. Lines on the ground in a similar form is broadly used for car and pedestrian navigation systems. Transferring this to AR preserves this familiarity. WIM resembles the well-known paper maps. Haptic Feedback is the least common approach as it makes the assumption that users can correlate an increase in vibration speed to positive directional feedback.

6 BENDING WORDS

Bending Words is our proposed discrete information display technique based on optimizing the criteria from section 4. In its core, this technique takes advantage of a person's ability to naturally follow turn-by-turn instructions. [Kim10] Yet, it overcomes the problem that turn-by-turn instructions are usually unable to show the exact location of the next turn [PB10]. During navigation *Bending Words* shows a three-dimensional text containing turn-by-turn instructions and adapts to the path in front of the user (Figure 2, top right, and Figure 3) both in terms of position and displayed text. The displayed text consists of two parts: The action keyword which is either *Go* or *Destination reached* to indicate if the goal is reached; and the direction keyword which is one of *right/left/straight/up/down* to indicate the next action at the next waypoint.



Figure 3: Bending Words: From left to right: The text aligns roughly with the path and the foreshortening as well as the content guide the user. If a new waypoint, e.g. corner, is approached the direction keyword changes and snaps to the waypoint's position. Once the waypoint is passed the technique returns to it's initial setup but always using the foreshortening effect to gently guide the user towards the next waypoint.

The displayed text is large enough to be easily readable but small enough to hide as little as possible of the environment. To improve readability and let the text stand from the environment, we opt for a black text with white contour.

The text aligns with the navigation path in front of the user. The action keyword *Go* is positioned at a small distance from the user and rotated around the y-axis to face the user at an angle. Due to the resulting foreshortening effect this indicates a walking direction for them. The absence of precise directions makes the user aware that this is only an approximate guidance. The direction keyword is placed at a larger distance and aligns with the path similarly as the action keyword. Its rotation is adjusted to keep a viewing angle that ensures readability. The angle adapts so that the foreshortening guides towards the next corner. When approaching a waypoint where the directional change would indicate an upcoming corner, the direction keyword snaps to the waypoint to emphasize this change before continuing along the path. An example is shown in Figure 3.

The advantage of this technique is that it shows users where to go and it tells them in text form, too. This supposedly increases the user's ability to understand the system's intention even if it suffers from low tracking accuracy. This way the user sees where they are headed to, based on the position of the words while the words themselves provide clear instructions in case of unclear situations.

7 DISCUSSION

In the following we discuss the proposed Bending Words display technique. Comparing it to the previous four display techniques, we obtain the ranking summarized in

	Deviation Range	Path Information Visibility	Device Orientation	Instant Feedback	Environmental Awareness	Multimodality Count	Familiarity	Average Rank
Lines on the ground	5	2	3	1	3	2	1	2.43
WIM	2	1	5	1	1	2	3	2.14
Digital Avatar	4	4	1	5	5	2	1	3.14
Haptic Feedback	3	5	3	1	4	1	5	3.14
Bending Words	1	3	1	1	2	2	3	1.86

Table 3: Ranking of the different display techniques (left) within the selected criteria (top), where lower numbers are better.

Table 3. A lower number means a better ranking in the respective category. The ranking of the baseline techniques has been discussed in section 5, here we only address the performance of Bending words in comparison to these techniques.

The weighting of the criteria should be adapted based on the goals, given environment or target group. For example when developing a solution for people with mild cognitive impairments, the Familiarity criterion could be weighed more than Path of Information Visibility. The exact weights would need to be determined using a user study. For this work, we weight all criteria equally. In many criteria, the proposed Bending Words outranks the other display techniques (Table 3), followed by WIM.

Deviation Range: As the words in Bending Words only hint towards the next waypoint the text supports a clear decision making, resulting in strong robustness even

with a rotational error of up to 10° ranking it the highest among the compared techniques. While the user can also correct any error when using WIM by comparing the map with their surroundings, this comes at a slightly stronger cognitive load though.

Strong translational and rotational errors can also have more drastic negative effects. The less dramatic effect would be that the system guides the user against a wall, in a more dramatic case, which we encountered during our own tests, Digital Avatar and Haptic Feedback would guide the user right down the stairs, even though the correct path continued to the right directly after the stairs. WIM can make it hard for the user to decide for the correct path, as the goal is displayed in the map but not necessarily the path. Even though the placement of Bending Words may also guide the user towards the stairs, the textual information always gives the user the possibility to correct the presented information. E.g. if Bending Words states "Go right" it is obvious that one should not go down the stairs as it would state "Go down" or even "Go down the stairs" instead.

Path Information Visibility: While Haptic Feedback and Digital Avatar only roughly indicate the current direction, Bending Words and Lines on the ground provide some contextual information (position of next waypoint, direction beyond that), and WIM even reveals the whole surrounding. We deem Bending Words to be slightly worse than Lines on the ground with this regard but better than Digital Avatar as it provides more information about the next waypoint earlier on.

Device Orientation: Similar to the Digital Avatar, Bending Words enforces the desired upright orientation of the tracking device. All other techniques implicitly enforce a worse orientation towards the ground.

Instant Feedback: Bending Words constantly reflects updates from the tracking system and quickly corrects known errors. Therefore, all techniques, except for Digital Avatar, perform equally well.

Environmental Awareness: An active process of localization supports the learning process of spatial knowledge required to independently navigate the building [Kui16]. Bending Words provides a neat way to give just enough guidance to stay on track, supporting memorization of the path. Therefore, we rank it slightly better than Lines on the ground. While we could give more precise context information, e.g. "Go right at elevator", this might clutter up screen space. Such extensions to our basic approach should be evaluated in the future.

Multimodality Count: As all techniques, except for Haptic Feedback, Bending Words mostly focus on the visual sense. It would be very simple to include additional senses, though, e.g. through audio feedback.

Familiarity: One can argue that we are used to textual instructions e.g. from assembly manuals or street

signs which resemble Bending Words. Though, we are probably more used to follow other persons, as with the digital avatar. However, few of us are used to interpret vibrations as an information channel.

Limitations: Within our study we did not investigate how display techniques could be combined to improve the shortcomings of each other. For example, the Digital Avatar could be combined with an indicator that tells the user where the avatar is currently waiting for them, or the non-visual Haptic Feedback technique could be combined with a visual technique for a similar effect.

Bending Words is also limited by its reliance on turn-by-turn instructions, which can have adverse effects on spatial learning [KAS]. Its utilization of 3D space imposes an additional constraint on the number of words that can be displayed, further limiting the amount of instructive information that can be conveyed.

8 CONCLUSION

This article has presented a set of seven objective evaluation criteria for error-robustness in AR indoor navigation. Based on these, we have introduced a new display technique called Bending Words that can reduce the impact of tracking errors within an AR navigation application. We have evaluated and compared it to four other baseline techniques. Bending Words ranks best within the criteria, closely followed by WIM. Bending Words expands the spatially registered information provided by an AR display with precise instructions that can be easily interpreted.

In the future, new display techniques could be constructed for each visualization category, using the criteria presented in this work. It would be interesting to see how these criteria and other human factors interact with each other.

9 ACKNOWLEDGMENTS

Partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 491805996 / GR 5932/1-1.

10 REFERENCES

- [APM⁺16] Askarzadeh, F., Pahlavan, K., Makarov, S., Ye, Y., and Khan, U. Analyzing the effect of human body and metallic objects for indoor geolocation. In *2016 10th Int. Symposium on Medical Information and Communication Technology (ISMICT)*, pages 1–5, 2016.
- [ASB18] Arifin, Y., Sastria, T. G., and Barlian, E. User experience metric for augmented reality application: A review. *Procedia Computer Science*, 135:648–656, 2018.

- [ASWK15] Adler, S., Schmitt, S., Wolter, K., and Kyas, M. A survey of experimental evaluation in indoor localization research. In *2015 Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10, 2015.
- [AZLK12] Arning, K., Ziefle, M., Li, M., and Kobbelt, L. Insights into user experiences and acceptance of mobile indoor navigation devices. In *Proc. of the 11th Int. Conf. on Mobile and Ubiquitous Multimedia*, pages 1–10. ACM, 2012.
- [BCL15] Billingham, M., Clark, A., and Lee, G. A survey of augmented reality. *Found. Trends Hum.-Comput. Interact.*, 8(2-3):73–272, 2015.
- [BEP15] Bettadapura, V., Essa, I., and Pantofaru, C. Egocentric Field-of-View Localization Using First-Person Point-of-View Devices. In *2015 IEEE Winter Conf. on Applications of Computer Vision*, pages 626–633, 2015.
- [BP13] Brown, M. and Pinchin, J. Exploring Human Factors in Indoor Navigation. In *The European Navigation Conf.*, page 7, 2013.
- [CFBM13] Calvo, A. A., Finomore, V. S., Burnett, G. M., and McNitt, T. C. Evaluation of a mobile application for multimodal land navigation. *Proc. of the Human Factors and Ergonomics Society Annual Meeting*, 57(1):1997–2001, 2013. Publisher: SAGE Publications Inc.
- [DRMS07] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.
- [FPS⁺20] Feigl, T., Porada, A., Steiner, S., Löffler, C., Mutschler, C., and Philippsen, M. Localization Limitations of ARCore, ARKit, and Hololens in Dynamic Large-scale Industry Environments. In *VISIGRAPP*, 2020.
- [GFW21] Gomes, A., Fernandes, K., and Wang, D. Surface prediction for spatial augmented reality applications. *Virtual Reality*, 25(3):761–771, 2021.
- [GLB05] Grasset, R., Looser, J., and Billingham, M. A step towards a multimodal AR interface: a new handheld device for 3D interaction. In *Fourth IEEE and ACM Int. Symposium on Mixed and Augmented Reality (ISMAR'05)*, pages 206–207. IEEE, 2005.
- [Gri09] Grifoni, P. *Multimodal Human Computer Interaction and Pervasive Services*. IGI Global, 2009.
- [HFH04] Hallaway, D., Feiner, S., and Höllerer, T. Bridging the Gaps: Hybrid Tracking for Adaptive Mobile Augmented Reality. *Applied Artificial Intelligence*, 18(6):477–500, 2004.
- [KAS] Krukar, J., Anacta, V. J., and Schwering, A. The effect of orientation instructions on the recall and reuse of route and survey elements in wayfinding descriptions. 68:101407.
- [KAZ04] Krüger, A., Aslan, I., and Zimmer, H. The effects of mobile pedestrian navigation systems on the concurrent acquisition of route and survey knowledge. In Brewster, S. and Dunlop, M., editors, *Mobile Human-Computer Interaction - MobileHCI 2004*, Lecture Notes in Computer Science, pages 446–450. Springer, 2004.
- [KH15] Kang, W. and Han, Y. SmartPDR: Smartphone-Based Pedestrian Dead Reckoning for Indoor Localization. *IEEE Sensors Journal*, 15(5):2906–2916, 2015.
- [Kim10] Kim, H. J. Turn-by-turn navigation system and next direction guidance method using the same. In *US Patent 7844394B2*, 2010.
- [KJ08] Kim, J. and Jun, H. Vision-based location positioning using augmented reality for indoor navigation. *IEEE Transactions on Consumer Electronics*, 54(3):954–962, 2008.
- [KSS20] Kuriakose, B., Shrestha, R., and Sandnes, F. E. Multimodal navigation systems for users with visual impairments—a review and analysis. *Multimodal Technologies and Interaction*, 4(4):73, 2020.
- [Kui16] Kuipers, B. *The "Map in the Head" Metaphor. Environment and Behavior*, 2016. Publisher: SAGE Publications.
- [LGD⁺15] Li, X., Ge, M., Dai, X., Ren, X., Fritsche, M., Wickert, J., and Schuh, H. Accuracy and reliability of multi-GNSS real-time precise positioning: GPS, GLONASS, BeiDou, and Galileo. *Journal of Geodesy*, 89(6):607–635, 2015.
- [Liv05] Livingston, M. Evaluating human factors in augmented reality systems. *IEEE Computer Graphics and Applications*, 25(6):6–9, 2005.
- [LLY⁺15] Lymberopoulos, D., Liu, J., Yang, X., Choudhury, R. R., Handziski, V., and Sen, S. A realistic evaluation and comparison of indoor location technologies: experiences and lessons learned. In *Proc. of the 14th Int. Conf. on Information Processing in Sensor Networks*, pages 178–189. Association for Computing Machinery, 2015.
- [LMM16] Liu, K., Motta, G., and Ma, T. XYZ Indoor Navigation through Augmented Reality: A Research in Progress. In *2016 IEEE Int. Conf. on Services Computing (SCC)*, pages 299–306, 2016.
- [MBMH01] MacIntyre, B., Bolter, J., Moreno, E., and Hannigan, B. Augmented reality as a new media experience. In *IEEE and ACM Int. Symposium on Augmented Reality*, pages 197–206, 2001.
- [MCJ02] MacIntyre, B., Coelho, E., and Julier, S. Estimating and adapting to registration errors in

- augmented reality systems. In *Proceedings IEEE Virtual Reality*, pages 73–80, 2002.
- [MEN15] Moura, D. and El-Nasr, M. S. Design techniques for planning navigational systems in 3-d video games. *Computers in Entertainment*, 12(2):2:1–2:25, 2015.
- [MKD⁺14] Möller, A., Kranz, M., Diewald, S., Roalter, L., Huitl, R., Stockinger, T., Koelle, M., and Lindemann, P. A. Experimental evaluation of user interfaces for visual indoor navigation. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 3607–3616. Association for Computing Machinery, 2014.
- [MKH⁺12] Möller, A., Kranz, M., Huitl, R., Diewald, S., and Roalter, L. A mobile indoor navigation system interface adapted to vision-based localization. In *Proc. of the 11th Int. Conf. on Mobile and Ubiquitous Multimedia*, pages 1–10. Association for Computing Machinery, 2012.
- [MMC00] MacIntyre, B. and Machado Coelho, E. Adapting to dynamic registration errors using level of error (LOE) filtering. In *IEEE and ACM Int. Symposium on Augmented Reality (ISAR 2000)*, pages 85–88, 2000.
- [MOP⁺09] Morrison, A., Oulasvirta, A., Peltonen, P., Lemmela, S., Jacucci, G., Reitmayr, G., Näsänen, J., and Juustila, A. Like bees around the hive: a comparative study of a mobile augmented reality map. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 1889–1898. Association for Computing Machinery, 2009.
- [MRBT03] May, A. J., Ross, T., Bayer, S. H., and Tarkiainen, M. J. Pedestrian navigation aids: information requirements and design implications. *Personal and Ubiquitous Computing*, 7(6):331–338, 2003.
- [MSS11] Mulloni, A., Seichter, H., and Schmalstieg, D. Handheld Augmented Reality Indoor Navigation with Activity-based Instructions. In *Proc. of the 13th Int. Conf. on Human Computer Interaction with Mobile Devices and Services*, pages 211–220. ACM, 2011.
- [MSTSP⁺21] Mendoza-Silva, G. M., Torres-Sospedra, J., Potorti, F., Moreira, A., Knauth, S., Berkvens, R., and Huerta, J. Beyond euclidean distance for error measurement in pedestrian indoor location. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2021.
- [PB10] Pielot, M. and Boll, S. "in fifty metres turn left": Why turn-by-turn instructions fail pedestrians. In *Proc. of Using Audio and Haptics for Delivering Spatial Information via Mobile Devices, Lisbon (Portugal)*, pages 1–3, 2010.
- [PDCK13] Pankratz, F., Dippon, A., Coskun, T., and Klinker, G. User awareness of tracking uncertainties in AR navigation scenarios. In *2013 IEEE Int. Symposium on Mixed and Augmented Reality (ISMAR)*, pages 285–286, 2013.
- [RC17] Rehman, U. and Cao, S. Augmented-Reality-Based Indoor Navigation: A Comparative Analysis of Handheld Devices Versus Google Glass. *IEEE Transactions on Human-Machine Systems*, 47(1):140–151, 2017.
- [RLJ⁺15] Rida, M. E., Liu, F., Jadi, Y., Algawhari, A. A. A., and Askourih, A. Indoor Location Position Based on Bluetooth Signal Strength. In *2015 2nd Int. Conf. on Information Science and Control Engineering*, pages 769–773, 2015.
- [SBS⁺15] Shu, Y., Bo, C., Shen, G., Zhao, C., Li, L., and Zhao, F. Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing. *IEEE Journal on Selected Areas in Communications*, 33(7):1443–1457, 2015.
- [SCP95] Stoakley, R., Conway, M. J., and Pausch, R. Virtual reality on a WIM: interactive worlds in miniature. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 265–272. ACM Press/Addison-Wesley Publishing Co., 1995.
- [SG04] Sutcliffe, A. and Gault, B. Heuristic evaluation of virtual reality applications. *Interacting with Computers*, 16(4):831–849, 2004.
- [SK00] Sutcliffe, A. G. and Kaur, K. D. Evaluating the usability of virtual reality user interfaces. *Behaviour & Information Technology*, 19(6):415–426, 2000.
- [WLPO94] Wickens, C. D., Liang, C.-C., Prevett, T., and Olmos, O. Egocentric and Exocentric Displays for Terminal Area Navigation. *Proc. of the Human Factors and Ergonomics Society Annual Meeting*, 38(1), 1994.
- [YNA⁺17] Yassin, A., Nasser, Y., Awad, M., Al-Dubai, A., Liu, R., Yuen, C., Raulefs, R., and Aboutanios, E. Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. *IEEE Communications Surveys Tutorials*, 19(2):1327–1346, 2017.
- [YS15] Yang, C. and Shao, H.-r. WiFi-based indoor positioning. *IEEE Communications Magazine*, 53(3):150–157, 2015.
- [YWMB01] Yeh, M., Wickens, C. D., Merlo, M. J. L., and Brandenburg, D. L. Head-Up vs. Head-Down: Effects of Precision on Cue Effectiveness and Display Signaling. *Proc. of the Human Factors and Ergonomics Society Annual Meeting*, 45(27):1886–1890, 2001.
- [ZYZH19] Zhang, X., Yao, X., Zhu, Y., and Hu, F. An ARCore Based User Centric Assistive Navigation System for Visually Impaired People. *Applied Sciences*, 9(5):989, 2019.

Visualization of deviations between different geometries using a multi-level voxel-based representation

Andreas Dietze
Fulda University of
Applied Sciences
Leipziger Straße 123
36037 Fulda, Germany
andreas.dietze@cs.hs-fulda.de

Paul Grimm
Darmstadt University of
Applied Sciences
Haardtring 100
64295 Darmstadt,
Germany
paul.grimm@h-da.de

Yvonne Jung
Darmstadt University of
Applied Sciences
Haardtring 100
64295 Darmstadt,
Germany
yvonne.jung@h-da.de

ABSTRACT

We present an approach for visualizing deviations between a 3d printed object and its digital twin. The corresponding 3d visualization for instance allows to highlight particularly critical sections that indicate high deviations along with corresponding annotations. Therefore, the 3d printing thus needs to be reconstructed in 3d, again. However, since the original 3d model that served as blueprint for the 3d printer typically differs topology-wise from the 3d reconstructed model, the corresponding geometries cannot simply be compared on a per-vertex basis. Thus, to be able to easily compare two topologically different geometries, we use a multi-level voxel-based representation for both data sets. Besides using different appearance properties to show deviations, a quantitative comparison of the voxel-sets based on statistical methods is added as input for the visualization. These methods are also compared to determine the best solution in terms of the shape differences and how the results differ, when comparing either voxelized volumes or hulls. The application VoxMesh integrates these concepts into an application and provides the possibility to save the results in form of voxel-sets, meshes and point clouds persistently, that can either be used by third party software or VoxMesh to efficiently reproduce and visualize the results of the shape analysis.

Keywords

3D Object Comparison, Difference Visualization, Shape Similarity, Digital Twin, Voxel-based Modeling

1 INTRODUCTION

The acquisition, analysis and processing of 3d data based on depth data is still a current research area, as examples in the fields of visual computing like digital construction monitoring [DGJ20] show. This correlates with the fact that due to advances in augmented reality (AR), 3d sensing, and 3d scanning, many new mobile devices, such as the Samsung S21 Ultra or Apple's iPad, have advanced technologies for acquiring 3d data integrated into their product lineup. In terms of mobile devices, the technologies used (e.g., SfM, ToF, LiDAR) primarily serve to enrich a real scene with digital content, but are also used in the field of 3d reconstruction, as can be seen in the example of Microsoft's HoloLens mixed reality headset or Apple's LiDAR sensor, which are used, for example, in the context of digital construction monitoring and surveying [WWWH21, DGJ21].

In addition, more and more affordable and professional devices or systems in the field of 3d reconstruction are appearing on the market (e.g., from Shining 3D), which are suitable for high-resolution and detailed 3d reconstruction of smaller parts (e.g., gear wheel, EinScan - SP1¹) to medium sized objects (e.g., car door, EinScan HX2²). The resulting 3d data by such scanners, for example, can be used for a comparison between the 3d geometry of the planning data and a 3d reconstruction of a printed object (digital twin) in terms of their shape similarity. This allows the localization and measurement of deviations between these two geometries and could be used as a non-destructive testing method [WZL⁺20] in the field of additive manufacturing processes like 3d printing.

One of the problems that impacts the overall quality or functionality of the printed object is the 3d print warping of individual areas. These deformations are caused by the material shrinkage due to heating (expansion) and cooling (contraction) of the material. Another source of deformations is the use of support mate-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

¹ Shining 3D EinScan-SE: <https://www.einscan.com/desktop-3d-scanners-de/einscan-sp/>

² Shining 3D EinScan HX: <https://www.einscan.com/multifunktionaler-3d-scanner/einscan-hx/>

rial that can be wrongly placed or even missing, which obviously negatively influences the final shape of the printed object.

The main contribution of this paper is a novel method for visualizing and highlighting deviations between 3d printed products and their 3d reconstruction based on a multi-level voxelization (MLV) of topologically different geometries. This approach utilizes volumetric mesh voxelizations (including interior voxels) as well as voxelized mesh hulls (lacking interior voxels), where the latter allows higher grid resolutions. The MLV also allows determining a quality measure with statistical methods. The measurement of found deviations allows to classify if deviations are within a given tolerance, while the determined value of the shape similarity reflects an overall status of the deviations and could be used as a threshold value.

The benefit is to visualize differences in an early stage between the 3d planning data and its 3d reconstruction of the printed object to avoid follow-up costs in mass production or special spare parts. Visualizing deviations and critical sections like missing parts or strong deformations can help in terms of quality assurance as a non-destructive testing method, but also can help to adjust the printing settings (e.g., missing support material or layer height) and to ensure the correctness of printed spare parts.

2 RELATED WORK

For visualizing deviations between topologically different geometries via voxel-based representations, related work in the fields of object registration, voxelization, shape analysis and geometric similarity have to be considered. Efficiently highlighting the results of shape analysis is a broad field by itself and several approaches and tools already have been developed (cp. e.g. [OGBS06]).

In Novotni et al. [MR01] a method is described, where objects are first superimposed and aligned based on their center of mass and the eigenvectors before a similarity of the 3d objects is determined on the basis of their geometric properties (geometric similarity). The determination of geometric similarity was realized via distance fields to calculate offset-hulls, which provide information about overlapping areas of two volumes and are illustrated in the form of distance histograms.

Another method for measuring the similarity of 3d models is described in Chen et al. [CHL⁺17], in which a similarity measurement is performed on the basis of skeleton trees. For this purpose, the skeleton trees are created based on the topology of the 3d model's skeleton, so that the topological and geometric properties of the 3d models are represented and compared using the tree structure.

Furthermore, in Doboš et al. [DFFW18] a method is presented that detects differences between 3d models in the screen space and visualizes them for the user. To detect differences, various data such as color, depth, normals and texture coordinates are compared within screen space.

A method with which 3d planning data for an additive manufacturing process is analyzed for its geometric and mechanical properties prior to the manufacturing process is presented in Rupal et al. [RMWQ19]. This is achieved by converting the sliced print data of the 3d model back into a CAD model in a reverse process to enable optimization of the print settings.

Furthermore, in the past years neural networks are increasingly used for a geometric comparison of 3d models, even if they are mostly used for object recognition and classification in the sense of object similarity [KWL20, NZL⁺20, LEAM⁺19]. A comparison of CAD planning data and its components using machine learning is described in Bickel et al. [BSSW21]. Here, the 3d planning data for new components are compared with the 3d planning data for existing components.

3 CONCEPT

Our proposed concept for visualizing deviations based on shape analysis and the comparison between the original 3d planning data and the 3d reconstruction of the 3d printed output is based on the following steps:

1. Data acquisition: use 3d planning data to print objects and scan the printed output to provide a mesh-based 3d reconstruction
2. Shape analysis: provide similarity indices and voxel sets for visualization
 - (a) Pre-processing: positioning (superimposition) and alignment of the acquired data sets
 - (b) Multi-level voxelization (MLV): both data sets are voxelized separately and written to a file
 - (c) Shape similarity: compute similarity based on common statistical methods of the resulting voxel sets
3. Visualization: found deviations are visualized using a voxel-based or mesh-based representation

Figure 1 contains the pipeline of the whole process to summarize the complete procedure described in the next chapters.

4 DATA ACQUISITION

To provide data sets used as input for our prototypical implementation of the concept, we printed and reconstructed four objects (teapot, cat, frog, gearwheel). In

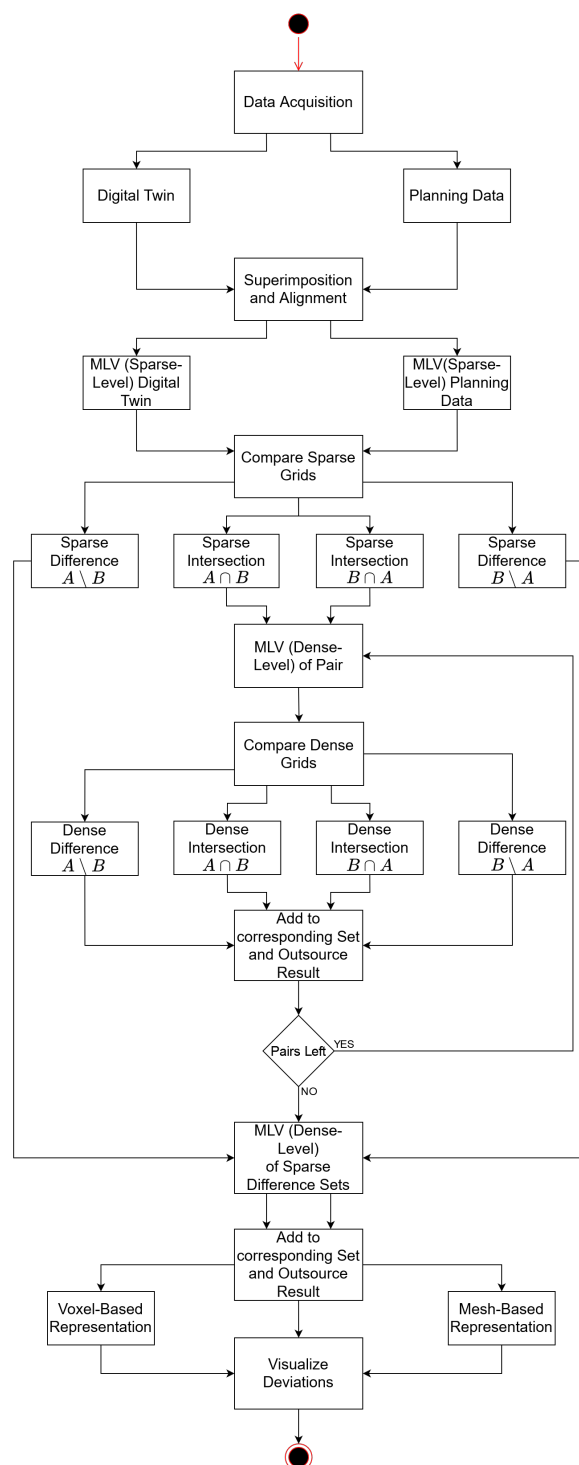


Figure 1: Illustration of the system pipeline.

addition, for one object (teapot) also manipulated planning data was used as reconstructed data, which made it easier to intentionally insert errors for testing.

As 3d printer, a Creality CR-10 V3 was used. The printer filament consisted of (green) PLA (Polylactic Acid), which is a common and widespread material

used for 3d printing. The objects were printed with a layer height of 0.28mm at an extruder temperature of 210° and a print bed temperature of 50°.

A model-based 3d reconstruction of the printed objects was carried out by a Shining 3D EinScan SE scanner and the included Software EXScan³ V.3.0.0 without any further mesh optimizations. This reconstruction represents the digital twin of the constructed planning data that was used for the comparison with the 3d planning data. Figure 2 contains the 3d planning data mesh as well as the corresponding 3d reconstruction from the printed object. Both, the planning data and the reconstruction differ in number of geometric primitives and their topology. The 3d reconstruction shown on the left consists of 93,627 vertices and 97,824 triangles. The planning data on the right consists of 10,206 vertices and 6,320 triangles.

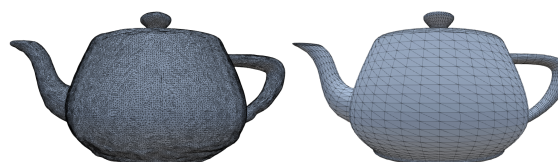


Figure 2: Topologically different data sets. Left: 3d reconstruction (93,627 vertices, 97,824 triangles). Right: original 3d planning data (10,206 vertices, 6,320 triangles).

5 SHAPE ANALYSIS

Pre-processing

For our proposed multi-level voxelization and shape analysis, an initial overlay regarding the position and orientation of the meshes to be compared is necessary, since the shape similarity is based on a statistical comparison of the different voxel sets resulting from the voxelization process (superimpositions and deviations, or intersection and difference sets). If the two objects are not initially superimposed and aligned, the objects are geometrically registered with the help of a semiautomatic approach using a principal component analysis (PCA) based on their center of mass and aligned using their eigenvectors. In the majority of our test cases, manual adjustment of the overlay and alignment (see Figure 3) was required following the application of the PCA due to differences in topology. As the focus of this paper is on comparison rather than alignment, this is sufficient but could be automated in the future.

Multi-Level Voxelization

The voxelization of the 3d planning data and the 3d reconstruction is performed in multiple voxelization steps

³ EXScan: <https://www.einscan.com/einscan-software/exscan-pro-software-download/>

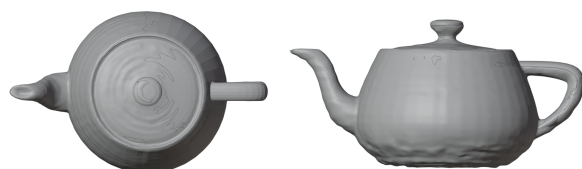


Figure 3: Superimposition and alignment of the planning data and 3d reconstruction of the printed object.

that differ in their implementation concerning the intersection of the voxel with the meshes. This is in relation with the voxelized results, that either can consist of voxelized hulls or volumes of the geometry. Regardless of the chosen voxelization result, the same input data provided by the Shining 3D EinScan SE is used.

In both cases, the first level consists of a sparse voxelization of the mesh hulls. This, on the one hand, is used as a space partitioning method to divide a predefined local space (e.g., 10cm x 5cm x 10cm) for each of the two meshes to be compared, resulting in a sparse 3d voxel grid for each mesh, in which a voxel either overlaps with the mesh or not. On the other hand, a comparison between the resulting voxel sets provides first information about critical sections, since there are deviations detected even at a sparse voxel resolution in the first level voxelization.

To determine the voxelized hull by an intersection of a voxel and the mesh, Unity's physics engine⁴ was used. In order to provide the voxelized volume, the intersection logic is performed via raycasting, where six rays starting from the center of a voxel check whether they collide with a surface of the mesh or not. The maximum number of voxels $v \in \mathbb{N}$ results from the length a_1 , height a_2 , and width a_3 of the local space to be measured and the parameterized voxel size $s \in \mathbb{R}^+$, which divides $a_1 \cdot a_2 \cdot a_3$. In Figure 4 a first level voxelization of the 3d reconstruction is shown. On the left, the voxelization consists only of the object's hull. On the right, the result consists of the voxelized hull (gray) and the volume (blue) of the mesh.

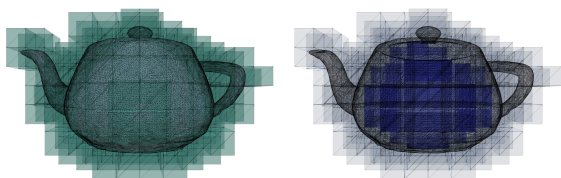


Figure 4: First level voxelization with a sparse grid resolution of 0.5cm of the 3d reconstruction. Left, voxelization of the mesh hull. Right, voxelization of hull and interior of mesh.

At the second level voxelization, the procedure is repeated for each sparse voxel cell. In case of the hull

voxelization, every voxel is divided into a dense 3d voxel grid and again, an intersection with the mesh and the voxels is performed with Unity's integrated physics engine that provides a high resolution voxelization, depending on the used dense voxel size. For the voxelized volumes, the intersection is performed using raycasting. Due to the cubic complexity of the voxelization process, the number of voxels, the runtime of voxelization and comparison, the memory consumption and a subsequent real-time visualization depend on the voxel sizes used for both (sparse and dense) 3d voxel grids and increase with higher accuracy (higher resolution due to smaller voxel size).

In the worst case, the number of voxels to be processed is larger than the available main memory. In terms of memory and runtime, the second level voxelization benefits from the first level voxelization, since here only the voxels resulting from the first level are voxelized in the dense resolution. In addition, each sparse voxel of the planning data result is compared one by one with the corresponding voxel of the 3d reconstruction's voxelization result to limit the memory consumption to those two cells instead of voxelizing all sparse cells at once.

In detail, the corresponding sparse cells are first voxelized using the dense 3d grid and either the physics or raycast procedure for the voxel-mesh intersection, which either results in a subset of the hull or volume of the meshpart enclosed by the sparse voxel. In the next step, both sparse voxels containing the second level voxelization result of those two cells are centered and each dense voxels are compared using the L2-norm detecting their euclidian distances. This determines which dense voxels intersect with both the planning data mesh and the reconstructed mesh (intersection set) and those that do not (difference sets).

Following this, the result in form of the determined voxel sets (intersection set and or difference sets) are written into a file and the used memory will be freed for the next sparse voxel pair. Sparse voxel cells that are either not overlapping with the planning data or 3d reconstruction are voxelized in the final step for each sparse voxel set individually and the second level voxelization result is added to the corresponding dense set (see Figure 5) and also written to the file, which, in addition, provides a persistent result of the whole process.

Shape Similarity

The determination of the shape similarity is based on the voxel sets resulting from the MLV. For the 3d planning data they consist of two voxel sets, one resulting from the sparse first level voxelization A_s , and the other dense voxel set A_d , resulting from the second level voxelization. Equivalent to this, the voxelization of the reconstructed results in B_s for the sparse set and B_d for the dense set.

⁴ <https://docs.unity.com>

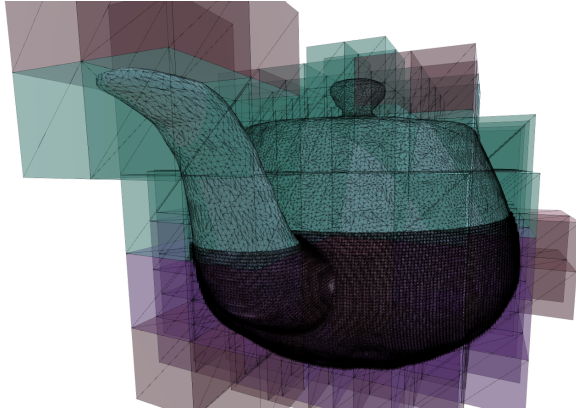


Figure 5: Visualization of the MLV process for the teapot's digital twin using a sparse resolution of 0.5cm and a dense resolution of 0.25mm. Green voxels are superimposed both with the meshes of the 3d reconstruction and the planning data respectively (intersection); red voxels do not overlap (difference set) and violet voxels have already been transferred to the dense voxel grid and compared in pairs. In a final step, the difference sets are voxelized.

A comparison between the two sparse voxel sets A_s and B_s is based on an overlay resulting in an intersection set $A_s \cap B_s$ that ideally corresponds to the union $A_s \cup B_s$ with a maximum degree of shape similarity, which also implies that the number of voxels in A_s equals the number of voxels in B_s . If the sets A_s and B_s have a different number of voxels, this results in the two difference sets $A_s \setminus B_s$ and $B_s \setminus A_s$. Moreover, this already provides information about critical sections since there are deviations detected between both meshes at the first level voxelization even without further approximation.

During the analysis of an overlapping voxel pair of $A_s \cap B_s$, all containing voxels of the two voxel sets A_d and B_d are checked for overlapping, resulting in the intersection $A_d \cap B_d$ and the two difference sets $A_d \setminus B_d$ and $B_d \setminus A_d$. Based on the quantity of those dense voxel sets, a shape similarity is determined by the use of established statistical methods providing a similarity coefficient using the Dice Index (DI) [ZWB⁺04], the Jaccard Index (JI) [FI18], and the Kulczynski Index (KI) [ZAB⁺16], where every similarity coefficient is in the interval [0, 1], while a higher value expresses a higher similarity.

The DI results of twice the quantity of the intersection set $|A_d \cap B_d|$ and the two difference sets $|A_d \setminus B_d|$ and $|B_d \setminus A_d|$.

$$DI = \frac{2|A_d \cap B_d|}{2|A_d \cap B_d| + |A_d \setminus B_d| + |B_d \setminus A_d|} = \frac{2|A_d \cap B_d|}{|A_d| + |B_d|} \quad (1)$$

The JI describes the cardinality of the intersection set $|A_d \cap B_d|$ and the union set $|A_d \cup B_d|$.

$$JI = \frac{|A_d \cap B_d|}{|A_d| + |B_d| - |A_d \cap B_d|} = \frac{|A_d \cap B_d|}{|A_d \cup B_d|} \quad (2)$$

The KI is defined by the intersection set $|A_d \cap B_d|$ and the two difference sets $|A_d \setminus B_d|$ and $|B_d \setminus A_d|$.

$$KI = \frac{1}{2} \left(\frac{|A_d \cap B_d|}{|A_d \cap B_d| + |A_d \setminus B_d|} + \frac{|A_d \cap B_d|}{|A_d \cap B_d| + |B_d \setminus A_d|} \right) \quad (3)$$

Results and Evaluation

All object pairs consisting of the planning data and the reconstructed meshes to be compared are located in two predefined local spaces (e.g., 10cm x 5cm x 10cm), which both are divided into a sparse and dense voxel grid by the MLV using the initially defined sparse and dense voxel size. Table 1 contains an overview of the voxel sizes used and the resulting resolution of the sparse and dense grid. Especially the size of a dense voxel is relevant here, since it corresponds to the unit size of a voxel in the voxelized meshes.

Table 2 contains the result of the comparison based on the voxel sets (hulls) of the 3d reconstruction (A) and 3d planning data (B). This consists of the determined number of voxels per mesh, the determined intersection, differences and union sets as well as the found shape similarity, based on the aforementioned Equations 1 (DI), 2 (JI), and 3 (KI).

It is also worth mentioning, that the first teapot in the table represents the shape analysis between the manually deformed digital twin (teapot without spout and handle, see Figure 7) and its planning data, whereas the second represents the 3d reconstruction of the printed teapot. The other three objects (cat, frog and gearwheel) are representing additional results from the shape analysis based on the MLV presented in Chapter 5.

Table 3 contains all objects using a volumetric MLV compared to the objects in Table 2, in which the voxelization results consists of the voxelized hulls. Noticeable is the similarity of the results between the DI and KI. Compared to the DI and KI, the JI results in a slightly smaller similarity coefficient. The voxelization of the planning data in the respective resolution is regarded as ground truth. A comparison of this data set with itself results in a shape similarity index of 1.0 concerning all statistical methods used, regardless of the chosen grid settings. Consequently, no deviations were identified either.

Comparing the similarity coefficients of Table 2 (voxelized hulls) and Table 3 (volumetric voxelization), it is noticeable that the results based on the hull generally give a lower value compared to the results based

on the volumes. In addition, the results are lower applying a higher resolution using the hulls compared to the volumes, where the results only slightly differ using different resolutions.

As test system, a Ryzen 9 3900X CPU with 32 GB DDR-IV, an NVidia RTX 3080 GPU on a 970 Evo Plus M.2 SSD, was used. Table 4 shows the required run-times for the voxelization process including the determination of the shape similarity for the hulls and volumes of the teapot (second object in Table 2 and first object in Table 3).

6 DEVIATION VISUALIZATION

The visualization of the detected deviations is based on the different voxel sets resulting from the MLV. These are either visualized directly (voxel-based representation) or the localized deviations are transferred to the input meshes (mesh-based representation). For both representations the following simple color scheme exemplarily is used:

BLUE: intersection $A_d \cap B_d$

RED: difference $A_d \setminus B_d$

YELLOW: difference $B_d \setminus A_d$

Voxel-based Representation

The voxel-based representation of all computed deviations is based on the intersection set and both disjoint difference sets between the voxel sets A_d and B_d , which can be individually visualized. In Figure 6, the incorrect teapot on the left is missing its spout and handle but also has a larger knob on the cover. It is representing the 3d reconstruction data that has been manually modified to provide some obvious deviations. The teapot on the right represents the 3d planning data. As can be seen, too, both meshes differ in their topology to simulate a mesh-based 3d reconstruction.

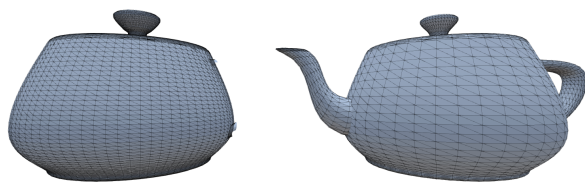


Figure 6: Left: 3d reconstruction with obvious deviations. Right: 3d planning data.

The found deviations are visualized individually on top of their corresponding mesh in Figure 7. On the left, deviations concerning the larger knob of the 3d reconstruction are shown, which represent an error that is located outside of the 3d planning data. In terms of the planning data, errors are located at the spout and handle but also concerning the smaller knob, as can be seen on the right. In the middle, the intersection set is shown.

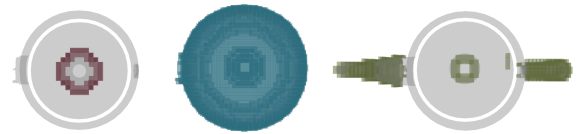


Figure 7: Left: difference set $A_d \setminus B_d$ (red). Center: intersection set $A_d \cap B_d$ (blue). Right: difference set $B_d \setminus A_d$ (yellow). The difference sets $A_d \setminus B_d$ and $B_d \setminus A_d$ are superimposed with their corresponding mesh.

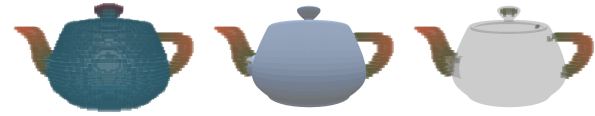


Figure 8: On the left, the union set $A_d \cup B_d$ using the color gradient in terms of the difference set $B_d \setminus A_d$ is visualized. In the center, only the difference set $B_d \setminus A_d$ is drawn on top of the 3d reconstruction. An overlay of the transparent digital twin and the difference set $B_d \setminus A_d$ to identify deviations inside the mesh is shown on the right.

A visualization using the voxel-based representation and a color gradient indicating the strength of the deviation is shown in Figure 8. On the left, the union $A_d \cup B_d$ is shown, which consists of both difference sets $A_d \setminus B_d$, $B_d \setminus A_d$ and the intersection $A_d \cap B_d$. In the middle, the difference set $B_d \setminus A_d$ is rendered on top with the mesh used as digital twin. Here, the voxels are representing the missing spout and handle. An overlay of the difference $B_d \setminus A_d$ on top of the transparent 3d reconstruction is shown on the right to identify deviations of the mesh.

The voxel color gradient from yellow to red indicates the strength of the found deviation and is realized by determining the distances between a dense voxel and the surface of the corresponding mesh, where the deviations are detected. This is accomplished by using the normal of the nearest vertex to that voxel as the direction for a ray starting from the position of the voxel to determine the intersection with the surface and thus the distance between the voxel and the mesh. If the ray does not intersect a surface, the distance between the voxel and the nearest vertex is used instead. This allows using a threshold for the detected deviations to determine at which distance the maximum color intensity for deviations is used (e.g., all deviations greater than 2mm are colored with a maximum color intensity).

The results of this representation based on the 3d reconstruction of the printed teapot using a dense voxel size of 1mm are shown in Figure 9. The reconstructed teapot consists of 97,824 triangles compared to the planning data with only 6,320 triangles. In this example, the deviations detected relatively to the planning data ($B_d \setminus A_d$) are superimposed with the reconstructed mesh (left) and vice versa. The middle contains the intersection set. The objects rendered on the top are using an orthographic projection and the objects at the bottom

Run	Sparse			Dense		
	Resolution	Size	Count	Resolution	Size	Count
1	10x5x10 voxel	1.0cm	500 voxel	5x5x5 voxel	2mm	125 voxel
2	20x10x20 voxel	0.5cm	4.000 voxel	5x5x5 voxel	1mm	125 voxel
3	20x10x20 voxel	0.5cm	4.000 voxel	10x10x10 voxel	0.5mm	1.000 voxel
4	40x20x40 voxel	0.25cm	32.000 voxel	10x10x10 voxel	0.25mm	1.000 voxel
5	40x20x40 voxel	0.25cm	32.000 voxel	20x20x20 voxel	0.125mm	8.000 voxel

Table 1: Resolution, voxel size and voxel count of the sparse and dense voxel grids.

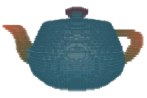
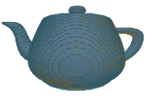
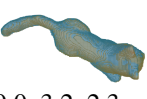


Object	R	A	B	A ∩ B	A \ B	B \ A	A ∪ B	DI	JI	KI
 3.0x1.5x1.9cm	1	371	408	355	16	37	408	0.930	0.870	0.931
	2	1,454	1,618	1,430	24	188	1,642	0.931	0.870	0.934
	3	5,794	6,537	5,706	88	831	6,625	0.925	0.861	0.929
	4	23,817	26,842	23,221	596	3,621	27,438	0.917	0.846	0.920
	5	95,800	108,047	92,712	3,088	15,335	111,135	0.909	0.834	0.912
 6.4x3.2x4.0cm	1	1,873	1,787	1,544	329	240	2,113	0.845	0.731	0.845
	2	7,520	7,264	5,701	1,819	1,563	9,083	0.771	0.627	0.771
	3	30,442	29,748	18,314	12,128	11,434	41,876	0.608	0.437	0.608
	4	120,966	119,610	47,401	73,565	72,209	193,175	0.394	0.245	0.394
	5	485,307	481,469	105,613	379,694	375,856	861,163	0.218	0.122	0.218
 9.0x3.2x2.3cm	1	1,653	1,692	1,525	128	167	1,820	0.911	0.838	0.911
	2	6,775	6,883	5,291	1,484	1,592	8,367	0.775	0.632	0.775
	3	27,400	27,371	15,668	11,732	11,703	39,104	0.572	0.401	0.572
	4	110,054	108,950	35,577	74,477	73,373	183,427	0.324	0.193	0.324
	5	440,181	435,901	73,386	366,795	362,515	802,696	0.167	0.091	0.167
 3.8x5.5x3.0cm	1	1,752	1,783	1,604	148	179	1,931	0.907	0.834	0.908
	2	7,059	7,287	5,919	1,140	1,368	8,427	0.825	0.702	0.825
	3	28,353	29,695	19,426	8,927	10,269	38,622	0.669	0.503	0.670
	4	113,593	119,734	48,181	65,412	71,553	185,146	0.413	0.260	0.413
	5	454,733	379,184	99,767	354,966	379,184	833,917	0.213	0.119	0.213
 3.8x0.7x3.8cm	1	558	399	388	170	11	569	0.810	0.682	0.834
	2	2,381	2,628	2,093	288	535	2,916	0.835	0.717	0.837
	3	9,692	13,472	8,715	977	4,757	14,449	0.752	0.603	0.773
	4	38,459	28,276	13,736	24,723	14,540	52,999	0.412	0.259	0.421
	5	154,927	145,666	33,655	121,272	112,011	266,938	0.223	0.126	0.224

Table 2: Results of the shape analysis based on the different voxel sets (hulls) to define the similarity $DI, JI, KI \in [0, 1]$.

are rendered with a perspective projection. Figure 10 shows a top and side view of the superimposed difference set $B_d \setminus A_d$ and the reconstructed mesh using the color gradient.

The difference sets $A_d \setminus B_d$ and $B_d \setminus A_d$ resulting at a dense voxel size of 0.25mm using the color gradi-

ent are shown in Figure 11. This result represents the fourth run of the second object in Table 3, where $A_d \setminus B_d$ consists of 73,565 voxel and $B_d \setminus A_d$ consists of voxel 72,209. Analyzing both difference sets with the use of the raycast method described in Chapter 5 allows an identification, whether a voxel is located inside or outside of the mesh. This means that the deviation associ-






Object	R	A	B	$ A \cap B $	$ A \setminus B $	$ B \setminus A $	$ A \cup B $	DI	JI	KI
	1	328	397	324	4	73	401	0.893	0.807	0.901
	2	2,570	2,651	2,570	0	81	2,651	0.984	0.969	0.984
	3	19,842	20,375	19,786	56	589	20,431	0.983	0.968	0.984
	4	139,910	144,518	139,398	512	5,120	145,030	0.980	0.961	0.980
	1	3,383	2,960	2,951	432	9	3,392	0.930	0.869	0.934
	2	27,286	24,366	24,306	2,980	60	27,346	0.941	0.888	0.944
	3	217,708	193,982	193,309	24,399	673	218,381	0.939	0.885	0.942
	4	1,741,196	1,553,114	1,547,720	193,476	5,394	1,746,590	0.939	0.886	0.942
	1	1,998	1,900	1,820	178	80	2,078	0.933	0.875	0.934
	2	15,564	15,112	14,518	1,046	594	16,158	0.946	0.898	0.946
	3	124,049	121,030	116,125	7,924	4,905	128,954	0.947	0.901	0.947
	4	992,360	968,189	928,345	64,015	39,844	1,032,204	0.947	0.899	0.947
	1	2,000	1,933	1,885	115	48	2,048	0.958	0.920	0.958
	2	16,356	15,763	15,492	864	271	16,627	0.965	0.931	0.965
	3	130,248	125,852	122,836	7,414	3,016	133,264	0.959	0.921	0.959
	4	1,041,852	1,001,092	982,146	59,706	18,946	1,060,798	0.961	0.925	0.961
	1	369	261	256	113	5	387	0.812	0.684	0.837
	2	2,779	2,660	2,633	146	27	2,806	0.968	0.938	0.968
	3	22,115	21,240	21,006	1,109	234	22,349	0.969	0.939	0.969
	4	178,507	169,520	166,079	12,428	3,441	181,948	0.954	0.912	0.955

Table 3: Results of the shape analysis based on the different voxel sets (volumetric) to define the similarity $DI, JI, KI \in [0, 1]$.

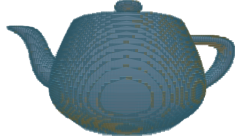
Object	Run	Hull	Volume
	1	00:03	00:02
	2	00:13	00:13
	3	01:02	01:43
	4	04:54	15:38

Table 4: Runtime in mm:ss format from run 1 to 4 for the voxelization process of the teapot resulting in voxelized hulls or volumes. The results reflect the processing time for the second object of Table 2 (hull) compared to the first object of Table 3 (volume).

ated to this voxel relates to the inside or outside of the mesh. Figure 12 illustrates the context. In the top row, first the difference set $A_d \setminus B_d$ is visualized. Here, the red voxels are representing outliers while yellow ones are representing inliers. Consequently, this has to be inverted for the differences set $B_d \setminus A_d$, as shown in the top right. The intersection set is shown in the middle. In the bottom row, the dense set A_d (left) and dense set B_d (right) is shown. In addition, the color gradient is used from red to magenta for outliers and orange to yellow for inliers to indicate the strength of the deviation.



Figure 9: In the top row contains an overlay of the difference set $B_d \setminus A_d$ (yellow), the intersection set $A_d \cap B_d$ (blue) and the difference set $A_d \setminus B_d$ (red) with transparent digital twin from above using a orthographic projection. In the bottom row, the same sets are rendered on top of the transparent 3d object from the side using a perspective projection.

Mesh-based Representation

Transferring detected deviations from the voxelization to its corresponding mesh is done via vertex colors that are associated to the voxel, in which a given vertex is inside. Since the elements of the sparse voxel set $A_s \cap B_s$ are compared in pairs to enable a more efficient shape analysis on the second level, the vertices that are inside in each of these voxel pairs have to be determined. Therefore, it is necessary to walk through all vertices

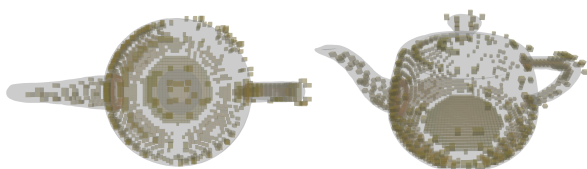


Figure 10: The left side shows a top view of the difference set $B_d \setminus A_d$ on top of the transparent 3d mesh using the color gradient to illustrate the strength of deviations. The screenshot to the right contains a side view of the scene.



Figure 11: Difference sets $A_d \setminus B_d$ (left) and $B_d \setminus A_d$ (right) resulting at a dense voxel size of 0.25mm (fourth run of the second object in Table 3). In addition, the color gradient is used to illustrate the strength of the deviations.



Figure 12: Top row: difference $A_d \setminus B_d$, intersection $A_d \cap B_d$ and difference $B_d \setminus A_d$. Red voxels are representing outliers, yellow voxels are representing inliers. Bottom row: A_d and B_d , outliers and inliers are combined with the color gradient.

of the planning data and 3d reconstructed meshes and check, which vertices are within both sets A_s and B_s .

To reduce the costs, each sparse voxel of sets A_s and B_s has information about the vertices that are associated with a voxel cell and its nearest neighbors (kernel) via the vertex indices. This means, that there has only to be checked, which vertices that are related to this sparse voxel cell and its neighbors intersects with the voxels of the dense voxel sets A_d and B_d . This results in an overall representation of the determined deviations provided by the high-resolution second level voxelization and corresponding shape analysis (see Figure 13).

Real-time Visualization

As already mentioned at the end of the Multi-Level Voxelization Subsection in Chapter 5, the resulting voxel sets are written to files during the voxelization and shape analysis process to reduce memory usage, but also to allow a high resolution voxelization, which

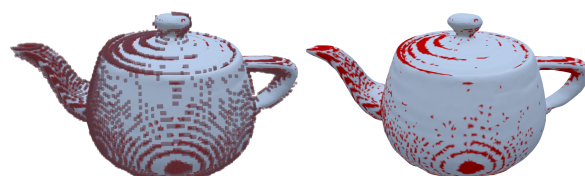


Figure 13: Left: combined rendering of the difference set $A_d \setminus B_d$ and the reconstructed teapot mesh. Right: determined deviations transferred to mesh data as final visualization for fast rendering and easy data sharing.

is performed step by step using the proposed MLV. Another advantage is that the voxel sets are saved persistently and the shape similarity can be recalculated whenever the data is loaded without a new comparison.

To alleviate performance issues, rendering the high resolution voxel sets in a standard Unity application (where every voxel represents one interactive object), a prototypical loader using Unity DOTS (Data-Oriented Technology Stack) that depends on the ECS (Entity Component System)⁵ paradigm [MG20], was also implemented. Compared to a standard Unity application, this results in a significant performance gain, as shown in Table 5.

DS	$ A + B $	VoxMesh	VoxViz
2mm	3,345	~355.6 fps	~744.8 fps
1mm	13,658	~119.5 fps	~504.4 fps
0.5mm	54,771	~21.7 fps	~211.6 fps
0.25mm	219,004	~4.6 fps	~65.1 fps
0.125mm	876,082	~1.0 fps	~16.3 fps

Table 5: Comparison of our VoxMesh and VoxViz tools for real-time rendering of the cat mesh's voxel sets.

7 RESULTS AND DISCUSSION

In addition to the resolution of the voxelization and the vertex count of the geometries to be compared, the performance also depends on the selected types of result presentation. The latter consist of representing the deviations either voxel-based using only one color for each resulting voxel set (see Figure 9) or two colors for each difference set determining inliers or outliers with the option, to also use a gradient to visualize the strength of the deviations (see Figure 10, 12), or a mesh-based representation, where deviations are visualized by coloring the mesh at the corresponding locations (see Figure 13). For example, the fourth run of the second object (reconstructed teapot) in Table 2 using the setup mentioned at the end of the Results and Evaluation Subsec-

⁵ https://docs.unity3d.com/Packages/com.unity.entities@0.51/manual/ecs_core.html

tion in Chapter 5 took 04:54 min using one color for each voxel set, 10:20 min with a enabled gradient and 07:42 min using the mesh-based representation. This is due to the fact that voxel-based representations using the gradient and mesh-based representations are considering the vertices of the meshes for visualization.

Concerning shape analysis and the compared statistical methods (DI, JI, KI as defined in Section 5) it can be mentioned, that the results of the DI and KI are mostly similar, which means that a choice between them using as a threshold does only matter if the result has to be very precise. In terms of JI, the results are a little bit lower compared to DI and KI.

Regarding the shape analysis between the voxelized hulls and interior volumes (see Table 2 and Table 3), it can be stated that this results in a higher deviation or lower shape similarity with decreasing voxel size using the voxelized hulls. On the one hand, this is in close correlation with the previous alignment and geometric registration of the two geometries to be compared, which also had to be corrected afterwards using a PCA.

On the other hand, the shape analysis is affected by the uneven surface structures of flat surfaces regarding the reconstructed meshes, if a high resolution (e.g., voxel size < 0.5mm) is chosen. With respect to the volumetric voxel sets, the resolution of the voxel grids did not have a large effect on the shape similarity results, as the resulting voxel sets only grew strongly in terms of intersection, which was significantly higher compared to the intersection of voxelized hulls with the same grid settings.

Errors or deviations that can occur during the scanning process and the 3d mesh reconstruction are not taken into account. In terms of scanning, the occurrence of errors is strongly correlated with the used scanning devices (in our case out-of-the-shelf hardware), scan settings and acquisition conditions (e.g., print material and lighting). For the mesh reconstruction the included software (ExScan) without any mesh optimizations was used to avoid deformations based on that optimizations.

The determined deviations are either visualized using a voxel-based representation or mesh-based representation. Using the voxel-based representation allows to visualize missing parts by using the voxelized geometry from the counterpart mesh, where this parts exists (see Figure 7). In addition, a threshold can be used for a maximum permitted deviation. All deviations smaller than this threshold are colored using a gradient based on the distance of the voxel to the next vertex in the mesh. All deviations above (e.g., deviation > 1mm) are colored using the maximum gradient.

Another advantage is based on the overlay of the voxel sets and the transparent mesh, which makes it possible to identify whether an individual voxel is outside or

inside the object (see Figures 8 and 10). In addition, inliers and outliers can also be classified during the MLV (see Figure 12). Regarding missing parts using our implementation of the mesh-based representation, a visualization of missing parts is only possible for objects, where this parts exist, by coloring these mesh parts (see Figure 13).

Our prototypical implementation also allows saving the center positions of all resulting voxel sets that can be either used as point cloud input or for reproducing the voxel sets in other animation software such as Blender with the use of geometry nodes.⁶ In addition, the resulting voxel sets can also be persistently saved and loaded using an internal format, but also using the *.obj* format, to support a wide range of third party 3d tools.

8 CONCLUSION & FUTURE WORK

In this paper, a concept for visualizing deviations between topologically different 3d planning data and the corresponding 3d reconstruction of 3d printed objects based on a multi-level voxelization was presented and prototypically implemented as a Unity application. During the voxelization process, a shape analysis using statistical methods was performed on the voxelization result to determine the shape similarity of the planning data along with its 3d reconstruction. Furthermore, this included a comparison between the resulting voxel sets, which either consisted of the voxelized hulls of the meshes or volumetric data. Determined deviations are either visualized using a voxel- or mesh-based representation to easily find errors in the 3d printed object.

In this context, further work regarding the voxelization is subject to continuous development and optimization, which also involves the initial overlay and alignment. This could be extended, for example, by using the ICP (Iterative Closest Point) technique. With regards to the voxelization of the meshes, the process could also be implemented using Unity's ECS system, as it is already used in the context of the result visualization in our VoxViz tool, which in addition to the associated increase in performance additionally enables a parallelization of the workflows (like parallelized comparison of several voxels of the sparse grid).

For the voxel-based representation a gradient was used to visualize the strength of found deviations, which has not been applied yet to the mesh-based representation to colorize the mesh surface depending on the strength of the deviations. Since a visualization of the voxelization result in the 1 to 2 millimeter range can also be performed by mobile devices, an overlay of the voxelized surfaces with the printed object is also conceivable in order to illustrate deviations using augmented reality.

⁶ https://docs.blender.org/manual/en/latest/modeling/geometry_nodes/index.html

9 REFERENCES

- [BSSW21] Sebastian Bickel, Christopher Sauer, Benjamin Schleich, and Sandro Wartack. Comparing cad part models for geometrical similarity: A concept using machine learning algorithms. *Procedia CIRP*, 96:133–138, 2021. 8th CIRP Global Web Conf. - Flexible Mass Customisation (CIRPe 2020).
- [CHL⁺17] Xin Chen, Jingbin Hao, Hao Liu, Zheng-tong Han, and Shengping Ye. Research on similarity measurements of 3d models based on skeleton trees. *Computers*, 6:17, 04 2017.
- [DFFW18] Jozef Doboš, Carmen Fan, Sebastian Friston, and Charence Wong. Screen space 3d diff: a fast and reliable method for real-time 3d differencing on the web. In *Proc. Web3D '18*. ACM, 2018.
- [DGJ20] Andreas Dietze, Paul Grimm, and Yvonne Jung. Visualization of differences between spatial measurements and 3d planning data. In *In Proceedings of the 25th Intl. Conf. on 3D Web Technology*, pages 1–5. ACM, 2020.
- [DGJ21] Andreas Dietze, Paul Grimm, and Yvonne Jung. Updating 3d planning data based on detected differences between real and planning data of building interiors. In Vaclav Skala, editor, *29th Intl. Conf. on Computer Graphics, Visualization and Computer Vision (WSCG '21)*, pages 1–6, Plzen, Czech Republic, 2021. CSRN.
- [FI18] Sam Fletcher and Md Islam. Comparing sets of patterns with the jaccard index. *Australasian Journal of Information Systems*, 22, 03 2018.
- [KWL20] Ha-Seong Kim and Myeong Won Lee. 3d object recognition using x3d and deep learning. In *25th Intl. Conf. on 3D Web Technology*, Web3D '20, New York, NY, USA, 2020. ACM.
- [LEAM⁺19] Zouhir Lakhili, Abdelmajid El Alami, Abderrahim Mesbah, Aissam Berrahou, and Hassan Qjidaa. 3d shape classification using 3d discrete moments and deep neural networks. In *Proc. of 2nd Intl. Conf. on Networking, Information Systems and Security*, NISS19, New York, NY, USA, 2019. ACM.
- [MG20] Mathieu Muratet and Délia Garbarini. Accessibility and serious games: What about Entity-Component-System software architecture? In *GALA 2020*, Laval, France, December 2020.
- [MR01] Novotni Marcin and Klein Reinhard. A geometric approach to 3d object comparison. In *Proceedings Intl. Conf. on Shape Modeling and Applications*, pages 167–175, 2001.
- [NZL⁺20] Weizhi Nie, Yue Zhao, An-An Liu, Zan Gao, and Yuting Su. *Multi-Graph Convolutional Network for Unsupervised 3D Shape Retrieval*, pages 3395–3403. ACM, New York, NY, USA, 2020.
- [OGBS06] Ipek Oguz, Guido Gerig, Sebastien Barre, and Martin Styner. Kwmeshvisu: A mesh visualization tool for shape analysis. *The Insight Journal*, 2006.
- [RMWQ19] Baltej Singh Rupal, Khaled G. Mostafa, Yeping Wang, and Ahmed Jawad Qureshi. A reverse cad approach for estimating geometric and mechanical behavior of fdm printed parts. *Procedia Manufacturing*, 34:535–544, 2019. 47th SME North American Manufacturing Research Conf., NAMRC 47, Pennsylvania, USA.
- [WWWH21] Martin Weinmann, Sven Wursthorn, Michael Weinmann, and Patrick Hübner. Efficient 3d mapping and modelling of indoor scenes with the microsoft hololens: A survey. *PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 89:319 – 333, 2021.
- [WZL⁺20] Bing Wang, Shuncong Zhong, Tung-Lik Lee, Kevin S Fancey, and Jiawei Mi. Non-destructive testing and evaluation of composite materials/structures: A state-of-the-art review. *Advances in Mechanical Engineering*, 12(4):1687814020913761, 2020.
- [ZAB⁺16] Fatima Rafii Zakani, Khadija Arhid, Mohcine Bouksim, Taoufiq Gadi, and Mohamed Aboulfatah. Kulczynski similarity index for objective evaluation of mesh segmentation algorithms. In *5th Intl. Conf. on Multimedia Computing and Systems (ICMCS)*, pages 12–17, 2016.
- [ZWB⁺04] Kelly Zou, Simon Warfield, Aditya Bharatha, Clare Tempny, Michael Kaus, Steven Haker, William Wells, Ferenc Jolesz, and Ron Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index. *Academic radiology*, 11:178–89, 02 2004.

Operational theater generation by a descriptive language

Matis Ghiotto

Aix Marseille Univ, CNRS
LIS, Marseille, France
matis.ghiotto@lis-lab.fr

Brett Desbenoit

Aix Marseille Univ, CNRS
LIS, Marseille, France
brett.desbenoit@univ-amu.fr

Romain Raffin

Université de Bourgogne
LIB EA 7534, Dijon, France
romain.raffin@u-bourgogne.fr

ABSTRACT

3D landscapes generation is an interdisciplinary field that requires expertise in both computer graphics and geographic information systems (GIS). It is a complex and time-consuming process. In this paper, we present a new approach to simplify 3D environment generation process, by creating a go-between data-model containing a list of available source data and steps to use them. To feed the data-model, we introduce a formal language that describes the process's sequence. We propose an adapted format, designed to be human-readable and machine-readable, allowing for easy creation and modification of the scenery. We demonstrate the utility of our approach by implementing a prototype system to generate 3D landscapes with a use-case fit for multipurpose simulation. Our system takes a description as input and outputs a complete 3D environment, including terrain and feature elements such as buildings created by chosen geometrical process. Experiments show that our approach reduces the time and effort required to generate a 3D environment, making it accessible to a wider range of users without extensive knowledge of GIS. In conclusion, our custom language and implementation provide a simple and effective solution to the complexity of 3D terrain generation, making it a valuable tool for users in the area.

Keywords

Geographics data, operational theater, descriptive approach, multi-modal geometry processing

1 INTRODUCTION

3D landscape generation is an active topic of research in the field of computer graphics and GIS. Operational theater consists in a subset of landscapes used for serious games or military simulations. The goal of operational theater generation is to create realistic representations of natural and urban environments based on real-world data, or GIS data. The integration of GIS data into 3D landscape generation is a challenging task. It requires the efficient processing of large and complex datasets as well as some knowledge of cartography and coordinates system. Environment generation in the industry are made by either procedural or sketching [2] tools. As procedural generation tends to create an artificial (yet realistic) terrain, as opposed to a terrain depicting a real place, it does not fit for operational theater generation. Sketching is more adapted, allowing human intervention to curate the terrain into something close to the existing surface it tries to emulate, but it is a lengthy process. Sketching requires expertise in

GIS and use of sketching software. In this paper, we propose a method to reduce time and complexity of operational theater generation by limiting sketching to a formalized description of the terrain to generate. Smeilik et al. [11] gave a relevant overview of the different elements used to compose a 3D environment, but also highlighted the lack of methods allowing for the generation of a complete terrain using all the different elements. In recent literature, this segregation between natural terrain and urban terrain is still existing. Eric Galin et al. [3] present a state-of-the-art review for the different methods of natural scene generation, but with no insight on how to integrate them in an urban landscape. In contrast, Hoang Ha et al. [8] only approach road network generation and Tang Ming [12] city generation. Each paper focuses on a specific type of terrain generation and, when they provide a new way to simplify their specific task, they do not provide an easy way to interface their work with others of the same type, in order to generate a complex landscape with natural and urban terrain alike. Our study-case is the generation of an operational theater fit for military simulation. It is a complex task with concrete industrial application. Operational theater includes both urban and natural environment. We address this problem by a flexible way, generating any type of terrain that can be easily enhanced by future field-specifics works. The output at this time is a full 3D scene, merging with geomet-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

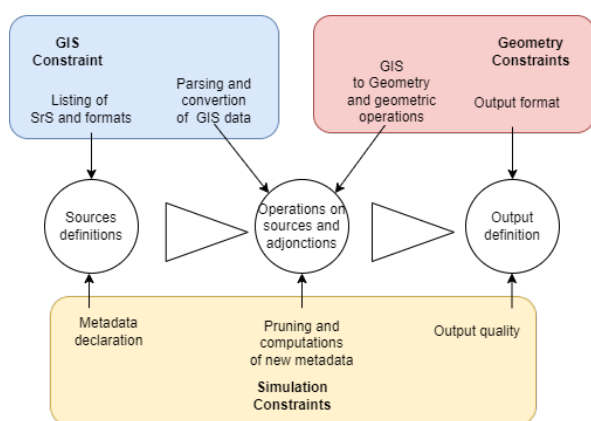


Figure 1: Language principle overview

ric processing of geographic data. The new method we propose rely heavily on the formal description of the scene to generate, so our first approach was to identify the different elements that make an operational theater.

This lead to a three-step process represented in figure 1:

- identify sources, ranges and format
- extract, process and merge data
- adapt outputs

We defined a data model and a specific language to generalize our approach and facilitate user's interactions. This language will be embedded in a file and will be sufficient to create a full operational theater.

1.1 GIS data diversity

The input of our process is mainly GIS data. It involves handling geospatial ranges, coordinates and references, data types and associated metadata. Surface coordinates face approximation due to the curved surface of earth. Geocentric coordinates are inconsistent with altitude, as the sea level is not constant and the earth is not a perfect sphere. This lead to historical disparities between coordinates systems [4]. It has become even more complex with new technologies and the number of digital data formats.

Initiatives have been made to solve this problem. The Open Geospatial Consortium (OGC) is a major actor in the interoperability of geographical data. The consortium has introduced many standards, helping to harmonize the use of GIS data and interoperability between them. Webservices such as Web Map Service (WMS) and Web Feature Service (WFS), are OGC standards. They allow users to fetch information (maps and features) from geographic servers. It is an important part of multi-format data process, as it decentralizes data sources. Each Webservice can implement a process for the data type it uses and requires minimal user's expertise. Webservices are a reliable bridge between OGC

data formats if the user knows which format they have access to and which format they want to use.

Even with OGC standard, some tasks are traditionally performed in specific formats that are not adapted for a generic operational theater generation. For example, describing a city with CityGML is a documented task [13]. But, if we need to incorporate this city into a larger environment, CityGML is no longer suitable. The transformation of CityGML data into another more suitable format will need processing and expert knowledge. Hopefully, the format is documented and based on norms.

Our first problem will be to use data in different formats that are not trivially intercompatible. To do so, we will need to list the formats and be able to use the right OGC protocol to translate them. This is depicted in figure 1 as GIS constraint in blue.

1.2 Geometry constraints

Rendering geographical data on screen require their conversion into geometrical elements. We will first describe how to construct the operational theater from the geometrical point of view. An operational theater is composed of a base, the ground terrain and surface elements, such as vegetation or buildings. The terrain is a Digital Terrain Model (DTM) [3] with geographic information superposed on it in most cases. This geographic elements, studied in [11], can be grouped in three geometric categories:

- Surface elements, which cover large parts of the DTM and represent a large homogeneous chunk of the environment such as sea, farm, forests...
- Discrete elements which are small meshes and usually represent buildings, individuals trees or particular geological formations.
- Continuous elements which are linear components, such as roads or rivers.

The geometry can cover a wide country-sized area. Even with optimization [5] [15], covering that surface and retaining details is not trivial. Large operational theaters, typically covering over 250,000 hectares (50 km × 50 km) with 1 meter to 10 meters precision, require tiling and level of detail (LoD) or a patch-based generation method [1]. Tiling refers to the subdivision of the terrain into smaller areas that can be loaded and cleared as needed.

LoD refers to the superposition of meshes of various resolutions to represent the same object, with only one visible at a time, depending on the proximity between objects and the viewpoint origin. The use of both LoD and tiling reduces much of the cost and allows large and more detailed scenery to be made suitable for rendering.

Our second problem is to split the GIS elements into the corresponding geometrical elements, and process them to create tiling and LoD appropriate for rendering. This can be generalized as all optimization operations made on the data, and is displayed in red on figure 1 (Geometry constraint).

1.3 Simulation constraints

Thirdly, there are specific requirements for simulation. Creating a scenery for 3D mobiles deals with different constraints than creating a web application for flood control [10], [6]. These constraints vary from a simulation to another and can be summarized as:

- Geographical extent, referencing the part of our world described by the scenery
- Theater accuracy, whether it is exactly the same as reality or if there is a margin of error on the present elements
- Metadata, such as soil quality, vegetation density, or street names
- Specific information requiring computation, such as an inter-visibility check or ground distance computations

A user may need an operational theater to comply with any number of these previously defined constraints. Modifying any of these constraint parameters calls for recreating the entire operational theater. It is a time-consuming operation without an automated generation process. Therefore, our third problem is to modify parameters or add new data or method to the operational theater without having to rethink the entire creation process. This impacts the generation process at all levels and is displayed in yellow in figure 1 (Simulation constraint).

These problems are difficult to handle. Users must understand simulation constraint as well as geometrical and GIS constraints. Moreover, once an operational theater is generated, it is not possible to modify it with new data or add support to a new simulation constraint, without restarting a new creation from scratch. In the method we proposed, we shift this difficulty into something easier to manage: the created landscape is still static, but the model representing it, written in a human-readable script, is easy to modify.

2 DATA MODEL

In previous sections, we identified difficulties linked to the many constraints of operational theater generation. To address them we present a new approach, based on a descriptive-oriented language, seeking to explicit simulations needs and constraints that can be used to automatically create an operational theater.

2.1 Overview

The identified constraints are linked to three main categories:

- GIS constraints
- Geometry constraints
- Simulation constraints

We consider these constraints as tangled elements rather than independent elements. We unraveled them and exposed that GIS constraints never impact the geometrical output outside construction. With the same logic, outside the specific data format that was already accounted as a GIS constraint, geometry constraints are never used to determine what kind of sources will be used as base material for the operational theater generation. Our approach is to propose a description of these two subsets of constraints (the available data and the expected output) to get parameterized input and output, adapted to computer process. Then, we define a subset of operations allowing to extract and modify the input to obtain the output. As shown in figure 1, this is the global structure of our solution.

Sources used are determined from available GIS data according to a geographical extent and metadata needed by simulations constraints.

Outputs are determined from geometrical constraints, limiting data quantity and format, and by simulation constraints requiring a minimal precision and quality.

2.2 Model specification

Generation methods based on scripting exist in the literature [12] but are traditionally not used in conjunction with wide real-world data. To handle these complex data, we took inspiration from L-System [9] and conceived a theoretical data model that allows data mutation. This abstract model is fed and manipulated by a scripting language. The language we defined is composed by three core-concepts: Data, Sources, and Operations. The main idea of the language is to handle these concepts to create a data model representing an abstract operational theater before processing.

Data represent all the information needed in the 3D scenery representation. They can be assigned as variables to feed operations.

Sources are all the protocols needed to communicate with external data sources. We collect sources to obtain raw data or letting a server do requested modifications on raw data before their acquisition.

Operations encompass all the modifications that can be done to process some data and result to other ones, assigning them to a variable.

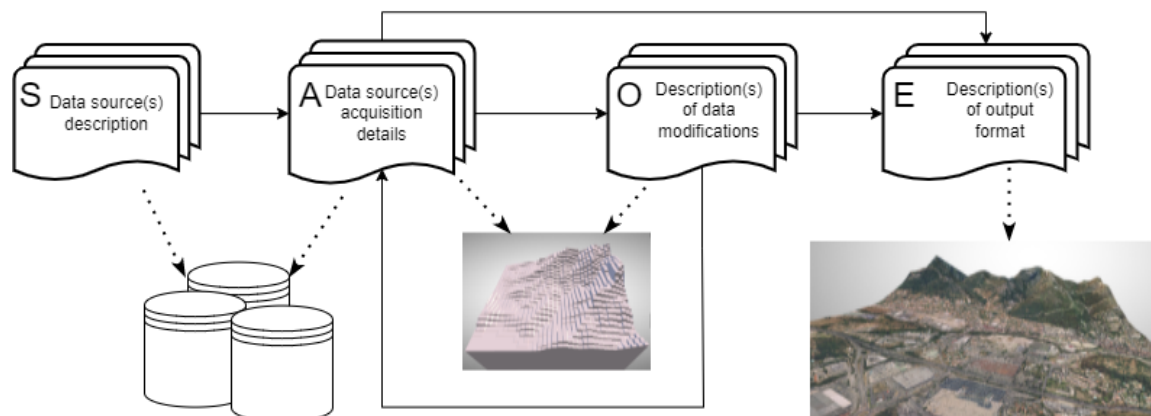


Figure 2: Description language schematic representation

The Data model is the most complex part of our language. This is the part where we perform acquisition and transformation of data sources, so we have usable and inter-operable data. The Data model allows a global and abstract representation of scenery while retracing the origin of all available data, lists all the transformations and applies them. As described before, these data can contain metadata about their actual quality or about their formats, and it's possible to use these as criteria to manipulate them. This model is innovative as it allows a representation of a 3D scenery composed of multiple georeferenced elements and their relative disposition without being a graphical data.

Operations can make use of external software by specifying the executable path and the output location. The commands will be handled as if the executable output is a data from a specified format coming from a local, non-mutable, source. It allows extensibility for the language, making use of state-of-the-art processes without having to re-implement them.

3 METHODS

3.1 Language

We proposed the implementation of the previous concepts in a language including the followings elements:

- One or more Source-declaration blocks (formula 2) that can be either:
 - A local source, indicating path, format and all necessary metadata (formula 3).
 - A geoserver source, indicating the connection address, the protocol (WMS, WFS, WPS) and optional connection parameters (formula 4).
- One or more data Acquisition blocks (formula 5), indicating a predeclared source, an identifier (id) for the newly obtained data and an acquisition parameter if the source allows it.

- Any number of Operation blocks (formula 6) declared with the following information:
 - A declared source data id.
 - A not previously declared new data id.
 - A processing name (already implemented or external command).
 - Arguments needed by the operation.
 - Any number of nested operation blocks.
- One or more Export blocks (formula 9), containing:
 - The format for the output model.
 - Constraints linked to this format (such as number of vertices, texture quality, etc.).
 - The list of data to use for model creation.
 - A fusion heuristic if the format is implemented with more than one fusion processes.

Formally, the language specification is as following:

Let D the description language, S a source block, A an acquisition block, O an operation block, and E an export block, we define:

$$D = S^+ A [A \cup O]^* E^+ \quad (1)$$

Figure 2 is a schematic representation of this equation 1.

3.1.1 Sources

Sources blocks noted S are defined as:

$$S = S_1 \cup S_2 \quad (2)$$

$$S_1 = t_1 asfg^* \quad (3)$$

$$S_2 = t_2 asg^* \quad (4)$$

With S_1 and S_2 respectively a local and distant source. t is a source type (t_1 covers file sources and t_2 covers

geoserver sources), a is an address or a path, f is a file format, c is a CRS, s is a source id. A specific s id must be present in only one S block. g is an unspecified argument, it may be used to convey additional informations, g is included in the language for the sake of extensibility. It can be any number of g arguments. Block S , which stands for Source, describes the data to obtain and how to obtain them. They are declared in the header of our description file and can show different behaviors depending on the given arguments at the time of data acquisition.

Sources can depict a folder, local or upon a network (t_1), a requestable database, or a geographical data server (t_2). Geographical data servers can alter data before serving them. These alterations are defined by the geoserver standard, and they can be, but are not limited to: georeferencial rebasement, aggregation of multiple data on the same bounding boxes and segmentation of data to obtain only the one inside a defined bounding box.

This differentiation is made via a specific keyword and does not impede future implementations of other data acquisition methods.

S blocks are logically linked to Acquisition block, denoted by A .

3.1.2 Acquisition

$$A = sdbg^* \quad (5)$$

s is a source id and d is a data id and b a geographic bounding box limiting the data to get. A specific source id denoted by s can be used only if it was previously declared in a S block. A specific d argument can't be reused if already used in a A block. A b argument must describe a bounding box included in the one declared in the corresponding s . Block A allows loading the information described in a source into the data model. It makes use of all the information specified in the source and the added arguments g to create a named data d , which will be viable to modify or export later. The same S block can be used by multiple A blocks to create different data by specifying various bounding boxes b or using the arguments to request different server-side data management.

3.1.3 Operations

Operations blocks noted O are defined as follows:

$$O = O_1 \cup O_2 \quad (6)$$

$$O_1 = od_1 d_2 g^* O^* \quad (7)$$

$$O_2 = pd_1 d_2 a f g^+ O^* \quad (8)$$

Where d_1 represents an already existing data (and thus must be previously declared) that will be used as source

for creating a new data d_2 . d_2 must not have been previously declared. In O_1 , an o operation will be declared to be executed by the interpreter, using an internal process to modify a d_1 data into a d_2 data using only this data and optional arguments depending on specifics o . In O_2 , an external process hosted at address p will be executed, its result must be stored in a . Once completed, the file at the address a is read and loaded into d_2 , as if it was a local source of format f . O blocks augment the data model with newly made data fitting the user's needs.

3.1.4 Export

Export blocks denoted E are defined as follows:

$$E = fd^+(cg^*)^* \quad (9)$$

c is used as a geometry constraint to create an exportable version of the data model.

The E blocks, or Export are in charge of producing the 3D scene or other data format specified by the user. This part allows the specification of the non-geographical simulation constraint c . These constraints are deeply linked to a specific format, and thus, do not have a place in the data model. These constraints can be the quality of the 3D meshes created, their sizes, or their formats. It can also be the used metadata, or simulation-only data that must be attached to the scenery elements.

We presented a proof of concept for a new method of landscape generation, incorporating the basic features required to create a landscape. Our approach includes the ability to import GIS data, as well as the ability to use process from external tools, such as GDAL, and maintain relationships with the source data.

3.2 Implementation

The implemented subset of the language contains the following elements:

- Data format
 - Image for texture purpose, PNG and TIFF
 - Raster data, tiff and XYZ
 - Vector data, Shapefile and GML
- Operation
 - HeightMap mesh creation from XYZ data
 - Mapping of terrain mesh and vector features
 - Texturing
 - Land flattening around vector features
- Output formats

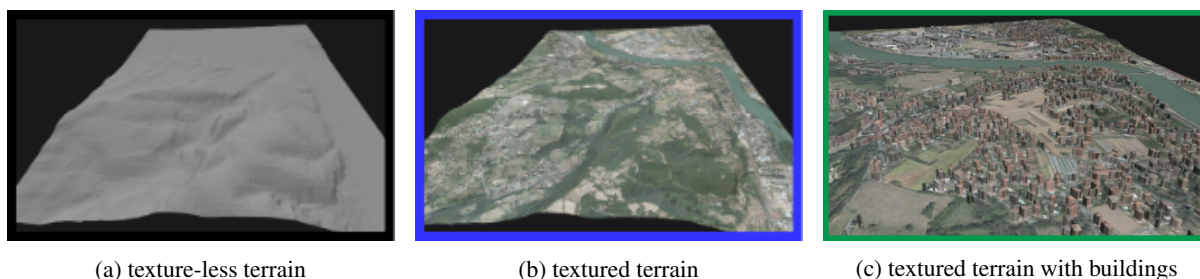


Figure 3: Visual examples

– GLTF as 3D mesh

In our proof of concept, we have selected XYZ, TIFF, GML and Shapefiles as sources data formats. We selected these formats due to their support from GIS standard libraries such as GDAL [14] and widespread availability, notably by local organizations such as the French survey (IGN, from which we get most of the data). For our test outputs, we have chosen the GLTF format, backed by Khronos, as it provides an optimized geometric format for rendering and is straightforward to write. This choice of formats and processes reflects a meaningful range of capabilities within the GIS field, as it covers all steps involved in the creation of a complete terrain. The process starts with data collection and includes the aggregation of vector and raster data, leading to the creation of a simulation-ready 3D operational theater. We have also added some specialized operations to demonstrate the versatility of our model, its ability to integrate various data sources, and its alignment with the needs of simulations.

The executable running the language is implemented with C++. The computer executing it has a AMD Ryzen 5 3400G processor with 4 cores and a 16 GB RAM.

This implementation is a showcase, and the operations can be further optimized with state-of-the-art algorithms. The test case's goal is to determine the difficulty to produce different operational theaters. We evaluate the time used to create a specific theater, and also assess the visual coherence of the results, the skill level required to conduct the test case and the ease of modifying it to meet other requirements.

4 RESULTS

To represent elements constituting our scenery, we chose an indicative and functional-like language, based on the JSON standard. JSON has the advantage to be easily interpreted by computers without being too harsh to read and write by humans.

4.1 Incremental complexity

The figures 6i, 6j and 6k show one of those files. We can see the all the elements of our language represented here.

Each letter under bracket (and text color) shows a different step of the test. The same bracket letter (and corresponding color) is used for figures 3 and 4. The construction of this file is incremental. The text under brackets "a" (black text) alone is enough to create a simple texture-less terrain, and adding step by step, "b" (green) and "c" (blue) will refine it into a textured terrain with buildings. "d" (red) shows an optional terrain modification to demonstrate the possibility of the language.

"Sources" in figure 6i that can contain one or more S Sources as defined previously, here contains a $t1$ "localAccess" named s SourceHeightmap at address a `"/GrandLyonData.XYZ"` with a data format f "XYZ" and a geographic system c "EPSG:3946". A second local source with similar parameters named "SourceBuildings" and a distant source "SourceOrthophoto" are also present, but we will see them in detail later with an advanced example.

"Populate" in figure 6j is the first part of the data model and can contain one or more A blocks. Here it contains the acquisitions of data described by the previous s , into a new data id d "LoadedHeightMap" (and respective "LoadedBuildings" and "RequestedOrtho"), but limiting the acquisition to the range of the specified bounding box b , expressed in the previously declared geographic system for or in a new one in the case of Webservice sources as in "b" blocks.

"Build", first half of figure 6k, can contain any number of O blocks. Here it contains three operations o . The first one, under bracket "a", named "HeightMapToMesh", builds a mesh from the previous data d_1 "LoadedHeightMap" to the new data d_2 "TerrainMesh". The second, under bracket "b", will similarly create a new data d_2 "TerrainMeshWithBuildings" using the previously created d_1 "TerrainMesh" and the data given by argument g "LoadedBuilding". The third, under bracket "d", will similarly apply a new geometrical process to the generated data.

"Outputs", second half of figure 6k, must contain at least one way to export E for the data. It contains here an f GLTF output from the d "TerrainMesh" (or "TerrainMeshWithBuild" or "TerrainMeshFlattened" and "RequestedOrtho" if we go farther into the exam-

ples) data using arguments *g*, such as the model size, bounding box and the output filename.

Sections "a" of this descriptor file show a short generation script for making an untextured 3D model. This illustrates the language simplicity. The only skill needed to use it is to be able to express the bounding box coordinate of model integration step in the coordinates reference system (CRS) declared in source description step. It is too simple for any real-world use, but it is a base easy to increment.

Sections "c" show the language capacity to communicate with a geoserver and aggregate sources of different origins. Data present on the geoserver referenced in figure 6i are initially under CRS EPSG:3857, but we can request them in another CRS providing the right arguments in figure 6j. If the data in the geoserver and the one used to create the terrain model are correct, they will match and create a textured terrain when declared together at the GLTF generation in figure 6k. This is as simple as the previous step and will require no more skills. The biggest difficulty is to verify beforehand if the data are not faulty. There is no easy automated way to do so embedded in the language, and it falls to the user charge to verify their data sources.

Sections "b" show the language capacity to re-use previously created element and aggregate complex vector sources. We declare a vector file containing buildings in 6i and 6j as done in the first steps. It should be mentioned that the data are this time from the same provider as the ones used for the Heightmap in the black part, so there was no need to verify if they were coherent with the already present ones. The fusion of the data is happening in the build step in 6k. The process "addBuildingsToMesh" takes a Mesh as source data and a vectorial feature list as argument and creates a new data that contains both. The implementation checks if the data used are of an expected format and issues an error if they are not. With this step we get a minimal but simulation ready operational theater we can further supplement by adding roads and rivers delimitations, vegetation or any required element by following the same steps. We see the progression of these three use-cases in figure 3.

Section "d" shows the model capacity to keep track of data hierarchy. The process declared is a simple geometrical operation that flattens the ground under given features. It may seem of no more interest for our example than the "b" process, as it takes the same kind of arguments and produces the same kind of output. But we only use as input for these process the data created with the precedent processes. This highlights that data are keeping a symbolic link to the data that was used for their creation, and that we can use this hierarchy to easily reuse data. The process can use the feature list of id "LoadedBuildings" referenced by "TerrainWith-

Buildings" to flatten the ground without damaging the buildings, as shown in the wireframe of figure 4.

4.2 External command

The main limitation of this language is the number of implemented processes that cannot possibly match the wideness of GIS. That why the language can also use external processes such as GDAL command line to process data. This is a less flexible way to use data, and it requires knowledge about the external program. The example shown in figure 5 was obtained by using the build step in listing 1.

4.3 Gathering results and analysis

These examples demonstrate how to add a complex element in the operational theater, as well as the communication and fetching data from geoserver. They also demonstrate that a small amount of knowledge of GIS or information processing is required to generate a simulation ready simple operational theater. Some may still be needed to understand errors inherent to the sources, such as a delta between different sources created by the conversion of CRS or by inaccurate sources, which are at least not format-specific knowledge. Once the sources are selected successfully, it is particularly easy to manipulate the script to display more or less elements or another geographic area.

Table 1 gathers analyses of the examples. Three additional tests to the incremental construction have been made: "Contouring with GDAL" is for the contouring example (see section 4.2) that displays a smaller area but with the call to an external process; The "Full large zone" is expensive to generate, but it is coherent as it covers a zone 625 times larger than the first example with the same level of detail. The high triangles count makes it difficult to display or load in a visual engine; the "Full zone split into 16 tiles" is, as the name suggests, the same zone but generated with 16 square tiles instead of only one in the previous example. It is obtained by adding multiple delimitation sections in Models and Outputs (respectively, figure 6j and figure 6k). This method enables reduced generation time and easier load/display.

Listing 1: External process example

```
"Build" : [  
  {  
    "pAdress" :  
      ". / gdal_contour -p -i 10.0",  
    "origData_id" : "HeightMap",  
    "pResult" : ". / contour.png",  
    "format" : "png",  
    "newData_id" : "Contouring"  
  }, ...  
]
```

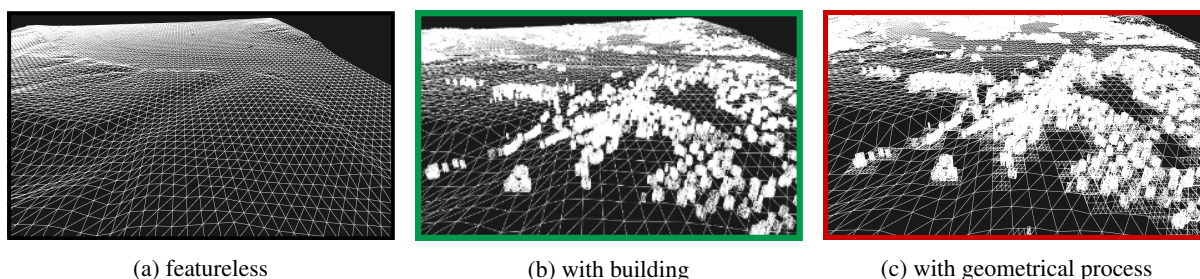


Figure 4: Wireframe examples

	Acquisition Time (s)	Triangles count (k)	Generation time	Surface
a (Black) - Base case	4	4,016	1 min 21 s	2 km × 2 km
c (Blue) - Texturing	12	4,016	1 min 31 s	2 km × 2 km
b (Green) - Adding building	10	8,126	2 min 15 s	2 km × 2 km
d (Red) - Geometric operation	10	10,252	4 min 10s	2 km × 2 km
Other - Contouring with GDAL	4	2.048	32 s	500 m × 500 m
Other - Full large zone	24	6,089,800	3 h 15 min	50 km × 50 km
Other - Full zone split into 16 tiles	24	380,600 × 16	1 h 07 min	50 km × 50 km

Table 1: Summary table

5 PERSPECTIVES

As we present this work, we do not implement all the operations that a user may need. It is a first proof of concept and the next version is planned to include a plugin system to allow GIS or simulation communities to add any sources needed from data or operations, and to upgrade it as new cutting edge algorithms, or with new data formats. Our objective is to focus in particular on the integration of 3D tiles formats, a natural extension of the GLTF we use, allowing georeferenced tiling. This is essential, as tiling shows very good performances compared to non-tiled methods for large surfaces. We also consider adding conditional branch-

ing structures to the language. Doing so will further reduce the knowledge needed for the user, by letting the implementation taking decisions based on formula or metadata. The conditional system, in addition to data-quality measurements, may alleviate to some extent the difficulty of sources selection, by choosing automatically the best of two sources under an objective criterion, such as approximation error on the CRS or comparison with another data marked as "trustworthy". Another perspective is to add new interfaces. We will add dynamic information provider to our data model to communicate the data from the model directly to a simulation client. This is included in a SaaS (Simulation as a Service) and WebGIS 2.0 [7] approach of the simulation's problem that breaks down complex simulation elements into more understandable services. Doing so improves accessibility for nonspecialized users who will only be confronted to a standard interface and is not required to understand all the underlying complexities.

Acknowledgements

The authors acknowledge support by SopraSteria and French ANRT "Association Nationale de Recherche et Technologie" under CIFRE n° 2020/0364. The authors also thank the GrandLyon and IGN open data for their resources.

6 REFERENCES

- [1] Leandro Cruz, Luiz Velho, Eric Galin, Adrien Peytavie, and Eric Guérin. Patch-based Terrain Synthesis. In *International Conference on Computer Graphics Theory and Applications*, Proceedings of the 10th International Conference

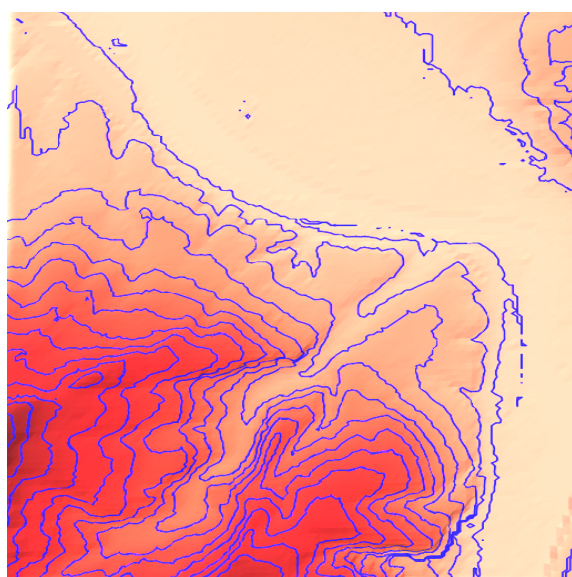


Figure 5: Contouring line by call to an external process

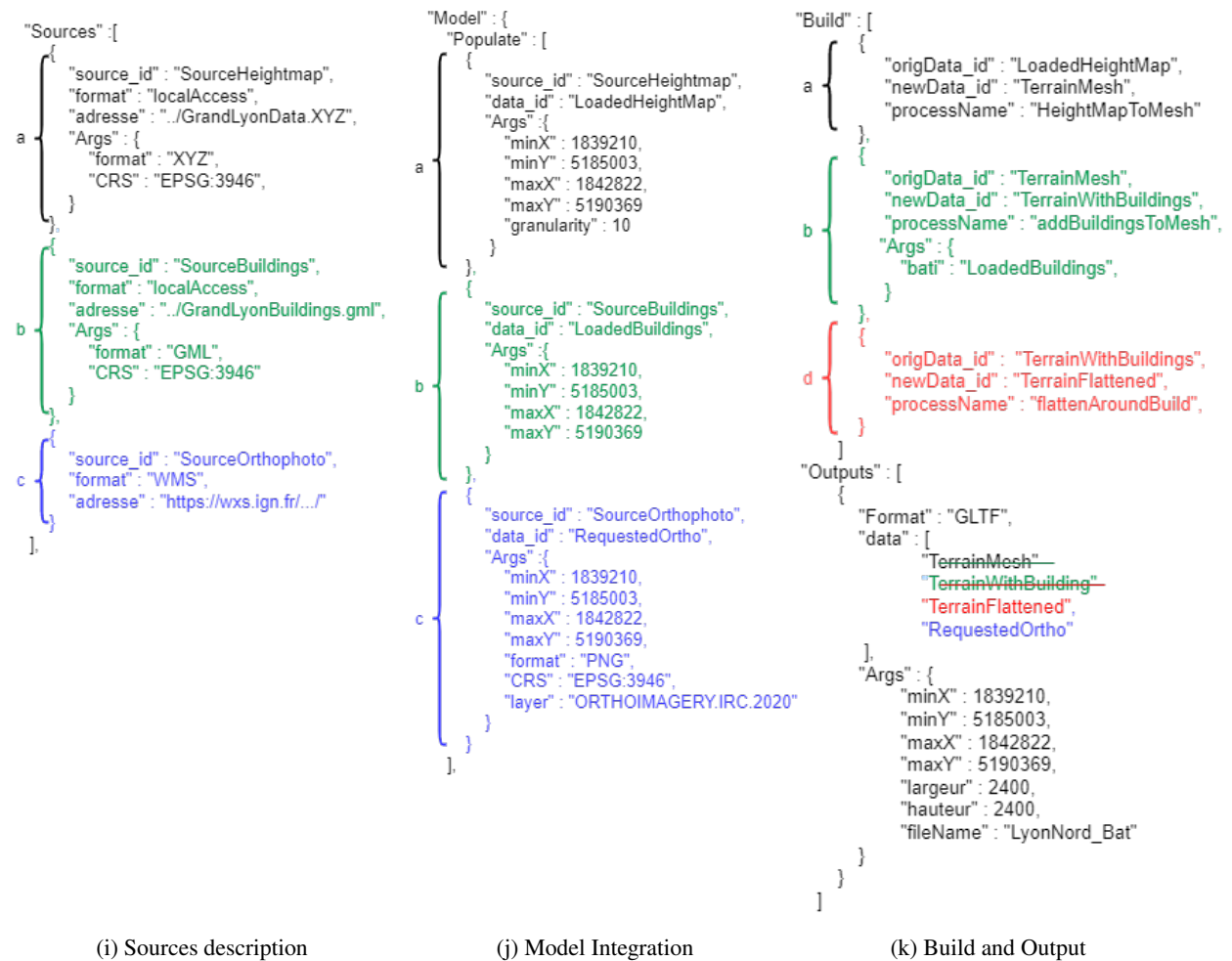


Figure 6: Descriptor file

- on Computer Graphics Theory and Applications, page 6, Berlin, France, March 2015.
- [2] James Gain, Patrick Marais, and Wolfgang Straßer. Terrain sketching. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, I3D '09, page 31–38, New York, NY, USA, 2009. Association for Computing Machinery.
 - [3] Eric Galin, Eric Guérin, Adrien Peytavie, Guillaume Cordonnier, Marie-Paule Cani, Bedrich Benes, and James Gain. A review of digital terrain modeling. *Computer Graphics Forum*, 38(2):553–577, 2019.
 - [4] Ian N. Gregory and Richard G. Healey. Historical GIS: structuring, mapping and analysing geographies of the past. *Progress in Human Geography*, pages 638–653, 2007.
 - [5] Pedro Morillo, Juan Manuel Orduna, Miguel Fernandez, and Jose Duato. Improving the performance of distributed virtual environment systems. *IEEE Transactions on Parallel and Distributed Systems* 16 (7), 637-649, 2005.
 - [6] Rostislav Nètek and Marek Balun. Webgis solution for crisis management support - case study of olomouc municipality. In *ICCSA 2014, Part II*, pp394-403, 2014.
 - [7] Rostislav Nètek, Vit Vozenilek, and Alena Vondrakova. Webgis 2.0 as approach for flexible web-based map application. In *ICCSA 2018*, pp1-5, 2018.
 - [8] Hoang Ha Nguyen, Brett Desbenoit, and Marc Daniel. Realistic urban road network modelling from GIS data. In *UDMV*, pages 9–15, 2016.
 - [9] Yoav IH Parish and Pascal Müller. Procedural modeling of cities. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 301–308, 2001.
 - [10] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.
 - [11] Ruben Smelik, Klaas Jan De Kraker, Tim Tutenel, Rafael Bidarra, and Saskia A Groenewegen. A

- survey of procedural methods for terrain modelling. In *Proceedings of the CASA workshop on 3D advanced media in gaming and simulation (3AMIGAS)*, pages 25–34. sn, 2009.
- [12] Ming Tang. City generator: GIS driven genetic evolution in urban simulation. In *SIGGRAPH Posters*, 2009.
- [13] Gwenola Thomas and Stéphane Donikian. Modelling virtual cities dedicated to behavioural animation. *Computer Graphics Forum*, 19(3):71–80, 2000.
- [14] Frank Warmerdam, Even Rouault, et al. Gdal documentation: Raster drivers. <https://gdal.org/drivers/raster/index.html>, <https://gdal.org/drivers/vector/index.html>, 2022. Accessed: 2022-09-13.
- [15] Ye Zhi, Yong Gao, Lun Wu, Liang Liu, and Heng Cai. An improved algorithm for vector data rendering in virtual terrain visualization. In *2013 21st International Conference on Geoinformatics*, pages 1–4. IEEE, 2013.

Real-Time Reflection Reduction from Glasses in Videoconferences

Marc-André Tucholke* Marie Christoph*

Lasse Anders* Raven Ochlich*

TU Braunschweig

Mühlenpfordtstr. 23

38106, Braunschweig, Germany

{m.tucholke, marie.christoph, l.anders, r.ochlich}@tu-bs.de

Steve Grogorick

Martin Eisemann

TU Braunschweig

Mühlenpfordtstr. 23

38106, Braunschweig, Germany

{grogorick, eisemann}@cg.cs.tu-bs.de

ABSTRACT

Surrounding lighting conditions cannot always be sufficiently controlled during videoconferences, yielding situations in which disturbing reflections might appear on the participants glasses. In this article, we present a retrained neural network to convincingly reduce such reflections. For real time performance we propose an asynchronous processing pipeline accompanied by a head pose-based caching strategy to reuse intermediate processing results. The implementation as virtual webcam allows the system to be used with arbitrary videoconferencing systems.

Keywords

reflection, glasses, video conferencing, image processing, deep learning, face detection, real time

1 INTRODUCTION

In the last years video conferences have undergone a huge rise in usage and popularity. Wearers of glasses often experience reflections in their glasses that distract their counterparts or could reveal sensible information. The aim of this work is a reduction of these reflections in real time. To achieve this we integrate an existing neural network for reflection removal, that is not real-time capable, in a real-time context.

Existing techniques for reflection reduction [LLY⁺23] from a single input image are currently still far from being real-time capable, often reporting processing times of approximately 400 ms. To remedy this, we propose to reduce the computational load by extracting and processing only the relevant part (glasses) of each frame, and propagating the results to subsequent frames.

We detect the region of interest using a learning-based face detection. The segment of the image that contains the glasses is then processed asynchronously by the reflection removal network. Based on the current head pose a reflection mask is applied to the current frame. Through this optimization the processing time per image is reduced to under 40 milliseconds on commodity hardware.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

In Section 2 we introduce the neural network that is used for the actual reflection reduction and differentiate our approach from other methods. We present the goals, approach, and realization of the components of our method in Section 3. An evaluation of the methods performance on multiple metrics is presented in Section 4 before concluding in Section 6.

2 RELATED WORK

Reflection removal has drawn attention during recent years, especially in the field of deep learning [AST⁺22]. Reflection-aware guidance (RAGNet) [LLY⁺23] is a neural network to remove reflections from glass surfaces in real or synthetic images of fully occluded objects or persons behind a glass panel. The task of reflection removal under these circumstances is similar to the presented task of reflection removal from spectacle lenses, but not identical. The main difference is the partial occlusion of the object and the curvature of the spectacle lenses. The V-DESIRR network [PSB⁺21] surpasses RAGNet in terms of reflection removal quality and inference time but both target solely reflections on plain glass. Neither the data set nor the code of the V-DESIRR network have yet been made available to the public, preventing its use in any subsequent research. Another promising approach was shown by Wan et. al. in [WSL⁺21] by removing reflections from images containing partially occluded persons behind a glass panel. Their approach focused solely on faces and incorporated specific facial priors.

* Authors contributed equally

The task is quite similar to the presented task, but the missing open source implementation is again preventing its application in research. Besides single-image reflection removal, multi image methods exist such as [LCL21]. Those methods are not applicable to our problem as we assume input from a single webcam.

The presented work differs from the aforementioned works by focusing on the real-time aspect and the curved surface of glasses.

3 METHOD

In this paper we investigate whether an existing reflection removal algorithm can be adapted to reduce reflections on a user's glasses in real-time. For this purpose we make use of RAGNet [LLY⁺23] an open-sourced current state-of-the-art reflection removal method.

As input we assume a simple RGB image stream from a 30 Hz live stream or input video of size 1920×1080 pixels. The output is a video or a video stream (virtual webcam) with a maximum resolution of 1920×1080 pixels. We further assume, that there is only one person in the image and the person is in focus and decently illuminated.

3.1 Overview

In this section we give an overview of our technique. The flowchart in Figure 1 shows the per-frame steps of our proposed procedure, separated in two asynchronous threads. The main thread reads the input image and computes the position of significant features in the persons' face, usually referred to as facial landmarks. If glasses are detected, the section of the image that contains the glasses (further referred to as glass-section) is extracted and scaled to a fixed resolution.

The glass-section and landmarks are fed to the side thread. There, the RAGNet generates two output images: the reflection map and the reflection reduced image. The output images as well as the landmarks are stored in a cache. The RAGNet distorts the original colors of the image and a color correction has to be applied prior to the storage process [RAGS01].

The main thread detects motion relative to the previous frame. If no motion is recognized, the previously detected reflection mask is reused. If, otherwise, motion exceeds a certain threshold, the cache is searched for previous results of a similar pose. If a matching pose is found, the cached reflection mask is warped to fit the current input image and is then applied to it. In favor of a real time frame rate, the frame is left unchanged if no matching pose was found in the cache.

For evaluation later on, we also implemented a synchronous mode running the RAGNet on each frame without motion detection and cache. Because of the long processing time of the RAGNet this mode is not real-time capable by itself.

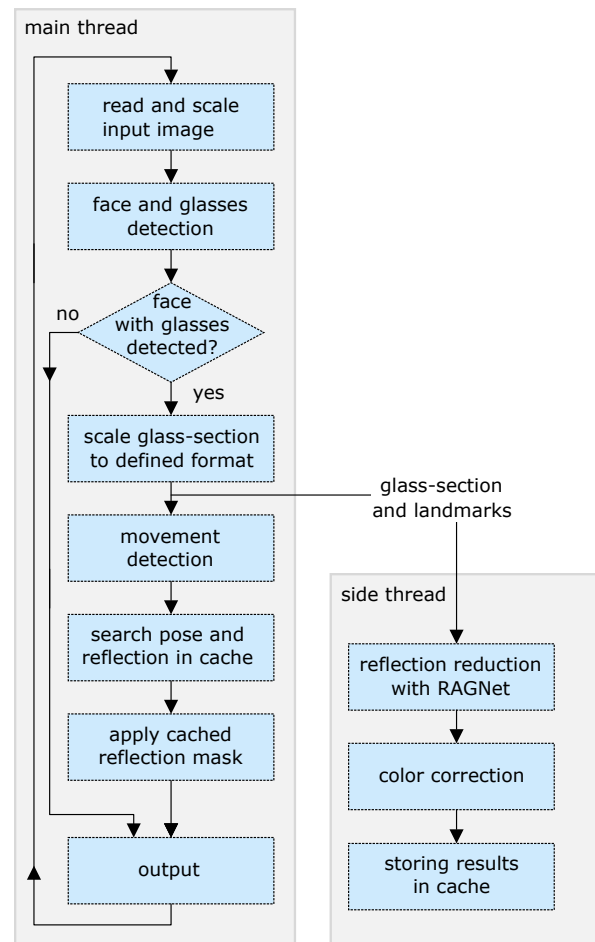


Figure 1: Asynchronous processing pipeline

3.2 RAGNet

The neural network RAGNet follows a two step approach [LLY⁺23]. In the first step the network computes a reflection mask. This mask together with the original image form the input for the second step, where the reflection reduced image is generated. One major task of the second step is to “hallucinate” the content of the image regions where the brightness of the reflection is clipped by the image format limits, i.e., in overexposed regions. This behavior can be seen in Figure 2.

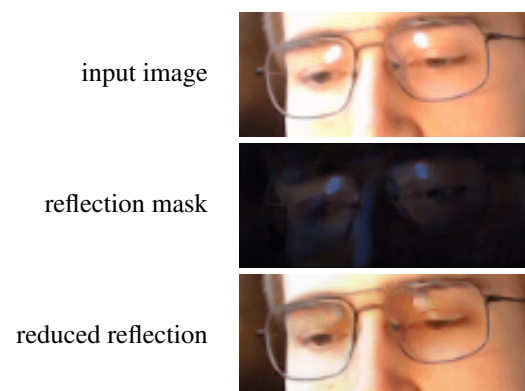


Figure 2: RAGNet hallucinates overexposed regions

3.3 Glasses detection

As applying RAGNet is computationally costly, we extract the region-of-interest containing the glasses and restrict further processing to this region only. The decision if glasses are present in the computed face is based on the presence of edges, i.e., frames of glasses, in three image regions: below each and in-between the eyes (see Figure 3, right). These facial regions, identified based on the landmarks [JBAB00, Tia19, Sid21].

The face detection is realized using DLIB [Kin09], a well established library that robustly handles variations in pose or illumination. Specifically, the landmarks shown in Figure 3 (left) are acquired using the DLIB facial landmark detector [SAT⁺16, KS14]. The computation speed of the DLIB algorithms can be improved by executing them on the graphics card using the CUDA toolkit [NVI]. The performance can further be improved by downscaling the input image. We empirically chose the resolution (1280×720 pixels) such that the rate of correctly identified facial landmarks is nearly equivalent and subsequent computations are not compromised.

The boundaries of the glass-section are determined by the bounding box of the landmarks around the eyes. This yields a robust, accurate and fast detection of the glass-section (see Section 4).

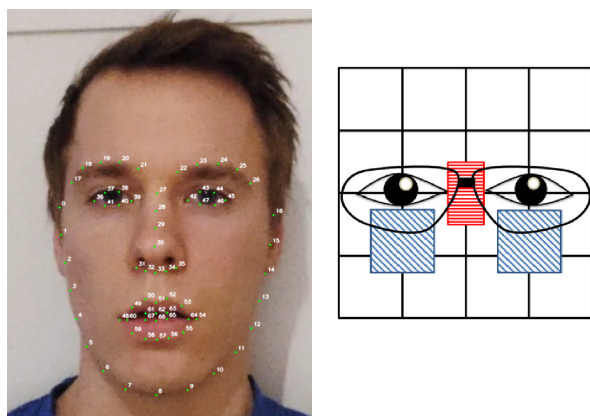


Figure 3: 68 facial landmarks detected with DLIB (left) and examined facial areas according to [Tia19] (right)

3.4 Refined RAGNet

The following sections describe our task-specific transfer learning to optimize RAGNet towards glasses.

3.4.1 Pretrained weights

The pretrained RAGNet [LLY⁺23] removes reflections from images where the content is fully covered by a glass plane. We found that the quality of the reflection removal is still acceptable for our scenario, where only a small part of the image is covered by glass, but the computation times are far from real-time, even when focusing on the glass-section only.

The processing time as well as the quality of the reflection removal depends strongly on the size of the image. We, therefore, scaled the glass-section to 711×300 pixels to achieve stable yet satisfying results.

3.4.2 Recording of and training with synthetic data

To improve the performance of RAGNet we additionally trained it with synthesized training data that specifically resembles our use case more closely than the original training data, i.e. persons with glasses. The synthesizing process to create the training data was similar as proposed in [FYH⁺17].

To train the RAGNet three images are needed, one that remarks the ground truth and has no reflections in it, one that represents the reflections in the image (reflection mask) and the last one that is the original image with the reflections in it (see Figure 4). The unprocessed training frames are extracted from a reflection free video. The glass-sections are then extracted as described in Section 3.3 and rescaled to the demanded size. The reflections are randomly selected from a set of handcrafted prerecorded reflection templates. Within reasonable limits, these templates are randomly scaled, rotated, and intensity-adjusted. In the last step the reflections are added to the ground truth and the edges are smoothed with a Gaussian filter.

With this algorithm a dataset holding 3000 entries was created and RAGNet was trained for 15 epochs with the setup recommended by Li et al. [LLY⁺23]. The validation was done based on the original loss function and the mean peak signal-to-noise ratio (PSNR) on 20 validation data set entries. The training after 15 epochs resulted in improved results, as depicted in Figure 5.

The results of RAGNet trained on the synthesized data sets did not perform well on real test data. This behavior was expected as it was observed by the authors of RAGNet too when they used synthetic data originally [FYH⁺17]. This could be due to overfitting or too unrealistic artifacts. To resolve this problem the training set was extended with real reflection images as follows.

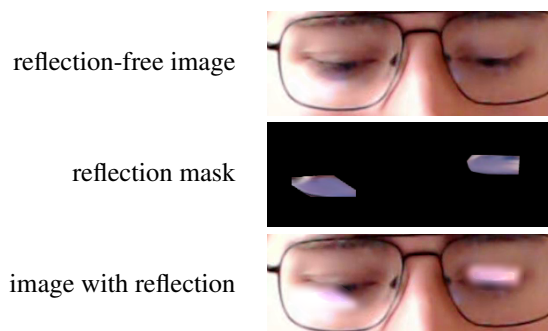


Figure 4: Example of the synthetic training data

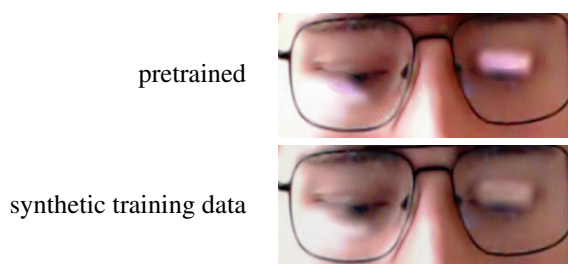


Figure 5: RAGNet original and retrained (synthetic)

3.4.3 Recording of stroboscope data

To generate real test data including images with and without reflection for the same head pose, we used the following setup. A person in front of a PC screen watched a program that alternated a full screen output between plain white and black. During each state, an image of the person was acquired using a webcam. The time between the state changes was chosen so that it enables the light to set and the camera to produce a stable picture but also short enough so that the person's head won't move significantly. As previously mentioned the training needs a third image per data set. The image representing the reflection is calculated by subtracting the reflection free image from the image with reflections. A set of the stroboscope images is displayed in Figure 6.

To create viable test data it has to be assured, that the room where the images are recorded doesn't contain additional reflection sources. The person the images are taken of is ideally illuminated from above and no background light is disturbing the image. Otherwise the whole face would be brighter if the white light of the screen is turned on. If the light from above is too bright the reflections on the glasses would not be significant enough to be seen.



Figure 6: Example of the stroboscope training data

3.4.4 Training with synthetic and stroboscope data

To improve the results of RAGNet the data set was extended by 1095 stroboscope entries and additionally

903 synthetic entries. The data was divided into 60% training, 20% validation and 20% test data, resulting in a training set with 2997 entries.

With this dataset the pretrained RAGNet was refined in the following three steps.

First, the network was further trained for 55 epochs until the loss started to converge. Second, to prevent training towards a local minimum, for which the first step of the RAGNet produced a empty reflection masks, we trained 30 epochs using a modified loss function that included the reflection mask only, until the RAGNet produced plausible reflection masks. Third, to mitigate errors in the reflection free images, the network was trained until convergence for additional 70 epochs with the original loss function, to finish the joint optimization of reflection mask and reflection reduced image generation.

After retraining, RAGNet produced plausible reflection masks and eliminated reflections better than the original version of the network when applied to images of faces with glasses. The inference for a single entry of the validation set is shown in Figure 7.

Since the reflection reduced images show a shift in their color distribution, we extend their processing with an appropriate color correction [RAGS01].

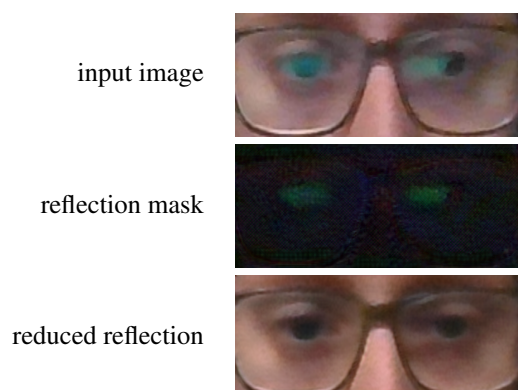


Figure 7: RAGNet retrained (synthetic + stroboscope)

3.5 Motion detection

To reduce temporal artifacts of the cached reflection reduction, that become most noticeable during small head motions, we perform a motion detection, to directly reuse the previous detection in these cases. Furthermore, the motion detection reduces the frequency of cache searches.

To detect motion, i.e., changes between successive images, we compare the current frame with its predecessor via the structural similarity index measure (SSIM) [WBSS04]. SSIM was chosen because it offers to compute similarity only for the brightness of two images, and reflections almost always affect image brightness.

A threshold of 0.95 for the SSIM score was empirically identified to give reliable results.

We further improve performance by computing the motion detection only for the eye region (see Figure 8). The position and size of the region-of-interest around the eyes is again computed based on the facial landmarks, including a certain margin around the eyes to allow keeping the bounds unchanged for the SSIM computation during slight head movements. It is automatically updated only if the eyes reach the current bounds.

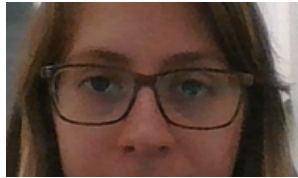


Figure 8: Section around eyes for motion detection

3.6 Asynchronous processing

Even for the size-reduced glass-section the RAGNet takes 190 ms to process a single frame. Therefore, we decided to move the RAGNet processing to a separate thread. The main thread supplies the RAGNet thread with the current frame and respective landmarks, as shown in Figure 1. The RAGNet side thread then processes the frame asynchronously and generates the reflection reduced image and the reflection mask. The reflection reduced image is then color corrected. The input images, output images and facial landmarks are stored in the cache using a ring buffer scheme.

3.7 Pose-based cache search

For each input frame the main thread searches for a fitting similar frame in the cache. The selection is based on similarity of the facial landmarks of the current and the cached frames. The search can be executed in 5 ms for 50 cache elements using this approach.

Excluding mouth and eyes, 35 out of the 68 facial landmarks are used for per-frame head pose encoding using a 2-column matrix, storing one position (x, y) per row (Equation 1). The dissimilarity d_i between the i th cached element's facial landmark matrix M_i and the current landmark matrix $M_{current}$ is determined by the Frobenius norm F of their difference (Equation 2). The cached element with the smallest d_i is selected, if it is below the threshold t_{norm} (Equation 3), which is an image size-normalized threshold with user-defined parameter t . A value of $t = 15$ was empirically found to yield a good trade-off between cache hit rate and visually pleasing output.

$$M = \begin{pmatrix} y_1 & x_1 \\ \vdots & \vdots \\ y_{35} & x_{35} \end{pmatrix} \quad (1)$$

$$d_i = F(M_{current} - M_i) \quad (2)$$

$$t_{norm} = t \cdot \frac{\text{image width}}{1000} \quad (3)$$

3.8 Cache-based reflection reduction

Mitigating the expensive, thus slow execution of RAGNet, cached results of preceding frames that were computed by RAGNet already, are now employed to reduce reflections on the current input image. As described above, we retrieve the data of the cached frame that is most similar to the current frame in terms of the detected head pose, assuming no significant changes in the background of the videoconference feed, i.e., the user's surrounding. To reduce the reflections in the current input image, we use the cached reflection mask, i.e., the difference between the cached input image and the cached reflection reduced image.

To account for slight differences between the head pose of the current and the cached frame, the reflection mask needs to be adjusted accordingly. To this end, the following three approaches for reflection mask adjustment were tested.

1. Homography transformation based on four facial landmarks: the outermost points along the eyebrows (17, 26) and the lower left and right parts of the chin (6, 10).
2. Affine transformation based on three facial landmarks: the outermost points along the eyebrows (17, 26) and the lowest point along the contour of the face at the middle of the chin (8).
3. Correlation: Application of the cached reflection mask at the location of highest correlation (normalized mean shifted cross correlation) between the glass-section of the cached frame and the current input image.

Since reflections are no fixed parts of glasses, but may instead change their position on the surface as the head moves, they do not necessarily move uniformly with the glasses. Thus, head motions may still result in some artifacts in the form of brightness mismatches along the edges of the cached reflection mask.

From the three tested approaches, correlation leads the fewest artifacts and is therefore suggested to be used by default. An exemplary result of the reflection reduction using correlation is shown in Figure 9.

4 EVALUATION

In the following we evaluate the different components of our proposed method.

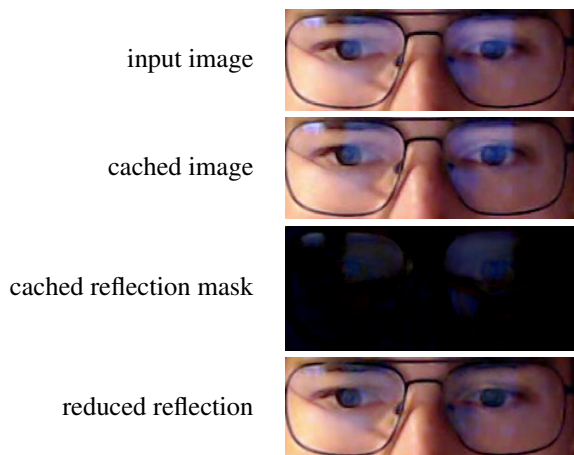


Figure 9: Cache-based reflection reduction

4.1 Glass-section detection

The face detection is evaluated on a test dataset of 2400 images. The test images originate from the recorded webcam streams of four different web meeting participants. The quality of the face detection is evaluated for scaled and unscaled input images in Table 1. The accuracy does hardly degrade for scaled images and is sufficiently high.

	Unscaled	Scaled
Scale factor	1	0,3125
TP ¹	2372 / 2400	2369 / 2400
Accuracy	98.83%	98.71%

Table 1: Face detection accuracy on scaled images

The glasses detection was optimized on a subset of the dataset. The quality of the algorithm was evaluated on the rest of the dataset. As shown in Table 2, the proposed glasses detection method performs similar to the reference method by Fernández et al. [FGUC15]. It should be noted, however, that different data sets were used for validation, because there is no reference implementation available for the comparison method.

	Training set	Test set	Reference method [FGUC15]
TP	392 / 397	1935 / 1971	2959 / 3000
FP ¹	0	2	-
FN ¹	5	34	-
Accuracy	98.74%	98.17%	98,65%

Table 2: Glasses detection accuracy

¹ TP: True Positives, FP: False Positives, FN: False Negatives

Since there is no reference glass-section cropping algorithm, our approach is validated against a previously manually selected image region. We use the excess image area and the missing image area relative to the manually selected area as metrics to determine the quality of the automatic glass-section cropping algorithm. The results are listed in Table 3.

	Training set	Test set
Images	400	2000
Excess area	14,21%	13,79%
Missing area	8,56%	9,55%

Table 3: Automatic eye region cropping

The overall relative error area is sufficiently small for the subsequent processing stages. The sum of errors is only slightly increasing from the training to the test set. This implies that the algorithm shows a good generalization and should be applicable to new previously unseen images.

4.2 RAGNet performance

The different retrained instances of the network are evaluated quantitatively by comparing their average PSNR and SSIM [HZ10]. The disjoint test data set consists of 1000 mixed stroboscopic and synthetic images. Table 4 shows both metrics for the generated reflection reduced output images.

Training set	Epochs	PSNR	SSIM
RAGNet original	150	15.24	0.731
Synthetic data	15	23.80	0.880
Synthetic + stroboscope data	55	28.86	0.935
Reflection-only + joint training	30/70	27.39	0.937

Table 4: Performance per training strategy

The network instance with reflection-only pre-training followed by full training, has the best average SSIM score and reaches the second highest average PSNR value. It is the only network computing a meaningful reflection mask for our scenario.

The network robustly detects reflections on the constrained input data and convincingly reduces reflections on single images. Even though very bright (clipped to white) reflections in input images result in visible artifacts, their appearance is still reduced noticeably.

Given the RAGNet (in synchronous mode) would run fast enough, it is only evaluated on individual frames without incorporating temporal consistency, resulting in a noticeable flickering. This directs towards future

research on, e.g., temporal low-pass filtering the reflection mask output or extending the RAGNet architecture to include recurrent layers for temporal context.

The reflection reduction works good on the constrained data set of similar test data. The model has problems generalizing on unseen footage. This limitation could clearly be overcome with a more diverse training set.

4.3 Motion detection

The motion detection was subjectively tested for plausibility. The estimated SSIM index correlates well with the present amount of motion, i.e., the SSIM index reaches values near one for non-moving persons.

4.4 Asynchronous processing

The temporal coherence and overall reflection removal quality was verified subjectively. The asynchronous processing introduces some additional flickering to the resulting video stream, caused by remaining differences between cached and current frames. While mismatches due to people's motion are limited to small offsets via motion detection, changes in lighting are implicitly compensated over time due to the ring-buffered cache.

Regarding the reflection mask adjustment, the homography approach yields mediocre results. Even with carefully selected landmarks there were some clearly visible remaining artifacts when applying the transformed cached reflection mask. Restricting the degrees of freedom by using affine transformations resulted in more consistent and therefore more pleasing results. Best results were achieved using the correlation approach, restricting the applied transformation even more, yielding the temporally most consistent results. This resulted in an overall visually more pleasing perception.

4.5 Execution time

The real time requirements require a strict optimization of the different components. All performance tests were performed using input videos with a resolution of 1920×1080 pixels on a system with an NVIDIA GTX 960 and an AMD RADEON VEGA 56. The glasses detection is running on the former while the RAGNet is running on the latter.

The largest performance improvement could be achieved by executing the RAGNet asynchronously. The reduction of the input resolution for the glasses detection and the movement detection resulted in further performance improvements. The mean execution times for processing a single frame, averaged over 200 frames, is displayed in Table 5. Finally, applying some common optimizations throughout the pipeline, such as reducing the number of image copy operations, yielded a final frame rate of 31.25 Hz.

Optimizations	Execution time [ms]
Synchronous	410
Asynchronous	199
Asynchronous & scaled	38
Further optimization	32

Table 5: Average processing time per frame

The composition of the frame processing time for a single exemplary frame is shown in Table 6.

Processing Step	Execution time [ms]
Read Frame	1
Glasses detection	20
Motion detection	6
Cache search	4
Transfer to current frame	2

Table 6: Processing time per system component

5 DISCUSSION

The model generally strongly depends on the input video stream. The best results are achieved under good lighting conditions and for reflections in the upper half of the glasses.

Generalizability. While using comparatively small data sets, like in this work, typically implies little generalizability, building upon the far more diversely pre-trained RAGNet mitigates this weak point for our approach. It should therefore also be possible to also reduce reflections from light sources other than screens, e.g. ceiling lamps, and even under different lighting situations.

Moreover, the restriction of the processing to the glass-section should further increase generalizability, as the network does not need to learn (to ignore) arbitrary environments.

Limitations. Reflections which directly occlude the eyes sometimes result in worse reflection removal performance with stronger artifacts, as shown in Figure 10.

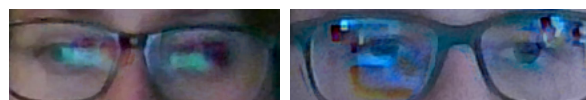


Figure 10: Artifacts for reflections covering the eyes

While strong variations in illumination will most likely not break the approach, they might reduce the effectiveness, resulting in, e.g., brightness or color mismatches.

Since both limitations arise from the limited data set, it is reasonable to assume that the proposed system can overcome them by extending the training to a larger and more diverse data set, which we leave for future work with a focus on robustness.

6 CONCLUSION

This paper presents an approach to reduce reflections on glasses in real-time. We showed that the RAGNet neural network can be arranged in an appropriate pipeline to convincingly reduce reflections on glasses. For application in live videoconference scenarios, we achieved real-time capability by reducing the network input size using the newly introduced glass-section detection and the proposed asynchronous processing scheme. Moreover, temporal consistency is strengthened via robust motion detection and color transfer.

While the goal of complete reflection removal was not achieved, the synchronous mode would result in visually more pleasing reflection removal on selected inputs, but is not real-time capable. The real-time capable asynchronous mode introduces some artifacts and flickering. Furthermore, some aspects of the implementation still offer potential for improvement, e.g., for multiple persons or handling of the remaining error cases, such as reflections largely occluding the eyes. The method is currently still very resource demanding, motivating further optimization, e.g., via motion compensation for cached frames.

Also beyond videoconferences, the proposed method could be a helpful tool for preprocessing videos in applications that use eye tracking or emotion analysis. The method could also be reduced in scope, to be used as a reflection detection.

7 ACKNOWLEDGMENTS

Partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 491805996 / GR 5932/1-1.

8 REFERENCES

- [AST⁺22] Amanlou, A., Suratgar, A. A., Tavoosi, J., Mohammadzadeh, A., and Mosavi, A. Single-image reflection removal using deep learning: A systematic review. *IEEE Access*, 2022.
- [FGUC15] Fernández, A., García, R., Usamentiaga, R., and Casado, R. Glasses detection on real images based on robust alignment. *Machine Vision and Applications*, 26(4):519–531, May 2015.
- [FYH⁺17] Fan, Q., Yang, J., Hua, G., Chen, B., and Wipf, D. A generic deep architecture for single image reflection removal and image smoothing. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3258–3267, 2017.
- [HZ10] Hore, A. and Ziou, D. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010.
- [JBAB00] Jiang, X., Binkert, M., Achermann, B., and Bunke, H. Towards detection of glasses in facial images. *Pattern Analysis & Applications*, 3:9–18, 2000.
- [Kin09] King, D. E. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. <http://dlib.net>.
- [KS14] Kazemi, V. and Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In *IEEE conference on computer vision and pattern recognition*, pages 1867–1874, 2014.
- [LCL21] Li, T., Chan, Y.-H., and Lun, D. P. K. Improved Multiple-Image-Based Reflection Removal Algorithm Using Deep Neural Networks. *IEEE Transactions on Image Processing*, 30:68–79, 2021.
- [LLY⁺23] Li, Y., Liu, M., Yi, Y., Li, Q., Ren, D., and Zuo, W. Two-stage single image reflection removal with reflection-aware guidance. *Applied Intelligence*, pages 1–16, 2023. <https://github.com/liyuks/RAGNet>.
- [NVI] NVIDIA Corporation. CUDA. <https://developer.nvidia.com/cuda-toolkit>. Visited on 19.01.2022.
- [PSB⁺21] Prasad, B. H. P., S, G. R. K., Boregowda, L. R., Mitra, K., and Chowdhury, S. V-desirr: Very fast deep embedded single image reflection removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2390–2399, October 2021.
- [RAGS01] Reinhard, E., Ashikhmin, M., Gooch, B., and Shirley, P. Color Transfer between Images. *IEEE Computer Graphics and Applications*, 21:34–41, October 2001.
- [SAT⁺16] Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, 47:3–18, 2016.
- [Sid21] Siddharth Mandgi. Real-time Glasses Detection. <https://medium.com/mlearning-ai/glasses-detection-opencv-dlib-bf4cd50856da>, September 2021. Visited on 19.01.2022.
- [Tia19] Tianxing Wu. Real-time Glasses Detection. <https://github.com/TianxingWu/realtime-glasses-detection>, November 2019. Visited on 19.01.2022.
- [WBSS04] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [WSL⁺21] Wan, R., Shi, B., Li, H., Duan, L.-Y., and Kot, A. C. Face Image Reflection Removal. *International Journal of Computer Vision*, 129(2):385–399, February 2021.

The Method of Mixed States for Interactive Editing of Big Point Clouds

Werner Benger

Airborne HydroMapping GmbH

A-6020 Innsbruck, Austria

w.benger@ahm.co.at

Center for Computation & Technology

Louisiana State University

Baton Rouge, LA-70803

Anca Voicu

Prowasser

C-tin Brancoveanu-64, Timisoara, Romania

anca.voicu@prowasser.ro

Ramona Baran

Airborne HydroMapping GmbH

A-6020 Innsbruck, Austria

r.baran@ahm.co.at

Loredana Gonciulea

Prowasser

C-tin Brancoveanu-64, Timisoara, Romania

loredana.gonciulea@prowasser.ro

Cosmin Barna

Prowasser

C-tin Brancoveanu-64, Timisoara, Romania

cosmin.barna@prowasser.ro

Frank Steinbacher

Airborne HydroMapping GmbH

A-6020 Innsbruck, Austria

f.steinbacher@ahm.co.at

ABSTRACT

We present a novel methodological approach for the interactive editing of big point clouds. Based on the mathematics of fiber bundles, the proposed approach to model a data structure that is efficient for visualization, modification and I/O including an unlimited multi-level set of editing states useful for expressing and maintaining multiple undo histories. Backed by HDF5 as high performance file format, this data structure naturally allows persistent storage for the history of modification actions, an unique new feature of our approach. The challenges of visually based manual editing of big point clouds are discussed and a proper rendering solution is presented. The implemented solution and its features as consequences of the underlying methodology are compared with two major mainstream applications providing point-cloud editing tools as well.

Keywords: point clouds, interaction, classification, data editing, fiber bundle data model, undo history

1 INTRODUCTION

Correct and accurate classification is an essential step in the LiDAR (Light Detection And Ranging) point cloud processing. More specifically, the classification of ALB (Airborne Laser Bathymetry) data focuses primarily on the definition of terrain and water surface points. The latter is required for the final refraction

and runtime correction of points lying beneath the water surface. According to the summary of [LG17], morphological (e.g., [Sit01, MWSCC09]) and surface (e.g., [KP97, LKS00]) based filters and their extensions/variants are among others utilized for the terrain detection. All described approaches solely represent automatically calculated classification results. Considering 3D data and their interpretation, recent algorithms cannot replace the human capability of cognitive abstraction and anticipation. Thus, a manual inspection and editing procedure of automatically classified point clouds is crucial in order to correct erroneous classification results if required, and to ensure data quality. The quality of automated ALB point classification results strongly depends on the overall raw data quality, which is mainly influenced by weather and water conditions during the survey, and on the general morphological appearance of a project area. A high humidity during an ALB survey results in increased flaw echo detection, and such noisy points need to be separated from the actual points of interest (Fig. 1). Further factors reducing the quality of the automated classification are:

- the terrain complexity, e.g. high mountainous relief vs. uniform flatlands [CZW13];
- a dense vegetation canopy with shadowing effects reducing the actual terrain coverage [WSB⁺12];
- high water turbidity limiting water penetration [Man20];
- white water areas impeding water-ground detection;
- water-bottom material consisting of substrates with increased light absorbing characteristics (e.g., underwater plants, organic matter mixed with bed material) hampering water-ground detection[Man22].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Plzen, Czech Republic.
Copyright UNION Agency – Science Press

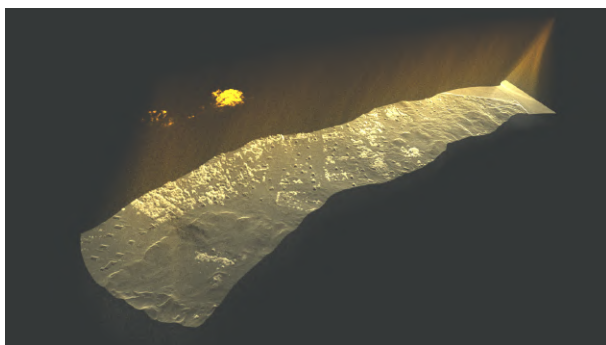


Figure 1: ALB scan strip with flaw echoes (yellow to orange) and relevant point data (white) that can be clearly separated by visual inspection.

Thus, the manual correction of the classification can be highly time consuming depending on these conditions as well as the spatial extent of the project area. To minimize the manual editing effort on one hand, recent software development is focused on the improvement of the automatic classification of ALB data by incorporating and combining further raw data attributes and derived geometric parameters into the classification process (e.g. [SDB⁺21]), or utilizing machine/deep learning approaches for specific classification purposes (e.g. [HEA⁺21]). On the other hand, an efficient manual point-cloud editing toolkit together with a fast and flexible point-cloud visualization and navigation regardless of file size are important prerequisites for minimizing manual correction efforts on classification results. Manual editing tools are often standard components of various available LiDAR software packages, such as Terrasolid by Terrasolid Ltd., RiProcess by RIEGL LMS, or LAsTools by rapidlasso GmbH. In this paper, however, we present the manual point cloud editing tools integrated in HydroVish, and how we optimize therewith the general ALB point-cloud processing.

This article's structure is: Sec. 2 reviews the foundations of the visualization environment, the data organization model and the underlying file format; Sec. 3 discusses the aspects of user-friendly and versatile selection of regions within a point cloud; Sec. 4 focuses on persistent storage of an unlimited undo history for big datasets while sustaining interactive performance; finally, section Sec. 5 presents a comparison of our implementation with two existing external applications.

2 PREREQUISITES

The work as presented here is implemented in the Vish Visualization Shell [BRH07], a general-purpose framework for visualization algorithms. Its specialization to bathymetric datasets, HydroVISH, is used in production by AHM GmbH¹ for large point clouds acquired by high-resolution airborne observations.

¹ www.ahm.co.at

2.1 Visualization Pipeline & Networks

Haber & McNabb [HM90] described a conceptual visualization process with three basic stages, transforming raw data into displayable images. These steps occur in most visualization processes and aim to convert data into information while maintaining the integrity of the content with best accuracy.

Modular visualization environments turn these stages into component parts that can be connected at runtime such to allow customization for a particular task. This type of visualization system is very widely used by the scientific community for its flexibility and programmability. The connections between visualization components may be set up via scripts or graphically via a user interface that allows to build a network representing the data flow. This approach of “visual programming” does not require any coding capabilities and is quickly intuitive to end-users. Beyond the underlying data flow also the control flow — the interaction of a user interface with steering parameters influencing the data processing — is important for flexibility and configurable user experience. For instance, the ability to couple two buttons with the same hotkey or mouse click per user-driven configuration allows for personalized preferences on how editing appears most convenient for a particular experience.

2.2 The F5 Fiber Bundle Data Model

The mathematics of fiber bundles provides a framework to model data for scientific visualization [BP89, Ben04]. This concept considers data sets based on the properties of their “base space” versus its “fiber space”. A fiber bundle is basically a set of points with neighborhood information and equally-sized data sets attached to each such point. For instance, a multidimensional homogeneous array constitutes a fiber bundle in the mathematical sense. The F5 data model [Ben09] builds upon this concept by grouping all dataset properties with the same number of elements, thereby defining “index spaces”. Any suitable dataset is dissected into a hierarchy of five levels according to its properties:

1. *Time* (temporal slicing),
2. *Grid* (geometrical entity),
3. *Skeleton* (topological property),
4. *Representation* (coordinates, relationships) and
5. *Field* (binary representations of numerical values).

An additional optional sixth level allows to split up a *Field* into a set of named *fragments* (chunks of data contiguous in memory) for easier handling of large datasets. Each of these fragments may come with its own size, but all fragments of the same name must be of identical size within the same *Skeleton*.

The bottom line is the ability to handle datasets based on explicit properties instead of a set of implicit built-in assumptions: the model answers the question “how is it?” about a dataset instead of the question “what is it?”. Algorithms need to be implemented in a way such to only request relevant properties of a dataset rather than the “type” of a dataset. This approach allows to cover a wide range of data categories using the same software infrastructure. Point clouds are a rather simple subset of this general framework, but by virtue of the fiber bundle data model the presented editing framework immediately applies also to other data types such as triangular meshes or line sets.

2.3 The HDF5 File Format

The hierarchical file format v.5 (HDF5 [HDF23]) is a general-purpose, self-descriptive open file format designed for high performance computing. It resembles a “file system within a file” with many features beyond an actual file system. For instance, structured data such as multidimensional arrays and user-defined compound types are supported natively. A wide range of data compression algorithms is available via a plugin system which allows optimization for different application domains and scenarios. The hierarchical data organization as used for the F5 data model from section 2.2 is very suitable to be directly mapped onto an HDF5 file.

3 ON-SCREEN POINT SELECTION

The point selection tool allows to draw an outline on the screen which is then projected into 3D space for selecting actual points. It provides multiple functionalities:

1. Polygon shape: each mouse click adds another point, points are connected via straight lines.
2. Free-Hand shape: While the mouse button is pressed, points are added to the shape while the mouse moves.
3. The shape can be stored and loaded, either as part of the visualization network which contains the state of all parameters of the current setup, or explicitly as a polygonal set of points in various file formats.
4. The shape can be dragged along the screen, similar to a “brush” in Photoshop™.

The drawing tool provides output actions, which by utilizing the capabilities of the visualization network, can be configured to perform different data editing actions according to the choice of the user:

1. Without configuration, drawing an on-screen selection requires an explicit button to be clicked to select points. This mode is useful if the user wants to carefully draw some shape first before applying a selection.

2. The selection can be performed immediately at each change of the shape, working for both adding more points to the outline shape (polygon mode, free-hand mode), as well as for dragging a fixed shape on the screen (“brush mode”).
3. The selection can be performed when the outline drawing is “closed”, i.e. the “last” point of an outline has been drawn and the outline is cleared such to start a new draw operation. This is usually done with a modified mouse-click, such as alt-mouse or right-mouse such to differentiate this operation from the shape drawing. This mode resembles the painting of polygons or free-hand forms on a white board, but in this case selects points within the point cloud.

3.1 Masked Editing via “Dots”

An additional level of security is given by displaying the what-if of a data editing operation, i.e. before actually performing the data modification immediately. This mode of editing is similar to utilizing a “selection” in Photoshop™ in order to limit some filter operation on a photo to this selected region. Similarly here we first mark - and visually enhance - the set of points that are intended for subsequent modification. This mask of points can be modified, like adding or removing parts of a selection, before an “apply” action (triggered by a button in the GUI, a hotkey, or a certain mouse even) modifies the actual data.

Per-point color attributes are sensitive pieces of information that already convey important properties such as RGB photographic data, height information, labels information (as elaborated in Sec. 3.2), etc. or an combination of those. In particular we display dots not as singular pixels on the screen, but via extended geometries resembling little spheres. We call these “dots” to distinguish them from single-pixel display methods. This “dot” display allows for highly precise study of fine details in the point cloud with intuitive depth perception; this is not possible by displaying all data points as single pixels. We explored several methods on to display markers on these “dots” without impacting their ability display of basis attributes, as demonstrated in Fig. 2. Some of the possible choices may be due to personal, aesthetic reasons, but there are also constraints as the choice influences the appearance when zooming out: as points shrink in screen-space, the marker information may get lost once point size approaches a single pixel - which is unacceptable when the conveying the selection information is important. Thus, special care must be taken for the modified rendering information, for instance using a view-distance-dependent fraction of the marker information versus basis information such that the marker information becomes dominant on overviews, as demonstrated in Fig. 3. In order to compensate for small dots (in screen-space) such that these

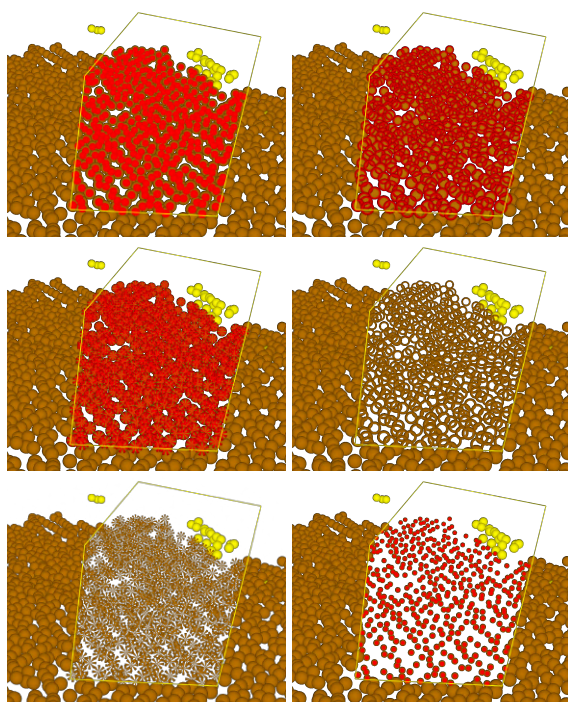


Figure 2: Displaying selections as per-point attribute independent of underlying colorization: colored inner core, colored outer rim, colored sections, transparent core, transparent sections, size adjustment.

will be rendered more like non-circular dots in an effort to counter anti-aliasing (which would hide those) we employ OpenGL's `fwidth()` function to consider the screen-space derivative of a dot's texture. In the GLSL shader this compensation works as follows:

```
in vec2 SplatTexture;
in float Mask;
in vec4 MaskColor, PointBaseColor;
uniform float Threshold, MaskRadius;

float R2, T, dR_dPixel;

R2 = dot(SplatTexture, SplatTexture);
dR_dPixel = fwidth(R2);
T = 1.0 - R2;
T += .5 * dR_dPixel * dR_dPixel;
T = clamp(T, -1.0, 1.0);

if (T < 0.0) discard; // Make dots round.

if (Mask > Threshold &&
    R2 + dR_dPixel > MaskRadius)
    color = MaskColor;
else
    color = PointBaseColor;
```

`MaskColor` is the color for the marked regions in modification points colored by the `PointBaseColor` (which may be true RGB colors from observations, height maps, intensity maps or any other color attribute).

Global parameter `MaskRadius` allows to fine-tune the visibility of the mask, a value of 0.5 values masking and colorization equally. Per-point attribute `Mask` defines a value between 0.0 and 1.0 specifying the strength of the mask; for a boolean mask, those values will be either 0.0 or 1.0; global parameter `Threshold` determines at which strength the mask should be displayed at all (0.5 per default). Selections can therefore be “hard” or “fuzzy”, which can be useful when e.g. assigning RGB color values in an airbrush-like manner.

OpenGL point sprites receive texture coordinates in variable `SplatTexture` for each pixel in the range $[-1, +1] \times [-1, +1]$. The code computes a radial distance from these and discards all fragments beyond a constant distance, effectively creating round dots on the screen from the rectangular point sprite. This radius is adjusted dependent on the size of the point sprite on screen such that smaller sprites have less pixels cut off, thus appear larger in relation. The same mechanism is applied to the section of the dot that is colored with the mask indicator - thus smaller sprites are weighted stronger and appear more prominently. The size of the sprites per point is determined by the previous geometry shader (code not shown here) based on view distance; thus more distant dots that got marked remain visible when zooming out.

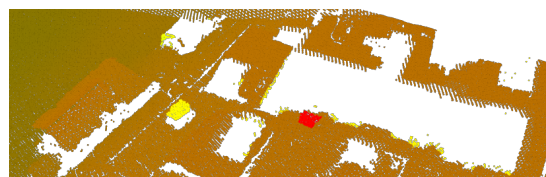


Figure 3: A good point-wise selection information must display information also when “zooming out” in overview mode.

Just making points transparent does not work easily within a three-dimensional scene as this would require depth-sorting of points or an equivalent technique suitable for millions of objects - very likely impacting performance, so we favored to use techniques without any such overhead. For editing a photo, a mask can be displayed by some two-dimensional overlay, but for editing three-dimensional point clouds a simple per-point overlay is insufficient because points in the foreground may hide points in the background - a situation that cannot occur when editing photos. A possible way to address the visual clutter is achieved by shifting marked points in screen space towards the observer, thereby “boosting” their visibility over other points (similar to OpenGL's `glPolygonOffset()` function), as demonstrated in Fig. 4.

3.2 Label-Constrained Editing

Labelling points by assigning integer numbers (representing specific meanings) to each point is the result

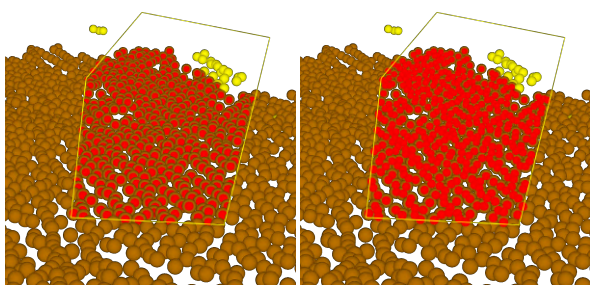


Figure 4: “Visibility Boost”: enhance the visibility of marked dots to avoid visual clutter.

of a classification process to identify objects in raw data. Manual correction of automatic pre-classification is enhancing accuracy and providing the essential input data for refined training of machine learning algorithms. With such pre-classified data sets only some points usually need to be re-labeled; it is thus desirable to define sets of labels that should be subject to editing whereas other sets of labeled points shall remain unmodifiable, as demonstrated in Fig. 5.

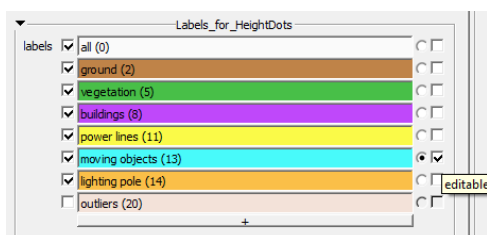


Figure 5: GUI element to control label-constrained editing: visibility checkbox, description, colorization, integer value, selected assignator, write-property.

3.3 Field-Based Editing

It is rather straightforward to also consider any data field – i.e., point attributes – for constrained editing, including per-point scalar values with global or local threshold or range constraints such as height information (“only allow modification of points beyond 834m elevation above sea level”) or point neighborhood information such as planarity [RBC⁺ 12] (“only allow modification of points that reside within a plane given a minimal deviation tolerance”).

Geometry-Constrained Editing Geometry is per-point coordinate information and can be constrained by an axis-aligned bounding box, or any other formula implementing an inside/outside check of a volumetric region. In practice, having the ability to define an axis-aligned bounding box, as demonstrated in Fig. 6, handles the systematic manual traversal of an extended data volume. Such a selected geometric region can, but does not necessarily have to, correspond to the geometric properties of data fragments. An additional geometric constraint can be defined by specifying a depth range

as seen from the observer, thus effectively defining a cross-section for editing.

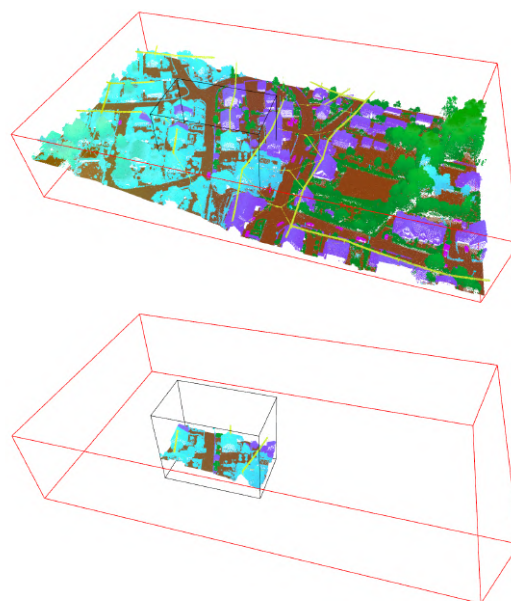


Figure 6: Example of a bigger point cloud (red bounding box, top) and a smaller tile of this point cloud (blue bounding box, bottom) in HydroVish. Editing can be restricted to arbitrary tile shapes with a shiftable box selection allowing to systematically traverse the entire volume in regular tiles.

Editing Coordinates Coordinates are another data field that can be subject to an editing operation, thereby enabling modification of a point cloud’s geometry, just as in CAD applications. Furthermore, as long as the number of point remains constant, also additional properties of an underlying geometric object are unaffected: if the connectivity information within a triangular mesh is not touched, then the point cloud editor is directly applicable to editing surface geometries as well.

Editing Colors Color information are RGB values per point; an editor may modify such attributes instead of integer labels, resulting in the same functionality of a Photoshop™ painting with a brush on a photo - but now painting on a three-dimensional point cloud, or even triangular mesh. However, in contrast to modifying label information a smooth transition from current value to new value may be desirable in this case, i.e. using a “soft” instead of a “hard” brush. This feature can be realized by computing the distance from each point in a selection to the boundary of the defining brush shape and weighting the resulting color accordingly.

4 UNDO OPERATIONS

The common approach to undo actions is a global history of states that allows to go “backward”, and sometimes also to go “forward” (redo operation). Undoing an undo action is equivalent to a redo action.

4.1 Three Level Handling

In our context undo/redo operations act on

1. drawing a free-style shape on-screen,
2. applying an on-screen selection to a mask,
3. modifying point attributes by a given mask.

All these operations are inherently independent and thus have their own undo history. It may be an option to merge all actions with a time stamp into a global history such to conform to the “standard” approach. However, as the application is capable to also perform more than single editing approach, such as editing entirely independent data sets within the same view, mixing such histories would disable the ability to treat each editing independently. Even with a single data set, the ability to undo data modifications immediately, without bothering to go through the masking and lasso history, is beneficial in speeding up practical work. Both the masking and lasso history are hardly ever needed. The three types of undo are useful with different relevance depending on the respective editing scenario.

4.2 Fragmented Data & Mixed States

To allow undo and redo operations a history of data states must be remembered. The “brute-force” approach would be to always keep a copy of the entire data set after each operation. While this is the simplest way to implement a history, it is undesirable for large data due to performance and memory requirements. A more elaborated approach is to remember only the differences between two data modification states under the assumption that the majority of data remains unchanged. However, this comes with an additional computational effort, and computing differences of the entity of a big data set is inefficient as well.

In our context of the F5 data model as presented in Sec. 2.2 all data are split up into fragments such that each fragment contains only a few million points (out of hundreds or thousands of millions in its entity). Only those fragments that are visible as part of an editing operation are loaded into RAM (and eventually the GPU), all others remain on disk. Consequently, a history of data modification operations only needs to take those data fragments into account that are affected by each operation. An option is then to keep a list of the differences of fragments that are modified at each operation. However, as computing and applying differences of millions of points does take noticeable computation time, we decided against using differences, but rather to keep copies of the involved data fragments before modification. An undo operation can then be implemented by merely switching pointers to previous data fragments, avoiding any copy or computational operation for millions of points and/or their attributes.

For instance, let us denote an editing action as the transformation of a set of fragments $\{A, B, C, D, E\}$ to another state $\{A', B', C', D', E'\}$ (out of possibly many more). A first editing operation modifies fragments $\{A, B\}$, a second editing operation modifies fragments $\{C, D, E\}$. To cover this situation of two operations, the undo history must remember three states, denoted hereby as ①, ②, ③:

①	A	B			
②	A'	B'	C	D	E
③			C'	D'	E'

Each such state is stored as an “undo field” in the fiber bundle data model as introduced in Sec. 2.2 (Fig. 7 shows the structure as stored in an HDF5 file). When editing a fragment, also those fragments from the previous undo field in the current undo field have to be copied, even when they have not been edited. This means that an undo field contains both un-edited as well as edited data fragments, thus is a mix of edited and un-edited states. In the example given here, the undo field ② is largest as it contains five data fragments, the other two undo fields ① and ③ require less memory. An undo operation needs to replace fragments as stored in ③ by the respective fragments as stored in ② (but not all fragments in ②!). A subsequent undo operation then needs to replace fragments in undo field ② by the fragments stored in ①, if those exist. Thus, at each such operation, only the actually modified fragments are changed, same as when using differences: even though undo field ② contains five fragments, only two or three are replaced in this scenario at each undo operation.

4.3 I/O - Persistent Editing History

The usual approach of handling data is to

- load data from disk to RAM;
- if out of memory, let the operating system swap RAM data to disk, or utilize internal temporary files to store RAM information on disk;
- once data modification is finished, save the resulting data from RAM to exportable file formats;
- during data export, possibly load data from swap space (OS-provided or temporary files).

Whereas, via using HDF5 and the fiber bundle data model this functionality is simplified significantly:

- the file is parsed for metadata, only those are loaded;
- data fragments are only loaded when needed;

- modified data fragments are stored to the HDF5 if the application runs out of memory or terminates.

Hereby the HDF5 file serves as disk-image of the in-RAM data structure with no need of computationally intensive data transformations. This functionality mimics a memory-mapped file but is much more flexible since the data items can be extended dynamically at many “branches” of the hierarchically stored data. Via HDF5 highly performing compression filters such as LZ4 or ZSTD are available, such that disk space usage is minimal while I/O performance is higher than reading uncompressed data. Some of these high performance compression filters even claim [Alt10, Alt23] to be designed to unpack data faster than a traditional `memcpy()` operation could load them into CPU cache. In consequence, there is no need for “swap files” or “temporary files”, and even an explicit “save” operation becomes superfluous: any data modification is directly mapped to the underlying HDF5 file in an end-user ready file format available for further data processing.

```
/t=0.0/Lake/Points/UTM32N/Labels/Frag[8x16x0] Dataset (273499)
/t=0.0/Lake/Points/UTM32N/Labels/Frag[8x17x0] Dataset (75708)
/t=0.0/Lake/Points/UTM32N/Labels/Frag[9x15x0] Dataset (108860)
/t=0.0/Lake/Points/UTM32N/Labels/Frag[9x16x0] Dataset (980038)
/t=0.0/Lake/Points/UTM32N/Labels/Frag[9x17x0] Dataset (998418)
/t=0.0/Lake/Points/UTM32N/Labels/Frag[9x18x0] Dataset (468318)
/t=0.0/Lake/Points/UTM32N/Labels/Frag[9x19x0] Dataset (901)
/t=0.0/Lake/Points/UTM32N/Labels/Frag[10x15x0] Dataset (143972)
/t=0.0/Lake/Points/UTM32N/Labels/Frag[10x16x0] Dataset (921022)
/t=0.0/Lake/Points/UTM32N/Labels/Frag[10x17x0] Dataset (1004095)
/t=0.0/Lake/Points/UTM32N/Labels/Frag[10x18x0] Dataset (857482)
/t=0.0/Lake/Points/UTM32N/Labels/Frag[10x19x0] Dataset (255736)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0000/Frag[10x19x0] Dataset (255736)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0001/Frag[10x19x0] Dataset (255736)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0002/Frag[10x19x0] Dataset (255736)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0003/Frag[10x19x0] Dataset (255736)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0004/Frag[10x19x0] Dataset (255736)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0005/Frag[10x19x0] Dataset (255736)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0006/Frag[10x19x0] Dataset (1004095)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0007/Frag[10x19x0] Dataset (255736)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0008/Frag[10x19x0] Dataset (1004095)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0009/Frag[10x19x0] Dataset (255736)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0010/Frag[10x19x0] Dataset (980038)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0011/Frag[10x19x0] Dataset (998418)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0012/Frag[10x19x0] Dataset (980038)
/t=0.0/Lake/Points/UTM32N/Labels.Undo-0013/Frag[10x19x0] Dataset (998418)
/t=0.0/Lake/Points/UTM32N/Mask/Frag[10x18x0] Dataset (857482)
/t=0.0/Lake/Points/UTM32N/Mask/Frag[10x19x0] Dataset (255736)
/t=0.0/Lake/Points/UTM32N/Mask.Undo-0000/Frag[10x19x0] Dataset (255736)
/t=0.0/Lake/Points/UTM32N/Mask.Undo-0001/Frag[10x19x0] Dataset (255736)
```

Figure 7: Partial listing of an HDF5 file in F5 layout as described in Sec. 2.2 containing both “labels” information and undo history of a fragmented point cloud using the HDF5 standard tool “h5ls”. Note that fragments of the same name are of identical size.

The undo history as presented in Sec. 4.2 is stored as *Fields* in the fiber bundle model (as introduced in Sec. 2.2) and therefore subject to I/O like all other data fields. Consequently the undo history is persistently stored in an HDF5 file upon any data modification action and preserved when the application is restarted. Any data modification steps can be replayed days, weeks, years later, based on an HDF5 file containing the full history information. Fig. 7 demonstrates the effective partial structure of such an HDF5 file describing a point cloud with labels, undo information for labels, masking for labels and undo information for

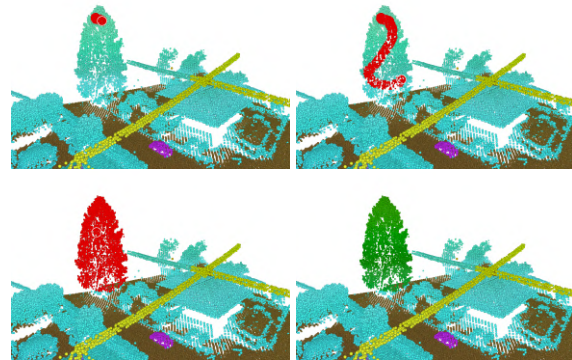


Figure 8: Example of free-style point selection (red points) and final classification assigned (green points).

masking, based on the previously mentioned five-level hierarchical data organization. The length of the history is only limited by available disk space. A length of 1024 is already overkill in practical applications, using a length of 8 turned out to be sufficient to cover an active editing process.

5 RESULTS

In the following, we compare our implementation for editing point-clouds with that provided by two other software solutions. Similar to our software (denoted as [3]), one of the two software solutions is a general and widely used software toolkit for point clouds (denoted as [1]) allowing to import datasets from various origins including LAS format support. The other one is more specific and represents a software solution provided by a sensor manufacturer (denoted as [2]). We intentionally omit the actual name of the respective software packages to avoid marketing issues. The key points of the comparison are summarized in Table 1.

The comparisons were performed using a desktop PC equipped with an AMD Ryzen 7 3700X 8-Core Processor, 3.59 GHz, 64.0 GB RAM, graphic card NVIDIA GeForce RTX 3070 running under Windows 10 Pro.

5.1 Manual Selection Options

All three software solutions provide similar point selection options but with differences in their actual details such as naming. A standard selection tool is the polygon shape. The polygon shape selection must be performed vertex by vertex, and must be closed manually at the same place where the selection was started ([1]), or is closed automatically by simple right mouse-button click ([2]) respectively is already closed from the beginning ([3]). Further basic selection modes consist of rectangular selection mode (e.g. [2]), line selection mode (e.g. [1]), as well as a free-style selection mode ([1], [2], [3]; Fig. 8).

5.2 Subdivision of Big Point Clouds

To facilitate a quick manual editing progress as well as maintaining an overview of the editing progress especially in big point clouds, it is highly beneficial that an entire dataset can be subdivided into smaller pieces that can be edited separately from each other. This opportunity is provided in [1] and [3] but missing in [2]. For [3], Fig. 6 shows an example of a small tile (bottom image) resulting from the subtiling of a larger point cloud (top image). In Fig. 9, the active selection progress (red points) in the small tile from Fig. 6 is indicated.

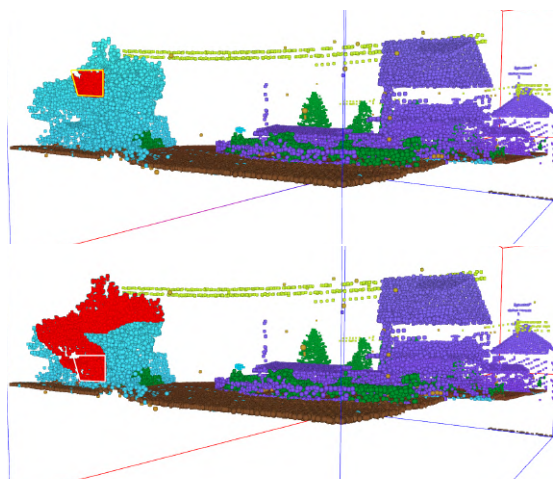


Figure 9: Active polygonal point selection progress (red points) in point cloud tile from Fig. 6.

5.3 Manual Classification Assignment

Once points were selected from a point cloud, the classification has to be assigned (Fig. 8). In one case, this is done immediately after the selection is finished ([1]). In the other case, the selection can still be edited further, e.g. expanding the selection or deselecting points ([2] and [3]). The classification is then assigned in a separate step by the user using a hotkey or button click in the respective GUI ([2] and [3]). All three software tools provide predefined classification ranges as well as the opportunity to introduce new classes according to the user's need. After assigning a class to a certain point selection, this selection is no longer maintained in all three software applications. In case of a mistakenly class assignment, [3] provides the opportunity of undo and redo operations at different levels, which are stored in the corresponding data file, and are still accessible after classification process is finished. In [1], single or multiple mistakenly class assignment(s) can be undone back in time, but these corrections are not accessible anymore after finalized classification. In [2], no undo operation is available to correct a mistakenly class assignment. Here, a renewed point selection and class assignment is required.

5.4 Display and Navigation

To facilitate the manual editing process of a point cloud, it is often required that only specific point class(es) are displayed and the point selection can be restricted to certain class(es). This is possible in all three software solutions evaluated here. In [2] it is further possible to restrict the display of points to a specified previous point selection. In [3], it is possible to simply set a certain cross-section depth and navigate back and forth through a point cloud in any desired direction just by mouse usage. In this case, points outside the defined cross-section depth are not displayed as well as not selectable for editing. In [1], a similar 3D navigation in cross-sectional view is possible but requiring a few more button clicks for specifying the navigation progress (Fig. 10). In contrast, in [2] it is only possible to display a cross-section of a certain depth throughout a point cloud based on an interactive 3D bounding box definition (Fig. 11 middle). All points inside this box are displayed afterwards (Fig. 11 bottom). For the user's orientation in a point cloud dataset, [2] provides separated 2D and 3D views. The zoom level of the 2D view defines the display area of the point cloud in 3D (Fig. 11). For re-sizing the 3D point cloud display, one need to adjust the zoom level in 2D first. Both views can be shown in parallel. In [1], the point cloud is usually displayed in several parallel 3D views, e.g. overview of entire dataset and detailed view from point cloud section for editing purpose (Fig. 10). The size of the views can be adjusted to the user's specific needs. In contrast to [1] and [2], we provide a single, quickly navigable 3D view of an entire point cloud in [3] (Fig. 12).

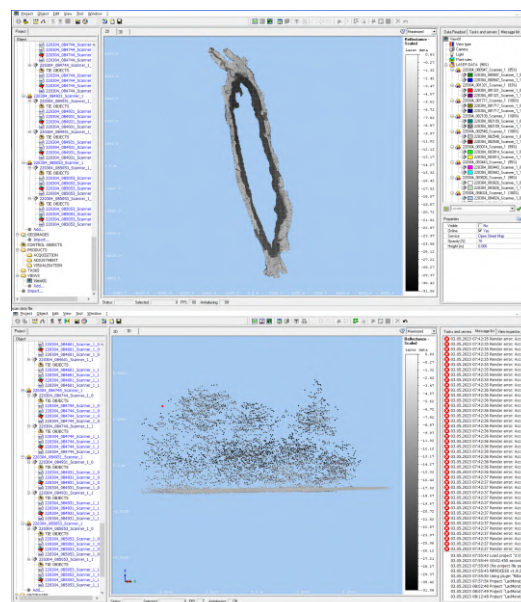


Figure 11: Solution [2]: GUI / view of an entire point cloud (top) and extracted cross-section view (bottom).

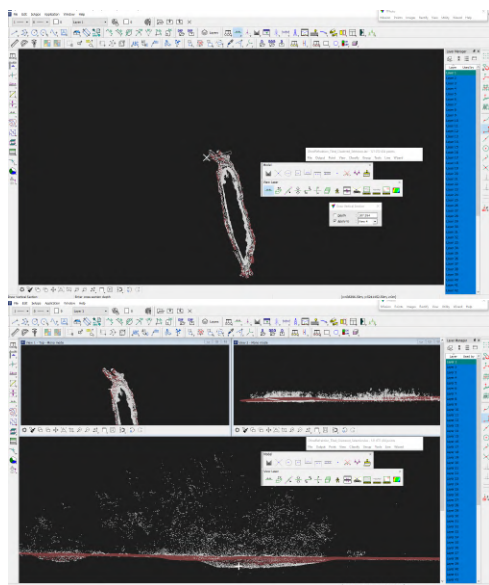


Figure 10: Solution [1]: GUI and 3D overview of an entire point cloud with indicated location of cross-sectional view (top). 3D views including detailed and cross-sectional view from area of interest (bottom).

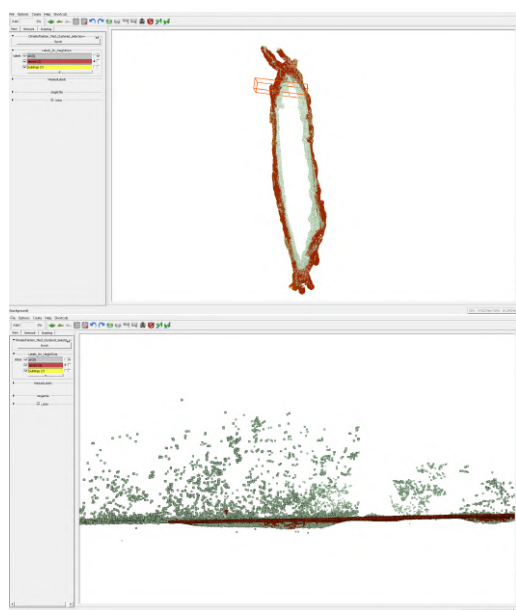


Figure 12: Solution [3] (ours): GUI and 3D view of an entire point cloud with marked sub-region (top) and cross-sectional view of sub-dataset (bottom).

6 CONCLUSION

We presented a systematic approach to allow interactive editing of big point clouds, and provided a comparison of our implementation with existing solutions. The presented methodology is based on the mathematics of fiber bundles and extends very naturally to various application scenarios, while providing high flexi-

bility and performance at the same time. The ability to store the editing history persistently in a data file is an unique functionality offering high security on data management and modification accountability. It further supports data quality management, e.g. tracking classification errors in the editing history which became obvious during a later processing step such as terrain modeling out of classified terrain points. Manual editing of a point cloud requires combining multiple user aspects to be timely efficient. The flexible selection and class assignment modi in combination with the three-level, preservable undo/redo options of point editing stages as well as the 3D navigation and display possibilities are highly beneficial in this context. This capability increases the user's manual correction performance in general, resulting in shorter editing times in both simple and complex areas of a point cloud. A big point cloud requires quick and easy access to smaller sub-areas of the point cloud such that a regular subdivision for a systematic aerial editing is highly advantageous (Fig. 6 and Fig. 12) and improves productivity.

To support our findings, we performed a manual edit of the same point cloud in [1] and [3] by classifying two roofs out of the point cloud. The manual editing steps consist of a polygon-shaped point selection and class assignment to the first roof, continued by a subsequent point selection requiring 3D navigation to allow for an appropriate point selection and class assignment to the second roof. This editing was carried out by three users in both software applications, demonstrating a performance gain of about 33% for manual editing.

REFERENCES

- [Alt10] Francesc Alted. Why modern cpus are starving and what can be done about it. *Computing in Science & Engineering*, 12(2):68–71, 2010.
- [Alt23] Francesc Alted. Blosc, an extremely fast, multi-threaded, meta-compressor library. <https://www.blosc.org/pages/>, 2023.
- [Ben04] Werner Benger. *Visualization of General Relativistic Tensor Fields via a Fiber Bundle Data Model*. PhD thesis, FU Berlin, 2004.
- [Ben09] Werner Benger. Classifying data for scientific visualization via fiber bundles. In Claude Leroy and Pier-Giorgio Rancoita, editors, *ICATPP-11, Como, Italy, Oct 5-9, 2009*. World Scientific, 2009.
- [BP89] David M. Butler and M. H. Pendley. A visualization model based on the mathematics of fiber bundles. *Computers in Physics*, 3(5):45–51, sep/oct 1989.
- [BRH07] Werner Benger, Georg Ritter, and René Heinzl. The Concepts of VISH. In *4th High-End Visualization Workshop, Obergurgl, Austria, June 18-21, 2007*, pages 26–39. Berlin, Lehmanns Media-LOB.de, 2007.

	1	2	3 (ours)
Polygon selection	yes	yes	yes
Free-style selection	yes	yes	yes
Selection modification (des-select / invert / add / subtract)	no	yes	yes
User-defined classes	yes	yes	yes
Class assignment to selection	instantly	confirmative	confirmative
Undo / Redo	yes	no	yes
Persistent editing history	no	no	yes
Subdivision for systematic editing	yes	no	yes
Restrict display to specific classes	yes	yes	yes
Viewer layout	multiple 3D views for overview & detail (Fig. 10)	zoomable 2D & dependent 3D display extent (Fig. 11)	single navigable 3D view of entire point cloud (Fig. 12)
Cross-sectional navigation	additional button clicks for continuous navigation within point cloud	navigation restricted to selected cross-section extent	simple mouse usage for continuous navigation within point cloud

Table 1: Feature comparison. The faster 3D navigation in HydroVish allowed all test users to perform the manual editing in about ≈ 15 seconds, that was about 3-6 seconds quicker than the ≈ 20 seconds required for the editing in the alternative application [1] (35-40% difference).

- [CZWea13] D. Chen, L. Zhang, Z. Wang, and et al. A mathematical morphology-based multi-level filter of lidar data for generating dtms. *Sci. China Inf. Sci.*, (56):1–14, 2013.
- [HDF23] HDF Group. Hierarchical data format version 5, 2000-2023.
- [HEA⁺21] S.S. Hansen, V.B. Ernstsens, M.S. Andersen, Z. Al-Hamdani, R. Baran, M. Niederwieser, F. Steinbacher, and A. Kroon. Classification of boulders in coastal environments using random forest machine learning on topo-bathymetric lidar data. *Remote Sens.*, 13(4101), 2021.
- [HM90] Robert Haber and D McNabb. Visualization idioms: A conceptual model for scientific visualization systems. *IEEE Computer Society Press: Los Alamitos, CA*, pages 74–93, 01 1990.
- [KP97] K. Kraus and N. Pfeifer. A new method for surface reconstruction from laser scanner data. 1997.
- [LG17] B. Lohani and S. Ghosh. Airborne lidar technology: a review of data collection and processing systems. pages 567–579, 2017.
- [LKS00] P. Lohmann, A. Koch, and M. Schäffer. Approaches to the filtering of laser scanner data. *Int Arch Photogrammetry Remote Sens XXXII I(B3/I)*, pages 534–541, 2000.
- [Man20] A review of airborne laser bathymetry for mapping of inland and coastal waters. *Journal of Applied Hydrography*, 116(6):6–15, 2020.
- [Man22] Underwater deadwood and vegetation from uav-borne topobathymetric lidar - the benefits of progress in uav and lidar sensor technology. *GIM International*, 2022.
- [MWSCC09] X. Meng, L. Wang, J. Silvàn-Càrdenas, and N. Currit. A multi-directional ground filtering algorithm for airborne lidar. *ISPRS J Photogrammetry Remote Sens 64(1)*, pages 117–124, 2009.
- [RBC⁺12] Marcel Ritter, Werner Benger, Biagio Cosenza, Keera Pullman, Hans Moritsch, and Wolfgang Leimer. Visual data mining using the point distribution tensor. Feb-Mar 2012.
- [SDB⁺21] Frank Steinbacher, Wolfgang Dobler, Werner Benger, Ramona Baran, Manfred Niederwieser, and Wolfgang Leimer. Integrated full-waveform analysis and classification approaches for topo-bathymetric data processing and visualization in hydrovish. *PFG Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 2021.
- [Sit01] G. Sithole. Filtering of laser altimetry data using a slope adaptive filter. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXIV(3/4)*, pages 203–210, 2001.
- [WSB⁺12] Gong Wei, Song Shalei, Zhu Bo, Shi Shuo, Li Faquan, and Cheng Xuewu. Multi-wavelength canopy lidar for remote sensing of vegetation: Design and system performance. *ISPRS Journal of Photogrammetry and Remote Sensing*, 69:1–9, 2012.

First Considerations in Computing and Using Hypersurface Curvature for Energy Efficiency

Jacob D. Hauenstein

The University of Alabama in Huntsville
301 Sparkman Drive
Huntsville, AL 35899
jacob.hauenstein@uah.edu

Timothy S. Newman

The University of Alabama in Huntsville
301 Sparkman Drive
Huntsville, AL 35899
timothy.newman@uah.edu

ABSTRACT

Energy consumption for computing and using hypersurface curvature in volume dataset analysis and visualization is studied here. Usage in both the base case and when more energy-optimal strategies, particularly computational (especially for linear algebraic steps) strategies, are the primary foci here and are considered for analysis tasks that are precursors to visualization. Compilation-based effects on energy usage are a secondary focus. Efforts here are on Intel x86, which is popular and has power measurement capabilities. Additionally, a first-time visualization of hypersurface curvature distributions in a brain imaging scenario is exhibited. The work aims to advance understanding of computing's energy footprint and to provide guidance for energy-responsible volume data analysis.

Keywords

Power Efficiency, Green Computing, Curvature, Volumetric Data, Hypersurface

1 INTRODUCTION

Energy consumption of data processing activities has recently received increased attention from policy-makers, data center deployment planners, and computing researchers. Methods to characterize computing energy use thus has been one direction for energy-related research in computing, with characterization still an early-stage technology [Lan23]. In this paper, one of our focuses is characterizing energy use for certain computations related to analysis and visualization. Some computing environments, like recent Intel x86 CPUs (our focus here), have built-in power-monitoring features that are accessible for inspection by software developers, making such characterizations accessible to interested parties. Other environments do not well-expose their power usage, making characterization more challenging for them. (Some reports involving use of external monitoring systems to measure power consumption do exist (e.g., [Lin19]), though, and generic estimators based on memory and core use also exist (e.g., [Lan21]).) Some of the characterizations have involved system-level considerations, especially for large data centers, with

those characterizations sometimes employed in setting purchase specifications. A characterization of server energy use has been reported by Fuchs et al. [Fuc20], for example. Additionally, characterizing energy from manufacturing of computing systems has also been considered [Gup22]. One other area of investigation has considered characterizations of competing computing paradigms, such as opportunities using FPGA computation (e.g., [EM20]). Comparative analyses of many instances of a class of algorithms have also been reported (e.g., [Hen20]).

Finally, another research theme has been creation and examination of strategies that a given type of algorithm can use to reach its solution in a more energy-optimal manner. Such approaches offer great promise—they are perhaps the “most productive...green computing activity” [Lan21]. They can be pursued either in an end-device agnostic manner or with focus on a specific computing device. Beckitt-Marshall [BM21], for example, has examined compiler optimization strategies to determine best optimization settings to achieve low energy usage for two file compression approaches, an ambient occlusion renderer, and several other algorithms on a widely-used type of CPU. Another strategy has been to exploit cache properties to reduce energy usage (e.g., [Tit15]).

The effort here both characterizes power usage and explores energy-efficient strategies for a class of volume data analysis algorithms, hypersurface curvature determination, and one volume visualization scheme, maximum intensity projection, run on Intel x86 (one of the most popular end computing environments for volume

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

data analysis). (Hypersurface curvatures are discussed in more detail in Section 2.) As one of the first such efforts for the volume data analysis arena, a limited series of strategies is considered here, although they are applicable to many other volume data analysis and visualization approaches.

2 BACKGROUND

This section presents some necessary background. This includes details about hypersurface curvature (including its utility and the mathematics for computing it) and details about the four hypersurface curvature determination methods considered in our energy usage experiments (presented later). Finally, some details about the aims and motivations of the work are presented.

2.1 Hypersurface Curvature

Curvature measures of 3D manifolds (often called *hypersurfaces* [Mon92]) have been found to be very useful descriptors for a variety of tasks, especially in the fields of geology (where such curvature measures are useful for identifying faults or fractures [Bra10] and for visualizing seismic phenomena [Ald14]) and medicine (where such curvature measures are useful for classification of tumors [Hir18] and arterial measurement [Suz18]). For such tasks, the three principal curvature values, denoted κ_1 , κ_2 , and κ_3 (ordered such that $\kappa_1 > \kappa_2 > \kappa_3$), are of value.

Hypersurface curvatures are also useful for surface reconstruction [Pap07] and for classification of data points within a volume, where such points can be classified based on the relative, relative absolute, and average values of these three principal curvature values [Hir01].

Formally, hypersurface curvature can be defined as follows. Let (u, v, w) denote a grid (or sample) point within a scalar volume (where $0 \leq u < N_u$, $0 \leq v < N_v$, $0 \leq w < N_w$) for a volume of size $N_u \times N_v \times N_w$. The value at point (u, v, w) is then denoted $f(u, v, w)$ with f representing the underlying function that generates the volume and f_u representing the partial derivative of f in the u direction. The hypersurface's three principal curvatures (i.e., of f) are then the eigenvalues of the matrix:

$$\frac{1}{l} \begin{bmatrix} f_{uu} & f_{uv} & f_{uw} \\ f_{uv} & f_{vv} & f_{vw} \\ f_{uw} & f_{vw} & f_{ww} \end{bmatrix} \begin{bmatrix} 1 + f_u^2 & f_u f_v & f_u f_w \\ f_u f_v & 1 + f_v^2 & f_v f_w \\ f_u f_w & f_v f_w & 1 + f_w^2 \end{bmatrix}^{-1}, \quad (1)$$

where

$$l = \sqrt{1 + f_u^2 + f_v^2 + f_w^2}. \quad (2)$$

Thus, computation of κ_1 , κ_2 , and κ_3 requires knowledge of the first derivatives, second derivatives, and

mixed partial derivatives of f . Often, the continuous form of f is unknown, because the data under consideration was acquired via a sensor. In such cases, these necessary derivatives must be estimated in order to calculate κ_1 , κ_2 , and κ_3 from Eqn. 1.

2.2 Hypersurface Curvature Determination Methods

Our studies consider four methods for determining hypersurface curvature from volumetric data. These methods all work by first estimating the necessary derivatives and then computing the hypersurface curvature using Eqn. 1. These four methods have previously been comparatively studied on the bases of their accuracy and run time [Hau21]. Because the methods all differ in their derivative estimation approach, all four methods exhibit varying run times, accuracies, and energy usages. Here, we briefly describe the four methods considered.

2.2.1 Taylor Exp.-based Conv. Kernels (TE)

One hypersurface curvature determination method, denoted **TE**, estimates derivatives using convolution. Specifically, it uses convolution filters derived from the Taylor Expansion along each axis to estimate all the necessary derivatives. Once these derivatives are known, the three principal curvatures are computed as the eigenvalues of Eq. 1. The **TE** approach has previously been used in determination of both surface curvature [Kin03] and hypersurface curvature [Hau19], where it was found to be among the fastest approaches.

The method's filters are determined according to a framework that allows construction of filters with arbitrary accuracy and continuity. For our experiments with the **TE** approach, we used filters with C^3 continuity and fourth order accuracy, following prior works (e.g., [Kin03]). At these continuity and accuracy parameters, the first derivative kernel is size 5 and the second derivative kernel is size 7.

2.2.2 B-Spline-based Derivatives (BS)

The **BS** hypersurface curvature determination method uses f as the coefficients of a tricubic B-Spline [Hau19]. This B-Spline represents a continuous form that approximates f , and the derivatives of that continuous form, in conjunction with Eq. 1, are used to determine the principal curvatures.

It is possible to configure the B-Spline in a number of ways (e.g., with varying numbers of knots or varying degree). For our experiments, we configured the B-Spline similarly to other reports—with knot count the same as input volume dimensions.

2.2.3 Orthogonal Polynomials-based Convolution Kernels (OP)

Another hypersurface curvature method, denoted **OP**, also uses convolution to estimate the necessary derivatives and then computes the resulting curvatures via Eq. 1. **OP** uses convolution kernels derived from orthogonal polynomials.

The **OP** method uses kernels of odd size N that are generated according to three functions $b_0(x)$, $b_1(x)$, and $b_2(x)$:

$$b_0(x) = \frac{1}{N}, \quad (3)$$

$$b_1(x) = \frac{3}{M(M+1)(2M+1)}x, \quad (4)$$

$$b_2(x) = \frac{1}{P(M)}\left(x^2 - \frac{M(M+1)}{3}\right), \quad (5)$$

with $M = \frac{N-1}{2}$. $P(M)$ is:

$$P(M) = \frac{8}{45}M^5 + \frac{4}{9}M^4 + \frac{2}{9}M^3 - \frac{1}{9}M^2 - \frac{1}{15}M. \quad (6)$$

From these kernels, derivatives are estimated via:

$$a_{ijk} = \sum_{u,v,w \in m \times m \times m} f(u,v,w) b_i(u) b_j(v) b_k(w), \quad (7)$$

where $m = \left\{ \frac{-(N-1)}{2}, \dots, \frac{(N-1)}{2} \right\}$. For example, the estimate f_u would be found using a_{100} .

Aside from one set of supplemental experiments at the end of Section 4, all of our experiments follow prior works (e.g., [Hau19]) and use $N = 7$.

Since convolution with such kernels implicitly does a least squares fitting [Fly89], OP results match those of a linear regression-based local surface fitting (without explicitly performing any linear regression).

2.2.4 Deriche Filter-based Convolution Kernels (DF)

The **DF** hypersurface curvature determination method also uses convolution to estimate derivatives and then computes curvature via Eq. 1. It uses convolution kernels based on three Deriche Filters (\hat{f}_0 for smoothing and \hat{f}_1 , \hat{f}_2 to estimate the first and second derivatives, respectively) [Der90]. \hat{f}_0 , \hat{f}_1 , and \hat{f}_2 are defined as:

$$\hat{f}_0(x) = c_0(1 + \alpha|x|)e^{-\alpha|x|}, \quad (8)$$

$$\hat{f}_1(x) = -c_1x\alpha^2e^{-\alpha|x|}, \quad (9)$$

$$\hat{f}_2(x) = c_2(1 - c_3\alpha|x|)e^{-\alpha|x|}, \quad (10)$$

where c_0 , c_1 , c_2 , and c_3 are scaling factors defined as:

$$c_0 = \frac{(1 - e^{-\alpha})^2}{1 + 2e^{-\alpha}\alpha - e^{-2\alpha}}, \quad (11)$$

$$c_1 = \frac{-(1 - e^{-\alpha^3})}{2\alpha^2e^{-\alpha}(1 + e^{-\alpha})}, \quad (12)$$

$$c_2 = \frac{-2(1 - e^{-\alpha^4})}{1 + 2e^{-\alpha} - 2e^{-3\alpha} - e^{-4\alpha}}, \text{ and } \quad (13)$$

$$c_3 = \frac{(1 - e^{-2\alpha})}{2\alpha e^{-\alpha}}, \quad (14)$$

with α a smoothing term. Monga et al. [Mon92] noted that smaller values for α are often required when estimating second derivatives compared to when estimating first derivatives.

Aside from one set of supplemental experiments at the end of Section 4, we have set $\alpha = 1.0$, since that same value for α was used in a prior work ([Hau21]) that explored the **DF** method.

2.3 Aims and Motivations

This work is motivated by prior work that considered accuracy and computational performance of hypersurface curvature computation [Hau19].

Energy-efficient algorithm strategies for linear system solutions has been one focus of prior work. Köhler and Saak [Kö19], for example, have explored such strategies, including: (1) combining certain evaluation activities (to save on data transfers); (2) use of a Newton-type method to compute matrix sign; (3) use of Gauss-Jordan elimination in lieu of LU-decomposition; and (4) changing one internal storage scheme to improve cache utilization.

Our strategies here, described later, also include efforts aimed at saving on data transfers and improving cache utilization, including in linear system components of hypersurface curvature determination.

3 METHODS/STRATEGIES

The strategies to achieve energy efficiency in the hypersurface curvature computation, primarily involving improvements in linear algebraic computations and compiler products, are described in this section.

3.1 Linear Algebra Memory & Computational Improvements

Hypersurface curvature determination in volumetric datasets using the methods previously reported in the literature requires estimation of derivative quantities in the dataset followed by steps including finding inverses, determinants, and performing matrix multiplication. These operations implement the equations described above. Standard solver libraries can be used

to compute those, and we initially used the *Armadillo* library [San16] for that. (Later, we report comparisons versus use of *Armadillo*.) *Armadillo* has several characteristics that are not energy-optimal, though, for application here. For one, it uses its own internal format for matrices, which incurs some overhead in both time and energy consumption. Also, our core matrices are symmetric (but not necessarily positive definite). (N.B.: the matrix product is not symmetric, though.) Inspection of the *Armadillo* library functions revealed that the inverse and determinant computation in them did not exploit matrix symmetry for (non-positive definite) symmetric matrices, which means that there are redundant computations in the case of such matrices. Our approach avoids these redundancies, thereby saving both energy and time.

Our approaches to improve energy efficiency of linear algebraic-based computations for hypersurface curvature determination involved replacement of *Armadillo* with our own realizations that store matrices as standard arrays and avoid unnecessary computations performed in the *Armadillo* code. Some aspects of the savings our realizations achieve for matrix determinant follow. 3×3 matrices of the form:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad (15)$$

have determinants of the form:

$$a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}. \quad (16)$$

Computing such a determinant involves loading the 9 matrix entries into the control unit of the CPU and performing 5 additions and 9 multiplies. Ultimately, *Armadillo* realizes these same actions. Our linear algebra strategy for determinant calculation, however, exploits that in our symmetric matrices, $d = b$, $g = c$, and $h = f$. Thus, for our hypersurface curvature computation the determinant of the matrix shown in Eq. 15 has the form:

$$a \begin{vmatrix} e & f \\ f & i \end{vmatrix} - b \begin{vmatrix} b & f \\ c & i \end{vmatrix} + c \begin{vmatrix} b & e \\ c & f \end{vmatrix}. \quad (17)$$

The Eq. 17 can be reduced to a form such as:

$$(ei - ff)(a) - (bi - \mathcal{K})(b) - ecc, \quad (18)$$

where $\mathcal{K} = 2cf$, with \mathcal{K} computed in our algorithm as a two step process:

$$\mathcal{K} = cf; \mathcal{K} = \mathcal{K} + \mathcal{K}.$$

The net effect of our approach to finding the determinant is a computation with 5 additions and only 8 multiplies that also only loads 6 matrix entries into the control unit of the CPU. Our approach thus reduces loading, reduces register pressure, and can remove one instance of the most expensive operation in the process.

This arithmetic approach extends comparably to the computation of the inverse.

Our linear algebra strategy thus has four components in all. They include: (1) avoiding special, generic matrix storage formats (and unnecessary storage of those) via use of standard matrix storage; (2) exploiting matrix symmetry and size in determinant computation; (3) exploiting matrix symmetry and size in finding matrix inverses; and (4) exploiting knowledge of matrix multiplicand form in finding matrix products.

3.2 Compilation Improvements

We also considered and report here on two categories of compilation-based approaches for energy reduction for the language environment we used, C/C++. C/C++-based solutions are known to be among the most energy-efficient [Per17]. The two approaches are (1) the use of a very high optimization compiler setting (our compiler was gcc, and its very high optimization setting is the O3 one), denoted VH henceforth, and (2) the use of the special mathematical operation optimization setting (in gcc, this setting is called *ffastmath*), denoted FM henceforth. (N.B., because FM relaxes some floating point compliance within the compiler, it can impact accuracy. In our experiments on a real dataset, the average change in the curvature values due to the use of FM was 2.8×10^{-16} and the maximum change in any of the curvature values due to the use of FM was 1.5×10^{-14} .)

(Automated loop unrolling, denoted Unroll henceforth, was also considered and is reported later. It cannot be considered a general improvement in application to methods here, as discussed later.)

3.3 B-Spline Library-Related Improvements

For the **BS** method, we implemented two variants of the method. Our first variant, simply denoted **BS**, utilized the header-only *vspline* library [kfj23]. The second variant, denoted **BSWM5**, used the B-Spline functionality present in the larger *WildMagic 5* library [Ebe04]. Our results, presented later, consider the energy usage of both of these variants of the B-Spline-based approach.

4 RESULTS

Next, results are reported. All results were determined on a CPU similar to one used in some related work. Here, the environment used one core of an Intel Core i5-8279U CPU. The operating system used was Ubuntu Server 22.04.1 GNU/Linux. All runs were built using the gcc compiler (version 11.3.0). (N.B., we found that code compiled with gcc was consistently among the best (in terms of energy usage) compared to both

clang and icc.) Energy measurements and timings were done via the Performance API (PAPI) software [Bro00], which in turn measured energy via the processor's Running Average Power Limit (RAPL) feature. RAPL, which is supported on Sandy Bridge and newer Intel CPUs [Kha18], allows measuring CPU power usage. RAPL uses hardware counters in conjunction with factors such as temperature and leakage to estimate CPU energy usage, and it supports measurement of energy usage using different domains including *package* (total CPU package energy usage), *PP0* (CPU core energy usage), and *DRAM* (DRAM controller). A previous study found RAPL measurements to be accurate while also exhibiting negligible overhead [Kha18]. The governor on the computer was set to the "performance" mode in order to maintain more consistent clock rates, and all code was run as the "root" user (in order to ensure access to the RAPL hardware) using the *taskset* utility (in order to ensure that all runs were done on the same processor core).

To find the cost of determining hypersurface curvature via canned library (i.e., Armadillo) based linear algebraic routines, we performed timing and energy use. For all experiments, five runs were taken and the trimmed mean (i.e., discarding most and least energy usage run) was computed. For one experiment, a $256 \times 256 \times 256$ volumetric dataset mathematically generated from a polynomial function used in other reported work on curvature in volumes and called the "Genus3" polynomial there ([Hau20], Eqn. 39) was used. On such a dataset, the baseline (optimized) result (i.e., compilation using gcc with -O2 on the system reported above) for hypersurface curvature determination using the OP method was found to use 776.9 J (for CPU package energy) and take 46.6 seconds of CPU time. Using the VH optimization improved energy use by a factor of 1.066 and run-time by a factor of 1.075. Coupling VH and FM resulted in total improvement factors of 1.077 and 1.082, respectively. However, replacement of Armadillo with our linear algebraic strategies coupled with the VH compilation strategy resulted in both energy and time improvement factors of 1.25. (N.B., the measured energy usage and time typically varied by about 1% from run to run.)

A breakdown of individual linear algebraic-related strategy time and (package) energy improvement factors are shown in Table 1 (versus original baseline ("Base") energy use). To ensure that these experiments consider the overall impact of the underlying libraries in a wide cross-section of scenarios, we computed these results on a randomly-generated 256^3 volume, without regard to the end curvature estimator used. (Of note here: time savings does not always equal energy savings.) Total energy and time improvements from these strategies taken together are also shown.

Strategy	Base Energy	Energy Imp.	Time Imp.
Storage	25.40J	11.98x	7.08x
Det.	6.26J	2.95x	2.54x
Inv.	11.27J	1.43x	1.37x
Mult.	11.12J	1.40x	1.49x
Total:	54.05J	2.70x	2.18x

Table 1: L.A. Strategy Energy Use (J) & Improvement

The numbers in Table 1 show the energy usage measurements of these operations in isolation (i.e., not when used in combination as part of a larger problem solving task). When these operations are performed in combination as part of larger task, such as curvature determination, the improvements can be even more substantial. In fact, the results in Table 1 account for less than one-half the total net improvement in time and energy. Such results for these optimizations incorporated into curvature determination are described next.

Table 2 shows overall energy usage results for the complete curvature determination of the four methods when applied to a $256 \times 256 \times 72$ brain magnetic resonance angiography (MRA) dataset. The BS Method library-related results are compared in the last two columns of the table. The entries marked "Custom" here refer to the use of our linear algebraic strategies rather than use of the Armadillo library. The two best choices here are (1) VH with FM and the linear algebraic strategies and (2) VH with FM and automated loop unrolling with the linear algebraic strategies. Average energy consumption improvement is 16.72% for the first choice, which is equivalent to 1.20 times improvement. For the second choice, average energy consumption improvement is 15.91%, which is equivalent to 1.19 times improvement. Since the automated loop unrolling results are, overall, yielding performance improvement about the same as optimization without them, we recommend the first choice here, however practitioners using the DF curvature determination method would still benefit slightly from its use, it appears.

Table 3 shows run times for the same dataset. The locations of the red cells differs between Table 2 and Table 3, which is indicative of something that has also been observed in prior studies: there is not always a direct correspondence between run time and energy usage. Thus, as a general conclusion, results here (again) indicate that optimization for energy consumption and optimization for run-time may require different methods and approaches.

Fig. 1 shows a composite rendering of three maximum intensity projections (MIP), one per principal curvature, of the MRA dataset. (κ_1 's MIP forms the red channel of the composite here. κ_2 's forms the green channel. κ_3 's, which was clamped to remove low values, forms the blue channel.) MIP is popular for raw MRA data

Method	OP	TE	DF	BS	BSWM5
Baseline (Armadillo)	176.24 J	135.58 J	185.28 J	230.81 J	399.73 J
Baseline (Custom)	-11.91%	-17.75%	-8.49%	-10.24%	-6.52%
Baseline FM (Armadillo)	-0.84%	-0.92%	3.91%	-0.51%	-0.35%
Baseline FM (Custom)	-12.87%	-17.56%	-12.19%	-11.08%	-10.65%
Baseline Unroll (Armadillo)	2.42%	0.03%	-1.27%	4.86%	-5.49%
Baseline Unroll (Custom)	-15.02%	-17.93%	-10.86%	-5.35%	-12.34%
Baseline Unroll FM (Armadillo)	2.75%	-0.61%	-2.46%	3.11%	0.67%
Baseline Unroll FM (Custom)	-15.35%	-18.96%	-9.65%	-4.45%	-10.41%
VH (Armadillo)	-4.17%	-3.49%	-4.45%	-0.99%	-6.42%
VH (Custom)	-16.86%	-21.10%	-17.21%	-10.41%	-10.57%
VH FM (Armadillo)	-4.86%	-4.54%	-6.97%	-3.38%	-3.64%
VH FM (Custom)	-17.57%	-21.55%	-17.53%	-12.78%	-14.17%
VH Unroll (Armadillo)	-1.21%	-3.39%	-4.09%	0.64%	0.63%
VH Unroll (Custom)	-16.14%	-20.37%	-16.16%	-10.37%	-11.94%
VH Unroll FM (Armadillo)	-2.86%	-3.61%	-4.68%	-0.16%	-5.91%
VH Unroll FM (Custom)	-16.75%	-20.62%	-17.80%	-9.91%	-14.47%

Table 2: Energy usage on the MRA dataset (trimmed means of five runs), relative to Baseline for each curvature determination method in conjunction with the optimization strategies. Cell backgrounds are color mapped based on energy usage relative to baseline. In each column, the **bold** entry indicates the approach with lowest energy usage for that method and the yellow bordered entry indicates the approach with the fastest run time for that method. Overall, best energy improvement averages about 1.20 times.

Method	OP	TE	DF	BS	BSWM5
Baseline (Armadillo)	10.65 s	7.85 s	11.43 s	13.00 s	25.91 s
Baseline (Custom)	-11.12%	-18.02%	-8.54%	-10.45%	-5.24%
Baseline FM (Armadillo)	-0.42%	-0.65%	3.53%	-0.55%	-0.47%
Baseline FM (Custom)	-12.02%	-17.85%	-11.80%	-11.49%	-3.86%
Baseline Unroll (Armadillo)	2.45%	-0.24%	-1.63%	4.66%	-0.23%
Baseline Unroll (Custom)	-14.15%	-18.33%	-11.03%	-5.69%	-6.21%
Baseline Unroll FM (Armadillo)	3.15%	-1.08%	-2.31%	3.10%	0.81%
Baseline Unroll FM (Custom)	-14.40%	-19.12%	-9.90%	-4.88%	-3.70%
VH (Armadillo)	-3.92%	-3.94%	-5.05%	0.91%	-0.91%
VH (Custom)	-15.93%	-21.45%	-17.29%	-8.08%	-3.69%
VH FM (Armadillo)	-4.69%	-4.76%	-7.91%	-1.43%	1.93%
VH FM (Custom)	-16.62%	-21.96%	-18.05%	-10.22%	-7.35%
VH Unroll (Armadillo)	-1.13%	-4.01%	-4.97%	3.85%	0.70%
VH Unroll (Custom)	-15.46%	-21.10%	-16.86%	-6.78%	-6.15%
VH Unroll FM (Armadillo)	-3.29%	-4.14%	-5.75%	2.89%	-0.86%
VH Unroll FM (Custom)	-16.21%	-21.35%	-18.22%	-7.12%	-7.61%

Table 3: Run times on the MRA dataset (trimmed means of five runs), relative to Baseline for each curvature determination method in conjunction with the optimization strategies. Cell backgrounds are color mapped based on energy usage relative to baseline. In each column, the **bold** entry indicates the approach with lowest run time for that method.

rendering. To our knowledge, this figure here is the first MIP rendering of hypersurface curvature features in MRA data, however. **OP**'s results were used to produce Fig. 1. The amount of energy used to produce the rendering is 146.94 J on average (146.5 J for curvature computation and 0.42 J for MIPs).

Table 4 presents the results of energy usage experiments for two other sets of parameters for both **OP** and **DF** versus the values used in the prior experiments. In this table, **OP** N denotes application of **OP** with an estimation filter of size N and **DF** α denotes application of **DF** with a smoothing parameter of value α . As N increases, the energy usage of **OP** increases. This is expected, as

larger N values require more computation. Despite **OP** 5 and **OP** 9 being 28% smaller and larger, respectively, than **OP** 7, the energy differences are substantially less than that. To some extent, this is to be expected, as only some of the computation is related to the size of the estimation filter. **OP** 5 uses about 0.95 times the energy of **OP** 7 and **OP** 9 uses about 1.07 times the energy of **OP** 7, with energy usage scaling up more slowly when all strategies are used. The energy differences are much less noteworthy on **DF**. To some extent, this is to be expected, as α does not substantially change the amount of computation. We do note that running **DF** with $\alpha = 0.5$ uses slightly less energy than running it

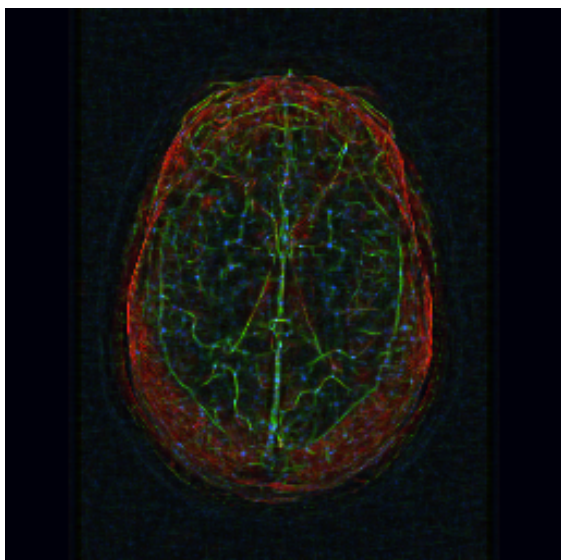


Figure 1: MIP: κ_1 , κ_2 , κ_3 Composite, MRA Dataset.

with $\alpha = 1.0$ and $\alpha = 2.0$. Our hypothesized reason for this is that the optimal filters used by **DF** rely on the *pow* function, which can be more optimal for certain exponents, in particular ones like 0.5 and 2, which appear to be exhibited more by the $\alpha = 0.5$ than by the $\alpha = 1.0$ and $\alpha = 2.0$ settings.

5 CONCLUSIONS

Most of our strategies exploit known properties of the hypersurface curvature determination domain to reduce energy consumption (and computation time!). Our storage-based strategy yielded the largest energy (and time) improvement for the linear algebraic computational components of the hypersurface curvature computation, achieving nearly a 12-fold improvement in energy usage, when considered in isolation. Overall, the linear algebraic computational strategies achieved nearly a 3-fold improvement in energy usage for those components of the hypersurface curvature determination, when taken together.

For practitioners using a B-spline based method for hypersurface curvature determination, use of the *vspline* library appears to enable significantly lower energy use than use of the *WildMagic 5* library; that results in about a 1.73 times improvement in energy consumption.

For practitioners, the best generic advice for energy-efficient hypersurface curvature determination is to use the VH and FM compilation strategies coupled with the linear algebraic strategies, which can result in about a 1.20 times energy improvement.

Our strategies can provide means for reduced energy consumption for hypersurface-based volume data analyses and visualizations. Data analyses and visualizations using them can thus extend battery life as well as reduce their environmental impact.

ACKNOWLEDGMENTS

The research reported here was conducted independently; it was not sponsored by any sort of energy-related organization.

6 REFERENCES

- [Ald14] G. Aldrich, A. Gimenez, M. Oskin, R. Strelitz, J. Woodring, L. H. Kellogg, and B. Hamann. Curvature-based crease surfaces for wave visualization. *Vision, Modeling & Vis.*, pp. 39–46, 2014.
- [BM21] J. Beckitt-Marshall. *Improving Energy Efficiency through Compiler Optimizations*. Bowdoin College (Honors Project 239 Report), 2021.
- [Bra10] L. Bravo and M. Aldana. Volume curvature attributes to identify subtle faults and fractures in carbonate reservoirs: Cimarona formation, middle magdalena valley basin, colombia. *2010 SEG Annual Meeting*. OnePetro, 2010.
- [Bro00] S. Browne, J. Dongarra, N. Garner, G. Ho, and P. Mucci. A portable programming interface for performance evaluation on modern processors. *Int'l J. High Perf. Comp. Appl.*, 14(3):189–204, 2000.
- [Der90] R. Deriche. Fast algorithms for low-level vision. *IEEE T-PAMI*, 12(1):78–87, Jan 1990.
- [Ebe04] D. Eberly. *3D game engine architecture: engineering real-time applications with Wild Magic*. CRC Press, 2004.
- [EM20] D. El Mezeni and L. Saranovac. Fast guided filter for power-efficient real-time 1080p streaming video processing. *J. Real-Time Image Processing*, pp. 511–525, 2020.
- [Fly89] P. Flynn and A. Jain. On reliable curvature estimation. *Proc., IEEE Comput. Vision and Pat. Recog.* '89, pp. 110–116, 1989.
- [Fuc20] H. Fuchs, A. Shehabi, M. Ganeshalingam, L.-B. Desroches, B. Lim, K. Roth, and A. Taso. Comparing Datasets of Volume Servers to Illuminate their Energy use in Data Centers. *Energy Efficiency*, 13:379–392, 2020.
- [Gup22] U. Gupta, Y. Kim, S. Lee, J. Tse, S. Hsien-Hsin, G.-Y. Wie, D. Brooks, and C.-J. Wu. Chasing carbon: The elusive environmental footprint of computing. *IEEE Micro*, 42(4):37–47, 2022.
- [Hau19] J. D. Hauenstein and T. S. Newman. Exhibition and evaluation of two schemes for determining hypersurface curvature in volumetric data. *J. WSCG*, 27(2):121–129, 2019.

Method	OP 9	OP 7	OP 5	DF 2.0	DF 1.0	DF 0.5
Baseline (Armadillo)	188.44 J	175.86 J	166.03 J	189.02 J	186.74 J	185.25 J
VH Unroll FM (Custom)	154.15 J	146.66 J	140.73 J	155.11 J	152.78 J	151.05 J

Table 4: Energy usage on the MRA dataset (trimmed means of five runs) for Baseline vs. all strategies at three different parameter settings for **OP** and **DF**.

- [Hau20] J. D. Hauenstein and T. S. Newman. Descriptions and evaluations of methods for determining surface curvature in volumetric data. *Computers & Graphics*, 86:52–70, 2020.
- [Hau21] J. D. Hauenstein and T. S. Newman. New methods and novel framework for hypersurface curvature determination and analysis. *J. WSCG*, 29(1–2):11–20, 2021.
- [Hen20] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning. *J. Machine Learn. Res.*, 21:1–43, 2020.
- [Hir01] Y. Hirano, A. Shimizu, J.-i. Hasegawa, and J.-i. Toriwaki. A tracking algorithm for extracting ridge lines in three-dimensional gray images using curvature of four-dimensional hypersurface. *Systems and Comput. in Japan*, 32(12):25–37, 2001.
- [Hir18] Y. Hirano. Categorization of lung tumors into benign/malignant, solid/GGO, and typical benign/others. K. Suzuki and Y. Chen, editors, *Artificial Intelligence in Decision Support Systems for Diagnosis in Medical Imaging*, pp. 193–208. Springer Nature, 2018.
- [kfj23] kfj. vspline: C++ template library for B-splines, 2023. <https://bitbucket.org/kfj/vspline/>.
- [Kha18] K. N. Khan, M. Hirki, T. Niemi, J. K. Nurminen, and Z. Ou. Rapl in action: Experiences in using rapl for power measurements. *ACM Trans. Model. Perform. Eval. Comput. Syst.*, 3(2), mar 2018.
- [Kin03] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller. Curvature-based transfer functions for direct volume rendering: methods and applications. *Proc., IEEE Visualization '03*, pp. 513–520, 2003.
- [Kö19] M. Köhler and J. Saak. Frequency Scaling and Energy Efficiency regarding the Gauss-Jordan Elimination scheme with Application to the Matrix-sign-Function on OpenPOWER 8. *Concurrency and Computation: Practice and Experience*, 31(6):e4504, 2019.
- [Lan21] L. Lannelongue, J. Grealey, and M. Inouye. Green Algorithms: Quantifying the Carbon Footprint of Computation. *Advanced Science*, 8:202100707, 2021.
- [Lan23] L. Lannelongue and M. Inouye. Carbon Footprint Estimation for Computational Research. *Nature Reviews Methods Primers*, 3:Article 9, 2023.
- [Lin19] J. Lin, C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. *Proc., IEEE/CVF Int'l Conf. on Computer Vision '19*, pp. 7083–7093, 2019.
- [Mon92] O. Monga and S. Benayoun. Using partial derivatives of 3D images to extract typical surface features. Research Report RR-1599, INRIA, 1992.
- [Pap07] L. Papaleo. An approach to surface reconstruction using uncertain data. *Int'l J. of Image and Graphics*, 07(01):177–194, 2007.
- [Per17] R. Pereira, M. Couto, F. Ribeiro, R. Rua, J. Cunha, J. Fernandes, and J. Saraiva. Energy efficiency across programming languages. *Proc., ACM SIGPLAN Int'l Conf. Soft. Lang. Engg.*, pp. 256–267, 2017.
- [San16] C. Sanderson and R. Curtin. Armadillo: a template-based c++ library for linear algebra. *Journal of Open Source Software*, 1(2):26, 2016.
- [Suz18] H. Suzuki, Y. Kawata, N. Niki, T. Sugiura, N. Tanabe, M. Kusumoto, K. Eguchi, and M. Kaneko. Automated assessment of aortic and main pulmonary arterial diameters using model-based blood vessel segmentation for predicting chronic thromboembolic pulmonary hypertension in low-dose ct lung screening. *Medical Imaging 2018: Comput.-Aided Diagnosis*, volume 10575, 2018.
- [Tit15] J. Tithi, P. Ganapathi, A. Talati, S. Aggarwal, and R. Chowdhury. High-performance energy-efficient recursive dynamic programming with matrix-multiplication-like flexible kernels. *Proc., 29th Int'l Par. and Dist. Processing Symp.*, pp. 303–312, 2015.

MS-PS: A Multi-Scale Network for Photometric Stereo With a New Comprehensive Training Dataset

Clément Hardy
Normandie Univ,
UNICAEN, GREYC
Caen, France
clement.hardy@unicaen.fr

Yvain Quéau
Normandie Univ, CNRS,
GREYC
Caen, France
yvain.queau@ensicaen.fr

David Tschumperlé
Normandie Univ, CNRS,
GREYC
Caen, France
david.tschumperle@unicaen.fr

ABSTRACT

The photometric stereo (PS) problem consists in reconstructing the 3D-surface of an object, thanks to a set of photographs taken under different lighting directions. In this paper, we propose a multi-scale architecture for PS which, combined with a new dataset, yields state-of-the-art results. Our proposed architecture is flexible: it permits to consider a variable number of images as well as variable image size without loss of performance. In addition, we define a set of constraints to allow the generation of a relevant synthetic dataset to train convolutional neural networks for the PS problem. Our proposed dataset is much larger than pre-existing ones, and contains many objects with challenging materials having anisotropic reflectance (e.g. metals, glass). We show on publicly available benchmarks that the combination of both these contributions drastically improves the accuracy of the estimated normal field, in comparison with previous state-of-the-art methods.

Keywords

Photometric stereo, 3D-reconstruction, normal map estimation, multi-scale architecture, new dataset

1 INTRODUCTION

Photometric stereo (PS) is a 3D-reconstruction technique that estimates the 3D normal at each point of the surface of an object, using three or more photographs taken from the same viewpoint but with different lighting directions. Early works in this field (e.g. [38]) considered the ideal case of a perfect Lambertian surface. However, most images of real world objects exhibit a wide variety of complex lighting effects, which are not well predicted by Lambert's law. Especially, objects' reflectance often includes a specular component, giving a more or less *shiny* appearance to the image surface. Translucent surfaces, such as glass and acrylic, do not respect Lambert's law either. These kind of materials remain in most cases, poorly managed by traditional photometric stereo solutions [31]. In order to manage non-Lambertian surfaces, deep learning methods based on convolutional neural networks have recently emerged as the most efficient ones [31, 34]. The quality of results obtained by such approaches relies on two main factors:

1. The architecture of the network, which must ensure a good capacity for generalization on new data, including data with a different size from the training set.
2. The quality of the learning dataset, which must be as representative as possible of the diversity of observable light phenomena, for the network to be able to differentiate materials from each other.

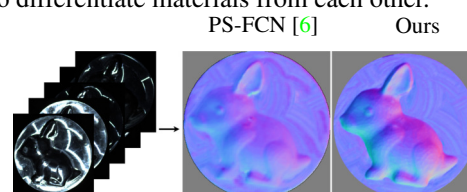


Figure 1: From a set of images taken under different illumination directions (left), photometric stereo estimates a normal map (right). Our proposed method is particularly efficient when used on challenging anisotropic materials, e.g. metal and glass as with this aluminium bunny from [31].

Contributions

Here, we propose a deep learning-based method for the problem of calibrated PS (known lighting direction and intensities), with the following features:

- A multi-scale network architecture for PS, which analyzes the input images simultaneously at different scales;
- A new synthetic training set featuring a wide variety of geometry and non-Lambertian reflectance.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Using these two contributions together, we show that challenging materials with anisotropic reflectance (e.g. metal, glass) can be handled appropriately in the PS problem (Fig. 1). The underlying core idea is that information over the *whole* image is indeed necessary to infer the 3D normal. Otherwise, complex lighting effects like inter-reflections in metallic objects or sub-surface scattering inside glass cannot be analyzed. On the contrary, our proposed multi-scale architecture takes advantage of all available complex geometric/lighting information and long-distance pixel correlations when inferring the 3D normal map.

2 RELATED WORK

Deep learning techniques for photometric stereo are all based on the use of Convolutional Neural Networks (CNN). Typically, a fully CNN architecture requires a fixed number of input images. However, in photometric stereo, the number of images depends on the acquisition procedure. To avoid having to train a different network model for each possible number of input images, two alternatives have been considered in the literature.

Observation map VS pooling

The first alternative consists in using an observation map [11, 13, 22, 25, 41], which projects all observations of each pixel under different illuminations into a fixed-size space - typically a sampled hemisphere. Therefore, an observation map makes a fixed-size summary of the information contained in a variable-size set of images. However, the spatial information (intra image) is lost, and the performance drops when the number of input images is small (typically, <10) [12].

The second alternative rather resorts to specific pooling modules [5, 6, 16, 18, 37], which aggregate the different features of each image extracted by previous convolution layers. This allows to obtain fixed-size image features from a variable number of input images. Different pooling methods can be considered. It is shown in [6] that max pooling performs better than average pooling as soon as the number of images exceeds 16. The latter tends to over-smooth the salient features and to be too sensitive to the regions of images with little interest, although a max pooling can also sometimes ignore a large proportion of the features extracted [17]. Still, in contrast to the observation map approach, pooling methods pay attention to intra image information, despite using less the variations of pixel values across the images.

Architectural variants

To overcome the drawbacks of both these approaches, Yao et al. [40] introduced a graph method called GPS-NET. It first aggregates the inter-image information by using a graph structure, and then uses a CNN to predict a 3D normal map. This graph structure therefore allows to preserve the spatial information. More recently,

Ikehata [12] proposed a dual-branch transformer (PS-transformer). One branch takes as input the pixels under different illuminations to get the inter-information, the other branch processes the images to get the spatial one. The features extracted are then aggregated, and a CNN finally gives the 3D normal map. However, as mentioned in [12] transformers are not particularly suitable for dense problems (in our case, a large number of input images).

In this paper, we rather consider the pooling-based scheme from [6] as a baseline model, and broaden it to a multi-scale architecture. Multi-scale architecture for photometric stereo has already been used, e.g., by Lichy et al. in the context of directional lighting with few images (no more than 6 images in inputs) [24], or for near (non-parallel) lighting [23]. On the contrary, we design our method to handle the directional lighting case with a large number of input images (e.g. 96 images).

Existing training datasets

Regardless of its architecture, a neural network needs to be trained on a proper dataset to perform well. In practice though, it is very difficult to acquire a large dataset of real images with 3D ground truths of photographed objects. For this reason, deep photometric stereo networks proposed in the literature often rely on training datasets of synthetic 3D objects, notably the *Blobby* and *Structure* datasets introduced in [6], and *CyclePS* in [11].

The *Blobby* dataset is composed of 10 geometric shapes, each one observed from 1296 distinct view-points. As the name suggests, the shapes in *Blobby* are rather smooth and regular (Fig. 2a). The *Structure* dataset consists in objects with complex geometry containing fine details (Fig. 2b). It is composed of 8 objects, rendered in 3D from 1387 to 6874 view-points. To simulate surfaces with non-Lambertian light reflectance, a material from the *MERL* [28] dataset is randomly drawn and applied in each rendering, providing a total of 25920 samples for *Blobby* and 59292 for *Structure*. In both cases, each sample is rendered under 64 different light directions, randomly selected on the hemisphere (Fig. 3c).

Finally, the *CyclePS* [11] dataset is also composed of complex objects, but contains only 18 objects rendered from 10 views (Fig. 2c). However, the number of materials available is substantial because *Disney's principled BSDF* [3] parametric reflectance model is used. It allows the variation of the base colour, roughness, proportion of specular reflectance, etc., thus the objects can be rendered using a near infinite number of materials. The training dataset presented in the present paper will also feature the possibility to generate as many materials as needed, while also considering much more geometric shapes than in existing sets.

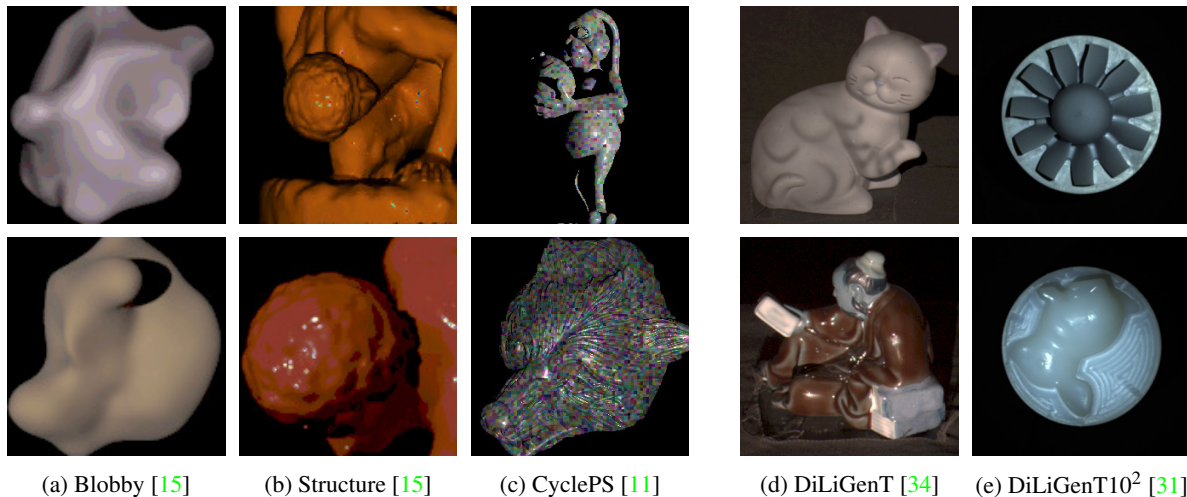


Figure 2: Samples from existing datasets. The first three [15, 11] are synthetic ones, used for training the neural networks. Both the last ones [34, 31] are real-world datasets used for benchmarking. Our proposed multi-scale architecture is evaluated on both benchmarking datasets, and trained on a new synthetic training set, which contains much more objects with *non-Lambertian* reflectance.

Existing benchmarking datasets

To validate the relevance of the training datasets, as well as to verify that the models trained on these synthetic data are able to generalize to real images, two real-world datasets exist: *DiLiGenT* [34] and *DiLiGenT10²* [31].

The *DiLiGenT* dataset comprises 10 different objects, taken from the same viewpoint under 96 different illuminations (Fig. 3a). The reflectance of the objects in this dataset goes from quasi-Lambertian to moderately specular. For each photographed object, the ground truth normal map is provided, as well as the calibrated lighting directions and intensities. Therein, the ground truth geometry was acquired by manually registering laser scans with the images.

The *DiLiGenT10²* dataset contains 10 different objects. Each object was explicitly fabricated with 10 different materials and photographed under 100 calibrated illuminations (Fig. 3b). The ground truth was not obtained by scanning the objects, but from the 3D digital models used to machine the objects. This real dataset is particularly interesting for evaluating performances on highly specular materials and translucent ones. Indeed, it contains metallic materials, such as aluminium or steel, and a translucent one (acrylic). This dataset also contains diffuse and slightly specular materials, hence most of real-world material characteristics are present. The diversity of object shapes is also high as it contains objects with simple geometry like balls but also complex ones like turbines. It offers the opportunity to test the impact of diverse inter-reflection, shadow and shading effects. Today, it is the most complete dataset composed of *real* images available in PS.

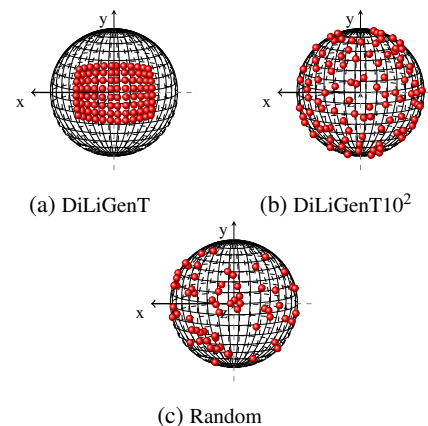


Figure 3: Distribution of illumination directions in the real *DiLiGenT* and *DiLiGenT10²* datasets, and an example of a random distribution. The *z*-axis corresponds to the optical axis of the camera, with the imaged object at coordinates (0,0,0).

Uncalibrated PS

In all the methods discussed above, the light directions and intensities are assumed to be known, i.e. we consider the *calibrated* PS problem. When these acquisition parameters are unknown, the problem is called *uncalibrated*. Uncalibrated PS has been studied e.g. in [5, 14, 20], and partially solved by defining a first neural network that predicts the lighting parameters associated with each acquired image. This estimated data is then fed into a second network that solves the problem of calibrated PS. Managing non-directional lighting, e.g. near point-light sources [26, 32] or natural illumination [9, 14, 29], is another ongoing research problem. In this paper we focus on the case of *calibrated* PS with known *directional* light sources.

3 A NEW MULTI-SCALE ARCHITECTURE FOR PS

The multi-scale architecture we propose builds upon the normal estimation network introduced in [6]. Therein, each image is first normalized by the calibrated lighting intensity, and then concatenated with the calibrated direction. The resulting “image” forms the input to the feature extractor which processes each (image, direction) pair independently. Then, all the independent features are aggregated through a feature aggregation module, and lastly a regression module predicts the normal map.

In order for the normal estimation to perform equivalently well on low-frequency geometry and high-frequency details, we propose to embed this network in a *multi-scale* approach which progressively refines the result as the spatial scale increases. Thus, our model first focuses on the *global* aspect of the object, then progressively insert *details* such as cracks, slight bumps, or holes as illustrated in Fig. 4.

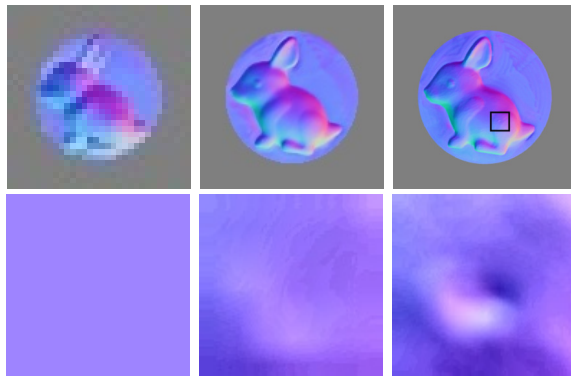


Figure 4: Multi-scale normal estimation at three different scales (bottom row is a contrast-enhanced zoom on the rectangle area). Low-detail geometry is reconstructed from the first levels. High-frequency details get refined as the scale increases.

The proposed multi-scale network combines two independent architectures (Fig. 5). The first stage takes as inputs the calibrated lighting directions and the images (downsampled from the original images to some initial resolution r_0), and outputs a low-resolution normal map with the same resolution r_0 . This first stage is essentially similar to the normal estimation network proposed in [6]. In the second stage, the low-resolution 3D normal map is up-sampled to a resolution $r_1 = 2r_0$ (using bilinear interpolation followed by normalization to enforce the unit-length constraint on normal vectors), and concatenated with the images (down-sampled from the original input images to resolution r_1) and lighting directions. The process is then repeated until the resolution of the original images is reached. In these sequential stages, the inputs differ from the first stage, thus a new, independent architecture is obviously nec-

essary. Yet, let us emphasize that since this new architecture is completely convolutional (except the pooling layer) and as only the spatial resolution changes from stage to stage, we can share the weights between each processed scale. Therefore, only two networks actually need being trained, independently from the number of scales. The network formed by these two sub-networks is trained by minimizing the cosine similarity, which measures the angular difference between the estimated 3D normals and the ground truth ones. It is defined as follows:

$$l_{normal} = 1 - \sum_{ij} N_{ij}^T \hat{N}_{ij}, \quad (1)$$

where \hat{N}_{ij} is the estimated normal at pixel (i, j) , and N_{ij} is the ground truth one. In terms of computational cost, our multi-scale CNN has 4.4 millions parameters. In comparison with the mono-scale approach, it uses only 5% more memory and takes 14% more time for inference.

As remarked in [24], one of the most interesting features of a multi-scale architecture is its ability to process images with arbitrary size (small or large) without loss of performance. Indeed, even if a single-scale model is fully convolutional and so can process high-resolution images, such a model with a fixed number of convolution layers may not have enough convolutions to synthesize the information over a whole, potentially large image. And, a network trained to handle a specific resolution may not behave well for much larger images. For example, information from the bottom left of the image may not be used to infer the normal at the top right. Yet, such an ability would be particularly useful for handling non-local reflectance effects such as translucency. See for instance the acrylic ball shown in the experiments section in Fig. 10, where light passes through the object. By propagating global information at different scales, such a limitation of local methods is overcome.

More importantly, the proposed multi-scale architecture with shared weights allows one to process images with higher resolution than the ones used during training. For example, in our implementation the first processing resolution is 8×8 pixels. By taking a resolution multiplier of two between two scales, four scales are necessary to reach a resolution of 128×128 pixels (which is the training resolution in our tests), and seven scales for the *DiLiGenT10²* images which have a resolution of 1001×1001 pixels. Yet, the same weights are used in both cases, hence a resolution-specific training is not necessary. In practice, this removes the need for either rescaling the input images to the resolution of the training images, or resorting to a (too local) patch-based approach.

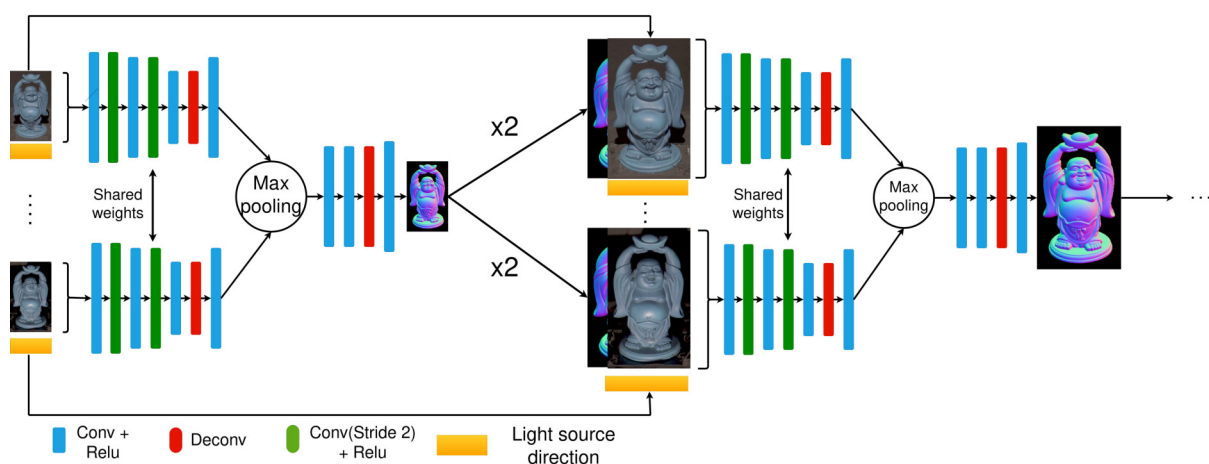


Figure 5: First two stages of the proposed multi-scale architecture. A first architecture, inspired by the PS-FCN method [6], takes as inputs the calibrated lighting directions and downsampled images, and outputs a low-resolution 3D normal map. The latter is then up-sampled and concatenated with lighting directions and higher-resolution images. A second architecture then infers higher-resolution normals, and this part of the process is repeated until the resolution of the original images is reached (network weights being shared by all scales).

4 PROPOSED LEARNING DATASET

As discussed in Section 2, the existing *Blobby* and *Structure* synthetic datasets lack of diversity in terms of geometry and textures. For example, although the *Structure* dataset is composed of complex objects, all these objects are statues. Similarly, the number of different materials in the MERL material base is only 100. This is clearly not enough to model the huge diversity of materials present in the nature. The *CyclePS* dataset partially solves this issue, by allowing to generate infinitely many materials by randomly selecting parameters from a parametric BSDF model. Still, it remains limited in terms of geometry. Overall, a greater diversity of shapes and materials in the images of the training dataset would be beneficial for training networks for photometric stereo. For these reasons, we propose here a new dataset, which includes a large variety of shapes and materials.

In order to create this dataset, we implemented our own image data generation pipeline. We used the *Blender* [8] software with the Cycles rendering engine. As a result, our new dataset is composed of two parts:

- *Our Blobby* contains objects with smooth surfaces;
- *Our Object* contains objects with complex geometry: strong discontinuities, edges, corners, textures details, etc.

Samples from our training dataset are shown in Fig. 6.

Our Blobby has 3000 distinct objects, generated by the sum of random Gaussian potentials, followed by iso-surface extraction using the *Marching Cubes* algorithm [27]. *Our Object* contains 76 detailed objects

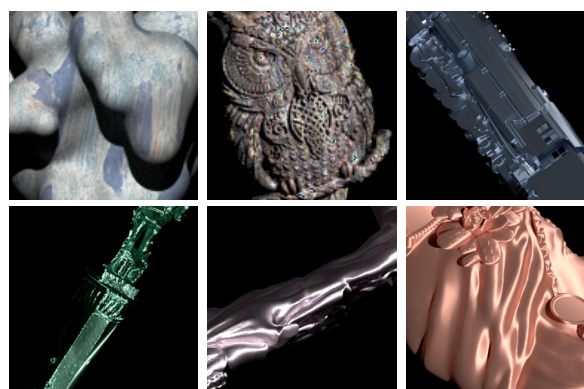


Figure 6: Examples of images from the proposed dataset.

which are 3D meshes from the *Sketchfab* [2] website. Moreover, to allow the learning of non-Lambertian surfaces, more than 1100 different real materials, extracted from the *ambientCG* [1] website, are randomly applied to the objects, much more than the 100 materials of *Structure* and *Blobby*. To complete a lack of diversity of the most complicated materials (metals, glasses, etc.) that could persist, we generated additional materials by randomly setting the values of some parameters (metallic, specular, roughness, anisotropic, etc.) of Disney's principled BSDF [3]. To ensure that all possible materials are represented, during the rendering we choose to apply to the object with a probability of 50% a real material (from ambientCG), with 17% a glass material and with 17% a metal one. The remaining 16% materials are constructed by randomly selecting all possible parameters in the principled BSDF (which may result in non-realistic materials).

	# objects	# views	# total number of samples	# lighting	# materials
<i>Blobby</i>	10	1 296	25 920	64	100
<i>Structure</i>	8	1387-6874	59 292	64	100
<i>CyclePS</i>	18	10	180	1 300	90 000
<i>DiLiGenT</i>	10	1	10	96	10
<i>DiLiGenT10²</i>	10	10	100	100	10
<i>Our Blobby</i>	3000	5	15 000	100	1 100 + infinity
<i>Our Object</i>	76	267	45 000	100	1 100 + infinity

Table 1: Summary of the characteristics of the different learning datasets used in photometric stereo.

If we set a single value for each parameter of the principled BSDF, we would obtain a material which is spatially uniform in terms of reflectance, as in the example of Fig. 7a. Yet, many real-world objects exhibit a spatially-varying reflectance, which is a known limitation of existing PS techniques [6]. To solve this problem in our generation pipeline, we rather incorporated a few spatially-varying material maps, as in the example of Fig. 7b. This technique was used for 50% of the renderings. It allowed us to create both objects with uniform reflectance, and others with spatially-varying one, as illustrated in Fig. 6.

Finally, to generate data having realistic lighting conditions, we rendered all the images with both random illumination direction (Fig. 3c) and random intensity. In total, 15 000 *blobby* samples and 45 000 *object* samples were generated this way. Table 1 summarizes the characteristics of the existing datasets, versus the ones we propose. In order to ensure the reproducibility of our results, the code and these learning datasets will be made publicly available online.

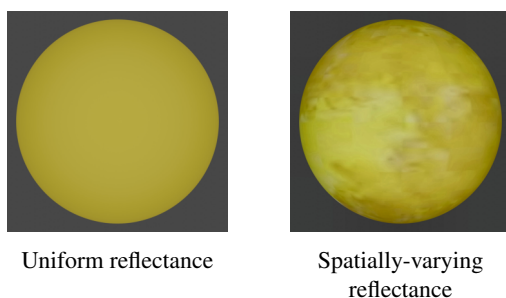


Figure 7: Rendering of the same ball with a uniform base color, or with a spatially-varying one.

5 EXPERIMENTS

In this section, we demonstrate the effectiveness of our proposed multi-scale architecture on publicly available benchmarks, namely DiLiGenT [34] and DiLiGenT10² [31]. To evaluate the impact of our new training dataset, we trained our network both on the pre-existing training datasets *Blobby* and *Structure* (this training is referred to as “DS1” in the following) and on our new training dataset (“DS2” in the following). In the rest of this section, “*Mono* (DS1)” will thus refer to the mono-scale architecture trained on

the pre-existing dataset, “*Multi* (DS1 + DS2)” to the multi-scale architecture trained on both the pre-existing and the new datasets, etc. We will first provide a few qualitative results to illustrate the importance of the two building blocks of our contribution, and then provide a thorough quantitative evaluation on the two benchmarks.

5.1 Implementation details

Both the “*Mono*” and the “*Multi*” architectures were implemented in Pytorch. The Adam optimizer [21] was used with a learning rate of 10^{-4} . We trained both the multi-scale and the mono scale architecture by taking 32 patches of size 128 by 128 as inputs. The training took a few days on a single Nvidia GeForce GTX 1080 Ti with a batch size of 3 (the maximum we can fit in our GPU). The inference time depends on the number of input images and their resolution. For example, by taking 100 images of 256 by 256 pixels, it takes approximately 1.6 seconds for our multi-scale methods on our GPU. The inference time scales linearly with the number of images, while it seems to be roughly multiplied by a factor of 4 when the resolution is multiplied by 2.

5.2 Qualitative evaluation

Let us start by showing two illustrative results on the DiLiGenT10² [31] benchmark, on challenging metallic objects (the copper golf ball and the copper hexagon). As we shall see, both the new training dataset and the new multi-scale architecture contribute to improving the estimation performances on such objects exhibiting an anisotropic reflectance. Since we do not have access to the ground truth normals, for visual purpose we show as “ground truth” the result we obtained with our *Multi* (DS1+DS2) approach, applied to the same object but fabricated in PVC (a matte material). The example of Fig. 8 shows that, independently from the training set, the multi-scale architecture largely contributes to improving the results on metals. In this example, the same dataset is used for training both the mono-scale and multi-scale architectures, and the latter offers visually more accurate results. Likely, the ability of the multi-scale architecture to propagate information in a global manner helps interpreting the anisotropic behavior.

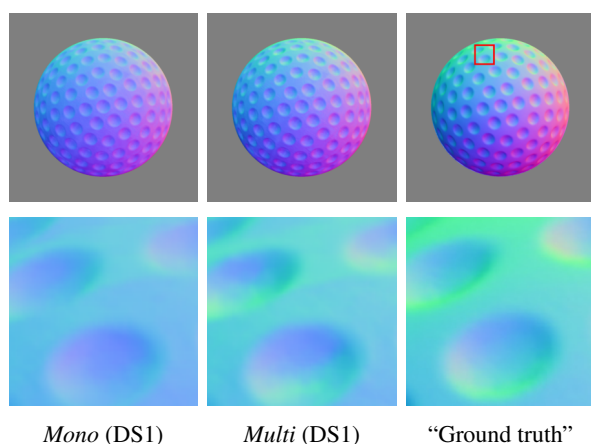


Figure 8: Results of our mono- and multi-scale architectures (both trained on the pre-existing dataset DS1) on the copper golf ball from [31]. The multi-scale architecture yields much sharper results, especially around the holes.

The example of Fig. 9, on the contrary, shows the importance of the presence of metallic objects in the training dataset, independently from the network architecture. It can be observed that the network performs much better when it is trained on our new training dataset, even without considering the multi-scale architecture.



Figure 9: Results of our mono-scale architecture on the copper hexagon from [31]. Since the new dataset (DS2) contains much more metallic objects than the existing one (DS1), training on our new dataset yields largely improved results.

Fig. 10 illustrates a particularly visible improvement brought by the multi-scale architecture, which is the correct handling of translucent materials. In this example, we consider again the golf ball from [31], but this time coated with an acrylic material. Acrylic is a glass-like material, with some of the light passing through the object. As can be seen in the top of Fig. 10, even when light comes from the right side of the ball, part of its left side appears illuminated. Without seeing the whole object the model could not imagine that there exists a path underneath the surface that lets the light go through. On the contrary, the multi-scale approach being global by construction, such non-local phenomena are better managed by the network and the overall reconstruction is clearly more accurate.

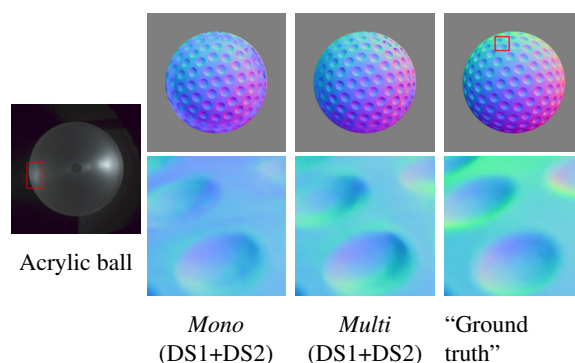


Figure 10: An image of an acrylic ball from [31], illuminated from the right, and results of our mono- and multi-scale architectures (both trained on the new dataset DS2) on the acrylic golf ball from [31]. The reconstruction of translucent objects is improved a lot by using the multi-scale approach.

Others common phenomenas which are cast-shadows and inter-reflections are also better handled by our multi-scale architecture, as Fig. 11 shows.

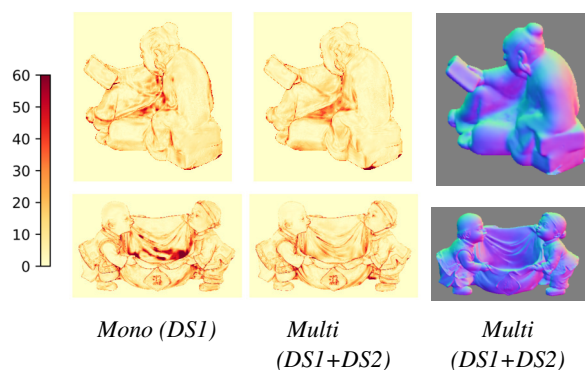


Figure 11: Angular error map and predicted normal map for the “reading” and “harvest” objects from [34]. The concave parts, where cast shadows and inter-reflections occur, are better handled by our approach.

Fig. 12 shows several additional qualitative comparisons of the result obtained with our baseline (mono-scale architecture trained on the existing dataset) and with both our building blocks included (multi-scale architecture trained on the new dataset). The convex objects (*Bunny* and *Propeller*) are very well reconstructed, despite being fabricated with anisotropic (Aluminium) or moderately specular (ABS, a type of plastic) materials. The steel turbine reconstruction is also improved, although on this object our approach shows its limitations. Indeed, this object exhibits concavities, which create many inter-reflections which are not very well handled by the network.

	ball	bear	buddha	cat	cow	goblet	harvest	pot1	pot2	reading	average
L2 (Baseline)[38]	4.10	8.39	14.92	8.41	25.60	18.5	30.62	8.89	14.65	19.80	15.39
GPS-NET [40]	2.92	5.07	7.77	5.42	6.14	9.00	15.14	6.04	7.01	13.58	7.81
CHR-PSN [19]	2.26	6.35	<u>7.15</u>	5.97	6.05	8.32	15.32	7.04	6.76	12.52	7.77
PS-transformer (10 images) [12]	3.27	4.88	8.65	5.34	6.54	9.28	14.41	6.06	6.97	11.24	7.66
MT-PS-CNN [4]	2.29	5.87	6.92	5.79	6.89	6.85	7.88	11.94	7.48	13.71	7.56
PS-FCN [7]	2.67	7.72	7.52	4.75	6.72	7.84	12.39	6.17	7.15	10.92	7.39
CNN-PS [11]	2.2	4.6	7.9	<u>4.1</u>	8.0	7.3	14.0	5.4	6.0	12.6	7.2
Mono (DS1)	<u>2.63</u>	<u>6.66</u>	<u>8.27</u>	<u>4.47</u>	<u>4.77</u>	<u>8.24</u>	<u>12.78</u>	<u>6.00</u>	<u>5.38</u>	<u>9.68</u>	<u>6.88</u>
Multi (DS1)	1.60	7.82	7.55	4.33	4.18	7.85	12.36	5.22	5.36	9.04	6.54
OB-Cnn [10]	2.49	<u>3.59</u>	7.23	4.69	4.89	<u>6.89</u>	12.79	5.10	4.98	11.08	6.37
PX-NET [25]	<u>2.03</u>	3.58	7.61	4.39	4.69	6.90	13.10	<u>5.08</u>	5.10	10.26	<u>6.28</u>
Multi (DS1+DS2)	<u>2.05</u>	<u>4.24</u>	<u>7.03</u>	3.9	4.00	<u>7.57</u>	<u>11.01</u>	4.94	<u>5.22</u>	8.47	5.84

Table 2: Mean angular error (in degrees) on the DiLiGenT [34] benchmark. The best result for each object is indicated in bold, and the second best one is underlined. The lines in blue indicate our results. Combining the proposed multi-scale architecture “Multi” and proposed training dataset “DS2” yields state-of-the-art results, by a large margin.

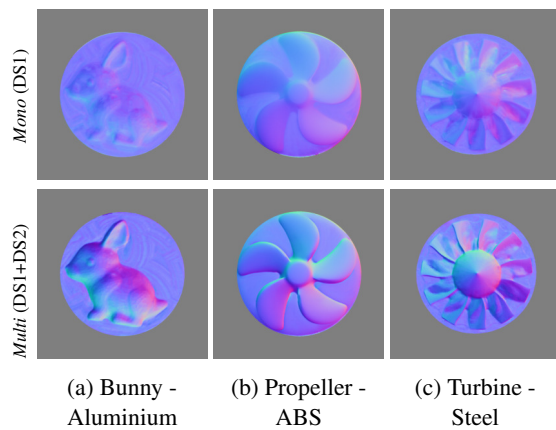


Figure 12: Visual comparison of the improvements brought by the combination of the new architecture and our new training set, on three objects from [31]. All three objects are much better reconstructed, although the steel turbine remains challenging.

5.3 Quantitative evaluation on DiLiGenT [34]

Next, we compare in Table 2 our results against the most recent state-of-the-art methods, on the DiLiGenT benchmark [34]. Let us however remark that PS-transformer [12] takes as inputs no more than 10 images, hence the comparison is biased. Besides, we emphasize that our mono-scale architecture is largely inspired from PS-FCN [6, 7], hence *Mono* (DS1) can be considered as an optimized version of [6, 7], where we let the training phase run for much longer. This table shows that the proposed multi-scale architecture provides a significant gain of 4.6%, in comparison with the mono-scale approach – compare *Mono* (DS1) and *Multi* (DS1). And, as soon as our new training dataset is considered, the state-of-the-art is outperformed and we reach an average angular error below 6°, with a particularly visible improvement on the most difficult “reading” object (Fig. 11).

5.4 Quantitative evaluation on DiLiGenT 10² [31]

We now quantitatively evaluate the impact of the multi-scale architecture on the DiLiGenT 10² benchmark [31]. Note that we process images at their full resolution (1024 pixels by 1024), requiring 7 scales in the multi-scale architecture. To this end, we show in Table 3 the difference between the mono- and the multi-scale approaches, when they are both trained on the pre-existing dataset. As can be observed, a significant gain of 9.3% is observed with the multi-scale architecture. The gain is most visible on objects which have a spherical shape and anisotropic material (top right of Tab 3c, see also Fig. 8 for a qualitative result on the Golf - CU object), as well as for the most challenging “acrylic” material, which is translucent.

mean: 17.77 median: 17.33											
	ball	bear	buddha	cat	cow	goblet	harvest	pot1	pot2	reading	average
BALL	36.0	6.3	12.67	9.7	14.33	10.67	24.33	20.67	24.0	28.67	
GOLF	15.0	14.0	14.0	9.9	14.0	12.0	22.67	17.67	21.0	31.33	
SPIKE	16.33	14.67	15.33	7.52	9.97	14.0	26.33	13.67	26.33	26.67	
NUT	19.0	14.0	19.33	6.78	21.0	11.67	23.67	18.0	20.0	26.0	
SQUARE	21.67	20.33	21.0	18.0	23.0	11.0	23.0	15.67	15.33	25.0	
PENTAGON	20.67	11.33	19.0	9.1	21.0	14.0	21.67	17.33	18.67	19.0	
HEXAGON	18.67	12.67	16.33	6.3	20.67	11.33	24.0	21.33	24.33	26.33	
PROPELLER	19.33	15.0	21.67	9.37	19.67	13.0	15.0	13.0	13.0	18.67	
TURBINE	30.33	17.33	33.0	14.67	32.33	25.0	28.33	25.67	25.0	24.67	
BUNNY	15.0	16.33	17.33	7.6	17.33	13.0	13.0	10.33	11.67	12.67	

(a) *Mono* (DS1)

(b) *Multi* (DS1)

mean: -1.5 median: -1.33											
	ball	bear	buddha	cat	cow	goblet	harvest	pot1	pot2	reading	average
BALL	-0.67	-2.07	-1.0	-0.3	-1.0	-1.0	-2.33	-2.0	-2.33	-2.67	
GOLF	-0.33	-1.33	-1.33	-1.2	-1.0	-2.0	-2.0	-1.33	-1.33	-2.0	
SPIKE	-1.33	-2.0	-1.0	-0.5	-0.14	-1.0	-2.0	-2.0	-2.0	-2.0	
NUT	-1.0	-2.0	-1.0	-0.9	-0.7	-1.0	-2.0	-2.0	-2.0	-2.0	
SQUARE	-1.0	-1.33	-0.33	-0.67	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	
PENTAGON	0.66	-1.0	1.67	0.87	-1.0	1.67	1.67	-2.0	-1.34	1.67	
HEXAGON	0.0	1.67	0.34	-0.17	-2.34	-2.06	-2.0	-1.0	-2.0	-2.0	
PROPELLER	0.0	-2.0	0.66	-0.87	-1.34	-1.0	-0.33	-0.67	-0.67	0.33	
TURBINE	2.0	4.0	4.0	-0.33	-1.0	-1.0	-1.0	-0.33	0.0	0.0	
BUNNY	1.0	-0.76	0.67	-0.63	-0.33	-1.5	0.0	-0.56	-0.67	0.33	

(c) *Multi* (DS1) - *Mono* (DS1)

Table 3: Mean angular on the DiLiGenT10² benchmark, considering either the mono-scale architecture or the multi-scale one, both trained on the pre-existing dataset DS1. The multi-scale approach yields a significant gain, most visible on the top-right part of the table (spherical shapes with anisotropic reflectance).

mean: 15.35 median: 14.05											
	POM	PP	NYLON	PVC	ABS	ACRYLIC	AI	CU	STEEL	ALUMINUM	
BALL	9.3	5.0	8.4	7.6	9.7	6.0	18.0	16.0	22.0	22.0	
GOLF	11.0	7.1	10.0	6.1	10.0	6.9	13.0	9.8	14.0	21.0	
SPIKE	11.0	7.8	10.0	7.2	8.6	8.0	20.0	11.0	20.0	30.0	
NUT	19.0	11.0	18.0	7.7	15.0	11.0	19.0	14.0	17.0	26.0	
SQUARE	19.0	10.0	19.0	11.0	15.0	8.7	17.0	9.5	13.0	18.0	
PENTAGON	22.0	12.0	21.0	10.0	18.0	13.0	17.0	14.0	16.0	22.0	
HEXAGON	18.0	9.8	17.0	8.8	14.0	8.8	18.0	12.0	18.0	23.0	
PROPELLER	23.0	12.0	24.0	9.6	19.0	12.0	14.0	12.0	13.0	14.0	
TURBINE	36.0	18.0	38.0	14.0	33.0	22.0	29.0	25.0	27.0	26.0	
BUNNY	18.0	11.0	19.0	9.2	15.0	11.0	14.0	12.0	12.0	14.0	

(a) Mono (DS1+DS2)

mean: 11.33 median: 9.98											
	POM	PP	NYLON	PVC	ABS	ACRYLIC	AI	CU	STEEL	ALUMINUM	
BALL	9.3	3.4	8.7	5.2	8.4	4.3	8.5	12.0	14.0	8.6	
GOLF	10.0	7.3	9.8	5.8	10.0	6.87	7.9	7.7	9.8	12.0	
SPIKE	12.0	8.8	9.9	6.3	8.5	7.9	12.0	7.6	12.0	17.0	
NUT	14.0	8.9	15.0	5.8	10.0	5.8	9.2	7.6	8.2	16.0	
SQUARE	18.0	11.0	17.0	8.2	14.0	5.5	12.0	7.2	7.9	11.0	
PENTAGON	18.0	8.4	17.0	8.0	17.0	9.4	11.0	9.4	13.0	20.0	
HEXAGON	16.0	7.5	15.0	6.1	13.0	7.1	11.0	8.1	11.0	20.0	
PROPELLER	13.0	8.9	11.0	7.9	16.0	9.7	11.0	8.4	8.2	19.0	
TURBINE	21.0	12.0	24.0	11.0	18.0	15.0	23.0	16.0	18.0	22.0	
BUNNY	17.0	8.2	16.0	6.5	12.0	8.3	8.6	7.3	8.0	18.0	

(b) Multi (DS1+DS2)

mean: 15.78 median: 13.99											
	POM	PP	NYLON	PVC	ABS	ACRYLIC	AI	CU	STEEL	ALUMINUM	
BALL	5.1	6.4	4.2	4.5	6.9	7.3	16.0	14.0	16.0	19.0	
GOLF	14.0	8.0	12.0	6.8	14.0	9.4	12.0	9.2	13.0	22.0	
SPIKE	11.0	9.4	11.0	11.0	12.0	9.5	14.0	8.3	16.0	28.0	
NUT	20.0	8.8	19.0	6.9	17.0	8.0	16.0	13.0	14.0	22.0	
SQUARE	21.0	8.1	22.0	6.7	19.0	8.1	13.0	4.9	7.9	18.0	
PENTAGON	26.0	9.5	26.0	9.8	22.0	9.6	15.0	13.0	15.0	23.0	
HEXAGON	18.0	7.5	19.0	7.2	17.0	28.0	18.0	10.0	17.0	21.0	
PROPELLER	28.0	12.0	35.0	8.4	23.0	11.0	16.0	9.6	9.8	17.0	
TURBINE	34.0	20.0	51.0	16.0	39.0	21.0	25.0	22.0	21.0	32.0	
BUNNY	24.0	11.0	27.0	7.8	21.0	9.1	12.0	7.7	12.0	14.0	

(c) CNN-PS [11] (DS1)

Table 4: Mean angular error on the *DiLiGenT10*² benchmark, with the results of CNN-PS [11] indicated for comparison. When incorporating both the new dataset and the multi-scale architecture, the state-of-the-art is largely outperformed.

We repeat this experiment in Table 4, but this time with our networks trained on the new dataset. Comparing Tables 3 and 4 allows one to quantify the benefits of using our new training dataset: the mono-scale architecture gets improved by 14%, and the multi-scale one by 30%. Comparing Tables 4a and 4b also allows one to quantify the impact of switching to the multi-scale architecture: the results improve by 26%. Particularly large improvements can be observed on the *Turbine* and *Acrylic Gulf* objects (see also Figs. 12c and 10). For such objects with non-local light transport (due to inter-reflections or anisotropic reflectance), the ability of the multi-scale approach to get access to a global information is indeed of primary importance.

Overall, the combination of the new architecture and dataset allows one to reach an average error of 11.33° on this benchmark. This is to be compared with the 15.78° achieved by CNN-PS [11] (Table 4c), which was the best performing method so far [31]. By comparing our results with all available state-of-the-art methods [5, 6, 11, 30, 33, 35, 36, 39, 40, 41], we found out that the proposed method is the best performer on 73% of the objects of this benchmark, as indicated in Table 5.

	POM	PP	NYLON	PVC	ABS	ACRYLIC	AI	CU	STEEL	ALUMINUM
BALL	9.3	3.4	8.7	5.2	8.4	4.3	8.5	12.0	14.0	8.6
GOLF	10.0	7.3	9.8	5.8	10.0	6.87	7.9	7.7	9.8	12.0
SPIKE	12.0	8.8	9.9	6.3	8.5	7.9	12.0	7.6	12.0	17.0
NUT	14.0	8.9	15.0	5.8	10.0	5.8	9.2	7.6	8.2	16.0
SQUARE	18.0	11.0	17.0	8.2	14.0	5.5	12.0	7.2	7.9	11.0
PENTAGON	18.0	8.4	17.0	8.0	17.0	9.4	11.0	9.4	13.0	20.0
HEXAGON	16.0	7.5	15.0	6.1	13.0	7.1	11.0	8.1	11.0	20.0
PROPELLER	13.0	8.9	11.0	7.9	16.0	9.7	11.0	8.4	8.2	19.0
TURBINE	21.0	12.0	24.0	11.0	18.0	15.0	23.0	16.0	18.0	22.0
BUNNY	17.0	8.2	16.0	6.5	12.0	8.3	8.6	7.3	8.0	18.0

Table 5: Mean angular error achieved by the best performer among [5, 6, 11, 30, 33, 35, 36, 39, 40, 41] and us, on the 100 objects of [31]. Green cases indicate when the proposed architecture, combined with the new dataset, gives the best results.

5.5 Limitations

Even if the combination of our multi-scale and our new training dataset improves the results on non-Lambertian materials, some shortcomings remain. For example, we notice that the normals at the border of some translucent objects are incorrectly predicted (Fig. 13). As shown in Fig. 14, in this example the opposite side of the incoming light is the most shiny part of the image. Although our multi-scale approach better handles such anisotropic than the mono-scale one or existing methods such as CNN-PS, it shows its limitations when the anisotropy is this much important.

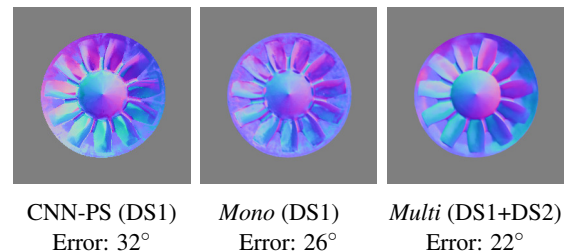


Figure 13: Results of CNN-PS, our mono-scale and our multi-scale architecture on the acrylic turbine.

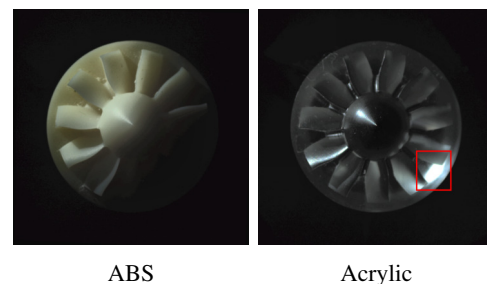


Figure 14: Same turbine, fabricated either with a diffuse (ABS) or an anisotropic (acrylic) material, and illuminated from the same direction (coming from “top left”). The bottom-right area, which is shadowed in the diffuse case, appears much shinier on the anisotropic object.

6 CONCLUSION

In this paper, we have proposed a novel deep normal estimation framework for the calibrated photometric stereo problem. It builds upon a multi-scale architecture which is independent from the resolution of the images, as well as a new comprehensive learning dataset. We have shown on publicly available benchmarks that the combination of these two features yields state-of-the-art results, with performances particularly improved on challenging anisotropic materials. In the future, we plan to extend our approach to handle observation maps [11] as well, which have recently been shown to benefit from physical interpretability [13].

7 REFERENCES

- [1] AmbientCG. <https://ambientcg.com/>.
- [2] Sketchfab. <https://sketchfab.com>.
- [3] B. Burley and W. D. Studios. Physically-based shading at Disney. *ACM SIGGRAPH Courses*, 2012.
- [4] Y. Cao, B. Ding, Z. He, J. Yang, J. Chen, Y. Cao, and X. Li. Learning inter- and intraframe representations for non-Lambertian photometric stereo. *OLEN*, 150:106838, 2022.
- [5] G. Chen, K. Han, B. Shi, Y. Matsushita, and Kwan-Yee K. Wong. Self-Calibrating Deep Photometric Stereo Networks. In *CVPR*, 2019.
- [6] G. Chen, K. Han, and K. Wong. PS-FCN: A Flexible Learning Framework for Photometric Stereo. In *ECCV*, 2018.
- [7] G. Chen, Kai Han, Boxin S., Yasuyuki M., and K. W. Deep Photometric Stereo for Non-Lambertian Surfaces. *PAMI*, 44(1), 2022.
- [8] Blender Online Community. *Blender - a 3D modelling and rendering package*, 2018.
- [9] B. Haefner, Z. Ye, M. Gao, T. Wu, Y. Quéau, and D. Cremers. Variational uncalibrated photometric stereo under general lighting. In *ICCV*, 2019.
- [10] D. Honzátka, E. Türetken, P. Fua, and L. Dunbar. Leveraging Spatial and Photometric Context for Calibrated Non-Lambertian Photometric Stereo. In *3DV*, 2021.
- [11] S. Ikehata. CNN-PS: CNN-based Photometric Stereo for General Non-Convex Surfaces. In *ECCV*, 2018.
- [12] S. Ikehata. PS-transformer: Learning sparse photometric stereo network using self-attention mechanism. In *BMVC*, 2021.
- [13] S. Ikehata. Does Physical Interpretability of Observation Map Improve Photometric Stereo Networks? In *ICIP*, 2022.
- [14] S. Ikehata. Universal photometric stereo network using global lighting contexts. *CVPR*, 2022.
- [15] M. Johnson and E. Adelson. Shape Estimation in Natural Illumination. In *CVPR*, 2011.
- [16] Y. Ju, J. Dong, and S. Chen. Recovering Surface Normal and Arbitrary Images: A Dual Regression Network for Photometric Stereo. *TIP*, 30:3676–3690, 2021.
- [17] Y. Ju, M. Jian, J. Dong, and K. Lam. Learning Photometric Stereo via Manifold-based Mapping. In *VCIP*, 2020.
- [18] Y. Ju, K. Lam, Y. Chen, L. Qi, and J. Dong. Pay Attention to Devils: A Photometric Stereo Network for Better Details. In *IJCAI*, 2020.
- [19] Y. Ju, Y. Peng, M. Jian, F. Gao, and J. Dong. Learning conditional photometric stereo with high-resolution features. *CVM*, 8(1):105–118, 2022.
- [20] B. Kaya, S. Kumar, C. Oliveira, V. Ferrari, and Van G. Uncalibrated Neural Inverse Rendering for Photometric Stereo of General Surfaces. In *CVPR*, 2021.
- [21] Diederik P Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *ICLR*, 2015.
- [22] J. Li, A. Robles-Kelly, S. You, and Y. Matsushita. Learning to Minify Photometric Stereo. In *CVPR*, 2019.
- [23] D. Lichy, S. Sengupta, and D. Jacobs. Fast light-weight near-field photometric stereo. In *CVPR*, 2022.
- [24] D. Lichy, J. Wu, S. Sengupta, and D. Jacobs. Shape and Material Capture at Home. In *CVPR*, 2021.
- [25] F. Logothetis, I. Budvytis, R. Mecca, and R. Cipolla. PX-net: Simple and efficient pixel-wise training of photometric stereo networks. In *ICCV*, 2021.
- [26] F. Logothetis, R. Mecca, I. Budvytis, and R. Cipolla. A CNN based approach for the point-light photometric stereo problem. *IJCV*, 131(1):101–120, 2023.
- [27] W. Lorensen and H. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH*, 1987.
- [28] W. Matusik. *A data-driven reflectance model*. PhD thesis, Massachusetts Institute of Technology, 2003.
- [29] Z. Mo, B. Shi, F. Lu, S.-K. Yeung, and Y. Matsushita. Uncalibrated photometric stereo under natural illumination. In *CVPR*, 2018.
- [30] T. Papadimitri and P. Favaro. A Closed-Form, Consistent and Robust Solution to Uncalibrated Photometric Stereo Via Local Diffuse Reflectance Maxima. *IJCV*, 2014.
- [31] J. Ren, F. Wang, J. Zhang, Q. Zheng, M. Ren, and B. Shi. DiLiGenT10²: A Photometric Stereo Benchmark Dataset with Controlled Shape and Material Variation. In *CVPR*, 2022.
- [32] H. Santo, M. Waechter, and Y. Matsushita. Deep near-light photometric stereo for spatially varying reflectances. In *ECCV*, 2020.
- [33] B. Shi, P. Tan, Y. Matsushita, and K. Ikeuchi. Bi-polynomial modeling of low-frequency reflectances. *PAMI*, 36(6):1078–1091, 2013.
- [34] B. Shi, Z. Wu, Z. Mo, D. Duan, S. Yeung, and P. Tan. A Benchmark Dataset and Evaluation for Non-Lambertian and Uncalibrated Photometric Stereo. In *CVPR*, 2016.
- [35] B. Shi, Z. Wu, Z. Mo, D. Duan, S. Yeung, and P. Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. In *CVPR*, 2016.
- [36] T. Tani and T. Maehara. Neural Inverse Rendering for General Reflectance Photometric Stereo. In *ICML*, 2018.
- [37] X. Wang, Z. Jian, and M. Ren. Non-Lambertian Photometric Stereo Network Based on Inverse Reflectance Model With Collocated Light. *TIP*, 29, 2020.
- [38] R. J. Woodham. Photometric Method For Determining Surface Orientation From Multiple Images. *Opt. Eng.*, 19, 1980.
- [39] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma. Robust photometric stereo via low-rank matrix completion and recovery. In *ACCV*, 2011.
- [40] Z. Yao, K. Li, Y. Fu, H. Hu, and B. Shi. GPS-Net: Graph-based Photometric Stereo Network. In *NIPS*, 2020.
- [41] Q. Zheng, Y. Jia, B. Shi, X. Jiang, L. Duan, and A. Kot. SPLINE-Net: Sparse Photometric Stereo Through Lighting Interpolation and Normal Estimation Networks. In *ICCV*, 2019.

Fast Incremental Image Reconstruction with CNN-enhanced Poisson Interpolation

Blaž Erzar

University of Ljubljana
Faculty of Computer and
Information Science
Večna pot 113
1000 Ljubljana, Slovenia
be6384@student.uni-lj.si

Žiga Lesar

University of Ljubljana
Faculty of Computer and
Information Science
Večna pot 113
1000 Ljubljana, Slovenia
ziga.lesar@fri.uni-lj.si

Matija Marolt

University of Ljubljana
Faculty of Computer and
Information Science
Večna pot 113
1000 Ljubljana, Slovenia
matija.marolt@fri.uni-lj.si

ABSTRACT

We present a novel image reconstruction method from scattered data based on multigrid relaxation of the Poisson equation and convolutional neural networks (CNN). We first formulate the image reconstruction problem as a Poisson equation with irregular boundary conditions, then propose a fast multigrid method for solving such an equation, and finally enhance the reconstructed image with a CNN to recover the details. The method works incrementally so that additional points can be added, and the amount of points does not affect the reconstruction speed. Furthermore, the multigrid and CNN techniques ensure that the output image resolution has only minor impact on the reconstruction speed. We evaluated the method on the CompCars dataset, where it achieves up to 40% error reduction compared to a reconstruction-only approach and 9% compared to a CNN-only approach.

Keywords

Image reconstruction, numerical interpolation, multigrid method, convolutional neural networks, autoencoder.

1 INTRODUCTION

Image reconstruction is a process used to recover the complete data from the incomplete ones that form scattered data. Reconstruction can be applied to both three-dimensional point clouds and two-dimensional images. In this paper we focus on reconstructing images which are generated out of input scattered data. Hereafter, we refer to these images as *corrupted*, although the data may be missing for a variety of reasons, e.g., errors in transfer between different systems, missing data before the transfer.

Missing data can be the result of a desire to save time or space, since in some cases the generation of data is resource heavy. One such example is ray tracing for rendering three-dimensional data. Despite the current powerful graphics processors, the method is still time consuming. The reason for this is the need to simulate the reflections of the light rays for each pixel in order to calculate its colour in the final image. This need for computation power could be lowered by simulating

only a small fraction of the rays and reconstructing the rest of the pixels.

Similar idea is used in *foveated rendering* [Jab+22], which is used in virtual reality. Here eye tracking is used to monitor where the user's view is focused. Most of the pixels on the screen are in the peripheral vision, where the sharpness is lower as in the central area. This means that fewer rays could be sent over those areas (or areas containing less details) and the reduction of image quality will not be noticed by the user.

We solve the reconstruction problem using Poisson interpolation, which allows us to use numerical methods for solving linear systems. These methods run fast on the GPU and also converge in small number of iterations, given we choose multigrid method used in this paper. Poisson interpolation allows us to use some other method for generating the first approximation of the solution, e.g. *ray tracing*. Image generated on a very small amount of rays could be interpolated to generate a fast preview or noise in scattered data could be removed since interpolation creates a smoother image. This way we would get a direct preview of the rendering process, since we can run reconstruction concurrently along with ray tracing.

2 THEORETICAL BACKGROUND

First we introduce some theoretical background and notation used in solving the reconstruction problem, which is solved using linear systems of equations.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2.1 Reconstruction

Let $\hat{\phi} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ be an image made of colour pixels $\mathbf{u}_{ij} \in \mathbb{R}^3$. It is defined on a rectangular grid of size $n \times m$ in points (x_i, y_j) :

$$\hat{\phi}(x_i, y_j) = \mathbf{u}_{ij}, \quad (1)$$

for indices $i, j \in \mathbb{Z}$, $0 \leq i < m$ and $0 \leq j < n$. Distance between neighbouring pixels in x dimension is defined as $h_x = 1/(m-1)$ and analogous for the y dimension.

We have a corrupted image ϕ , which we want to reconstruct. Because the image is corrupted, it contains actual pixel values only for points in the set of boundary conditions R , $|R| \ll nm$. The corrupted image is then defined as

$$\phi(x_i, y_j) = \begin{cases} \mathbf{u}_{ij}, & \text{if } (x_i, y_j) \in R, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

As the result of the reconstruction, we want a smooth image. We can model this using **Laplace's equation** $\Delta\phi = \nabla^2\phi = \phi_{xx} + \phi_{yy} = 0$, which is a partial differential equation [Str07b]. Multigrid solver actually solves the nonhomogenous version of this equation – **Poisson's equation**:

$$\Delta\phi = f, \quad (2)$$

which we will be solving from here on out.

2.2 Linear system

To be able to write Poisson's equation as a system of linear equations, it needs to be discretized first. This can be achieved using *finite difference* method for approximations of the partial derivatives. Using *backward difference* followed by the *forward difference*, the second partial derivative ϕ_{xx} can be approximated as

$$\phi_{xx} \approx \frac{\phi(x_i + h_x, y_j) - 2\phi(x_i, y_j) + \phi(x_i - h_x, y_j)}{h_x^2}$$

an analogous for ϕ_{yy} .

Using these approximations we can evaluate the left-hand side of (2) where we use h instead of h_x and h_y since grid size in both dimensions is the same:

$$\begin{aligned} \Delta\phi(x_i, y_j) \approx & \frac{1}{h^2} [\phi(x_i + h, y_j) + \phi(x_i - h, y_j) \\ & + \phi(x_i, y_j + h) + \phi(x_i, y_j - h) \\ & - 4\phi(x_i, y_j)]. \end{aligned} \quad (3)$$

This way we get *five-point centered approximation of the Laplacian* that can also be evaluated using a convolution with a 2D kernel.

By approximation (3) and (1) – which also hold for the function f – we write the **discrete Poisson's equation**:

$$\frac{\mathbf{u}_{i+1,j} + \mathbf{u}_{i-1,j} + \mathbf{u}_{i,j+1} + \mathbf{u}_{i,j-1} - 4\mathbf{u}_{ij}}{h^2} = \mathbf{f}_{ij} \quad (4)$$

that can be solved as a linear system.

After multiplying (4) by $-h^2$ on both sides, we write the system $\mathbf{A}\mathbf{u} = \mathbf{b}$. Vectors \mathbf{u} and \mathbf{b} are formed by writing the image and function \mathbf{f} as a vector row wise:

$$\begin{aligned} \mathbf{u} &= [\mathbf{u}_{11}, \mathbf{u}_{21}, \dots, \mathbf{u}_{m1}, \mathbf{u}_{12}, \dots, \mathbf{u}_{m2}, \dots, \mathbf{u}_{mn}]^T, \\ \mathbf{b} &= -h^2[\mathbf{f}_{11}, \mathbf{f}_{21}, \dots, \mathbf{f}_{m1}, \mathbf{f}_{12}, \dots, \mathbf{f}_{m2}, \dots, \mathbf{f}_{mn}]^T. \end{aligned}$$

The matrix of the system \mathbf{A} is a tridiagonal block matrix [GV96]:

$$\mathbf{A} = \begin{bmatrix} \mathbf{C} & -\mathbf{I} & & \\ -\mathbf{I} & \mathbf{C} & \ddots & \\ & \ddots & \ddots & -\mathbf{I} \\ & & -\mathbf{I} & \mathbf{C} \end{bmatrix} \in \mathbb{R}^{n \times m},$$

where $\mathbf{I} \in \mathbb{R}^{m \times m}$ denotes the identity matrix and \mathbf{C} the tridiagonal matrix:

$$\mathbf{C} = \begin{bmatrix} 4 & -1 & & \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix} \in \mathbb{R}^{m \times m}.$$

Matrix \mathbf{A} can also be decomposed as a sum of three matrices, diagonal matrix \mathbf{D} , lower triangular matrix \mathbf{L} and upper triangular matrix \mathbf{U} :

$$\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U}. \quad (5)$$

This decomposition is used in defining iterative methods.

3 RELATED WORK

The linear system $\mathbf{A}\mathbf{u} = \mathbf{b}$ can be solved using two different approaches. *Direct methods* [Wen17, chap. 3] solve it in a finite number of steps. Examples of direct methods are Gaussian elimination, LU decomposition, pivoting etc. These methods always return a solution – as long as the system has a solution – but they have high time and space complexity. The latter is in this case more problematic, since the matrix \mathbf{A} is a very sparse matrix. For this system of size nm , the Gaussian elimination has time complexity $\mathcal{O}(n^3m^3)$ and space complexity $\mathcal{O}(n^2m^2)$. On the other hand, *iterative methods* [Wen17, chap. 4] solve the system by generating approximate solutions that converge towards the final one. In each iteration, the next approximation is generated from the previous one. For a system of size nm each iteration typically has time complexity $\mathcal{O}(n^2m^2)$. This means that the iterative methods returns a solution faster than the direct method, as long as a good solution is obtained in less than nm steps. Moreover, to use iterative methods, we do not need to explicitly generate the

matrix \mathbf{A} , because we only need it to derive a formula to generate the next approximations.

Defining iterative methods as in [GQ20], each iterative method has its *iteration matrix* \mathbf{B} and vector \mathbf{c} , which are used to calculate every successive approximation given the previous one:

$$\mathbf{u}_{k+1} = \mathbf{B}\mathbf{u}_k + \mathbf{c}, \quad k \in \mathbb{N}_0.$$

The sequence of approximation converges towards the true solution $\tilde{\mathbf{u}}$, which minimizes the *residual* defined for the approximation \mathbf{u} as

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{u}.$$

3.1 Jacobi method

One of the simplest iterative methods is the *Jacobi method* [DF18]. Its iteration matrix \mathbf{B} and vector \mathbf{c} are

$$\begin{aligned} \mathbf{B} &= -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}), \\ \mathbf{c} &= \mathbf{D}^{-1}\mathbf{b}. \end{aligned}$$

Given this we can write down the rule for updating pixel values:

$$\mathbf{u}_{ij}^{(k+1)} = \frac{1}{4} \left(\mathbf{u}_{i-1,j}^{(k)} + \mathbf{u}_{i,j-1}^{(k)} + \mathbf{u}_{i+1,j}^{(k)} + \mathbf{u}_{i,j+1}^{(k)} - h^2 \mathbf{f}_{ij} \right).$$

3.2 Gauss-Seidel method

The *Gauss-Seidel method* [DF18] is similar to the Jacobi method, but it has slightly better convergence. This is achieved by using values from iteration $k+1$ when calculating the values of iteration $k+1$. The matrix \mathbf{B} and vector \mathbf{c} are

$$\begin{aligned} \mathbf{B} &= -(\mathbf{D} + \mathbf{L})^{-1}\mathbf{U}, \\ \mathbf{c} &= (\mathbf{D} + \mathbf{L})^{-1}\mathbf{b}, \end{aligned}$$

while the update rule is

$$\mathbf{u}_{ij}^{(k+1)} = \frac{1}{4} \left(\mathbf{u}_{i-1,j}^{(k+1)} + \mathbf{u}_{i,j-1}^{(k+1)} + \mathbf{u}_{i+1,j}^{(k)} + \mathbf{u}_{i,j+1}^{(k)} - h^2 \mathbf{f}_{ij} \right).$$

It is almost exactly the same as Jacobi's update, but here the first two terms are from the currently calculated iteration $k+1$ instead of the already calculated iteration k .

3.3 Successive over-relaxation

By introducing the relaxation parameter ω the *successive over-relaxation (SOR)* [QSS07] can be derived, whose matrix \mathbf{B} and vector \mathbf{c} are

$$\begin{aligned} \mathbf{B} &= -(\mathbf{D} + \omega\mathbf{L})^{-1}[(\omega - 1)\mathbf{D} + \omega\mathbf{U}], \\ \mathbf{c} &= (\mathbf{D} + \omega\mathbf{L})^{-1}\omega\mathbf{b}. \end{aligned}$$

While deriving the update rule it can be shown, that the next SOR approximation is actually a linear combination of the previous approximation and approximation calculated using the Gauss-Seidel method:

$$\mathbf{u}_{ij}^{(k+1)} = (1 - \omega)\mathbf{u}_{ij}^{(k)} + \omega\mathbf{u}_{ij,GS}^{(k+1)}.$$

By theorem 9.6 in [GQ20], the method converges under the condition $0 < \omega < 2$.

3.4 Conjugate gradient method

The *conjugate gradient method (CG)* [Wen17, chap. 6] is different from the previous ones, since it is not derived from the matrix decomposition (5). It is based on the same idea as gradient descend.

In every iteration the next approximation is calculated by

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \mathbf{p}_k,$$

which represents a move in the direction of the **conjugate gradient** \mathbf{p}_k , that is defined to be \mathbf{A} -conjugate to all other conjugate gradients. After defining $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{u}_k$ as the residual of the current approximation, the length of the next move can be calculated using

$$\alpha_k = \frac{\mathbf{p}_k^T \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k}.$$

The next residual can then be calculated as

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k$$

and the next conjugate gradient as

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$$

using

$$\beta_k = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k}.$$

It is chosen that $\mathbf{p}_0 = \mathbf{r}_0$, like in the gradient descend case.

3.5 Neural networks

Recently many new neural networks based on diffusion have been released. Models like Stable Diffusion [Rom+21], DALL-E [Ram+22] or Imagen [Sah+22] take a text prompt and generate an image based on its input. Some of these models even allow users to input an image and generate similar images. This approach could be used for some sort of reconstruction, but these models are usually very big. Stable Diffusion for example has around 890 million parameters, but some models are even bigger. Only the forward pass on these models requires a powerful computer. In contrast the model developed for this paper has less than a million parameters and can be run on a desktop computer with a dedicated graphics card. A smaller network runs faster and can be used alongside a fast reconstruction method.

4 METHOD

4.1 Multigrid solver

The problem of the methods described till now is slow convergence, which is the consequence of errors consisting of high and low frequencies. High frequencies are removed in a few iterations, while the low ones are being removed slowly. The idea of the *multigrid method (MG)* [Str07a, chap. 7.3] is to use grids of multiple resolutions, where low frequencies become high. The multigrid method is not a standalone method, but rather a high-level scheme for using the existing relaxation methods.

Since the method operates on grids of different resolutions, an operation is needed which generates a grid of lower resolution – **coarse grid** – from a higher resolution grid – **fine grid** – and the other way around. For simplicity, we will restrict ourselves to $n \times n$ quadratic images whose size is a power of 2. This means that we can reduce the image down to a 1×1 grid.

It is a recursive method, as it is also used to solve systems at lower levels. Depending on the order in which these systems are solved, we obtain different iteration schemes called *cycles*.

4.1.1 Restriction

The operation that reduces the size of the grid is called *restriction*. For an image \mathbf{u} (or residual \mathbf{r}), it returns an image \mathbf{u}' , which has a fourth of the input image pixels:

$$\text{restriction} : \mathbb{R}^{n^2} \rightarrow \mathbb{R}^{n^2/4}.$$

It works by calculating one pixel value in the restricted image as the mean of four pixel values from the input image. This operation can be performed efficiently on a GPU.

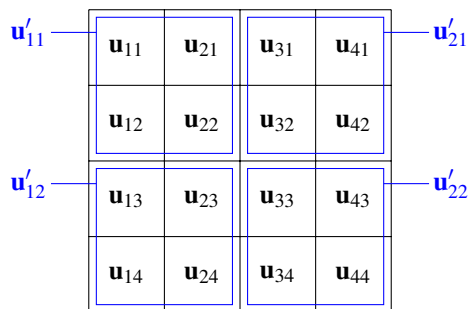


Figure 1: Restriction on image of size 4×4 .

4.1.2 Interpolation

To increase the size of the grid, *interpolation* operation is used, which creates a grid twice the resolution:

$$\text{interpolation} : \mathbb{R}^{n^2} \rightarrow \mathbb{R}^{4n^2}.$$

This is done using bilinear interpolation. This operation can be performed efficiently on a GPU.

4.1.3 Boundary conditions

In single-grid methods the boundary conditions are simple to account for – we do not update the pixels \mathbf{u}_{ij} for which $(x_i, y_j) \in R$, but we still use them for updating other pixels. In the multigrid method, this only works on the first grid level. On other levels, a different system is solved, which also has different dimensions. As we will see, this is an error system, so the boundary conditions are homogeneous – their value is 0.

However, because of the different dimensions, we have a problem, because we do not know which pixels fall under the boundary conditions. The solution, as proposed in [GT11], is that a point on the coarse grid becomes a boundary condition if it has been constructed from at least one boundary condition point on the fine grid in the restriction process, as it can be seen in Figure 2.

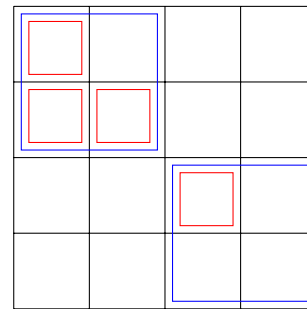


Figure 2: Example of boundary conditions on fine (red) and coarse (blue) grid.

Implementing this logic is fairly simple. Along the image \mathbf{u} we also need an image \mathbf{m} of the same dimensions, which holds the boundary conditions. Its elements are -1 where pixel is a boundary condition and 1 everywhere else. During the iteration of the multigrid method, we also apply restriction operation over \mathbf{m} . The result is that for boundary conditions (x_i, y_j) on all levels, $\mathbf{m}_{ij} < 1$, as the result of restriction will be 1 only if it has been calculated from four pixels which are not a boundary condition and have a value of 1 . If pixel is built from at least one pixel which is a boundary condition, the mean of elements of \mathbf{m} during restriction will certainly be less than 1 .

4.1.4 V-cycle

There exist different cycles (iteration schemes) of the multigrid method. The individual steps are the same for all cycles, but they differ in the order in which grids of different resolutions are considered. We use the V-cycle.

It is the simplest of all cycles and has very good convergence. Other cycles increase the running time of one iteration, but do not bring much improvement in convergence. We implement it on the GPU and it runs upwards of 100 times faster than the CPU implementation. Its steps are:

1. **Pre-smoothing:** Perform N_{sm} iterations of a single-grid method over the system $\mathbf{A}\mathbf{u} = \mathbf{b}$.
2. **Restriction:** Restrict (downsample) the residual \mathbf{r} to a coarse grid to get \mathbf{r}' .
3. **Solve:** We solve $\mathbf{A}\mathbf{e} = \mathbf{r}'$. If the size of grid is larger than n_{min} , the V-cycle is invoked recursively, otherwise N_{so} iterations of a single-grid method are performed.
4. **Interpolation:** Interpolate (upsample) solution \mathbf{e} to a fine grid to obtain correction \mathbf{p} .
5. **Post-smoothing:** Perform N_{sm} iterations of smoothing over the improved solution $\mathbf{u} + \mathbf{p}$.

For the smoothing we use SOR implemented since its iteration takes similar amount of time as the simple Jacobi method, but converges much faster. We implement it on the GPU using *red-black ordering* [Str07a, p. 568], because we cannot read and write to the same GPU memory at the same time. The value of N_{sm} is 20, while N_{so} is 10. For n_{min} we take 1. Another thing we need to be careful about is step 5 – since the correction is interpolated, it does not take boundary conditions into account, so we only need to apply the correction to points that are **not** boundary conditions on the fine grid.

Algorithm 1 V-cycle

Input: $\mathbf{u}_k, \mathbf{f}, \mathbf{m}, n$

Output: \mathbf{u}_{k+1}

$\mathbf{u}_{k+1} \leftarrow \text{smoothing}(\mathbf{u}_k, \mathbf{f}, \mathbf{m}, n, N_{sm})$

$\mathbf{r} \leftarrow \text{residual}(\mathbf{u}_{k+1}, \mathbf{f}, \mathbf{m}, n)$

$\mathbf{r}' \leftarrow \text{restriction}(\mathbf{r})$

$\mathbf{e} \leftarrow \text{empty image}$

$\mathbf{m}' \leftarrow \text{restriction}(\mathbf{m})$

if $n \leq n_{min}$ **then**

$\mathbf{e} \leftarrow \text{smoothing}(\mathbf{e}, \mathbf{r}', \mathbf{m}', n/2, N_{so})$

else

$\mathbf{e} \leftarrow \text{V-cycle}(\mathbf{e}, \mathbf{r}', \mathbf{m}', n/2)$

end if

$\mathbf{p} \leftarrow \text{interpolation}(\mathbf{e})$

$\mathbf{u}_{k+1}[\mathbf{m} == 1] \leftarrow \mathbf{u}_{k+1}[\mathbf{m} == 1] + \mathbf{p}[\mathbf{m} == 1]$

$\mathbf{u}_{k+1} \leftarrow \text{smoothing}(\mathbf{u}_{k+1}, \mathbf{f}, \mathbf{m}, n, N_{sm})$

The algorithms is written using pseudocode in Algorithm 1. Both functions *restriction* and *interpolation* take a single image to process as input, while functions *V-cycle* and *residual* take approximation \mathbf{u} , right side \mathbf{f} , boundary conditions \mathbf{m} and image size n . Function *smoothing* takes the same parameters as *V-cycle* with another parameter for number of iterations to perform.

4.2 Detail recovery

Since the corrupted images contain only a small percentage of the original pixels, much of the detail in the image is lost despite the reconstruction. For the recovery of these details we have used a neural network. The neural network was trained with the reconstructed images at the input and the original images at the output. This gave us a model into which we could later feed new reconstructed images, resulting in images with recovered details. We used an autoencoder architecture depicted in Figure 3. In the figure, N stands for the number of filters, n for their size and m for down or upsampling size.

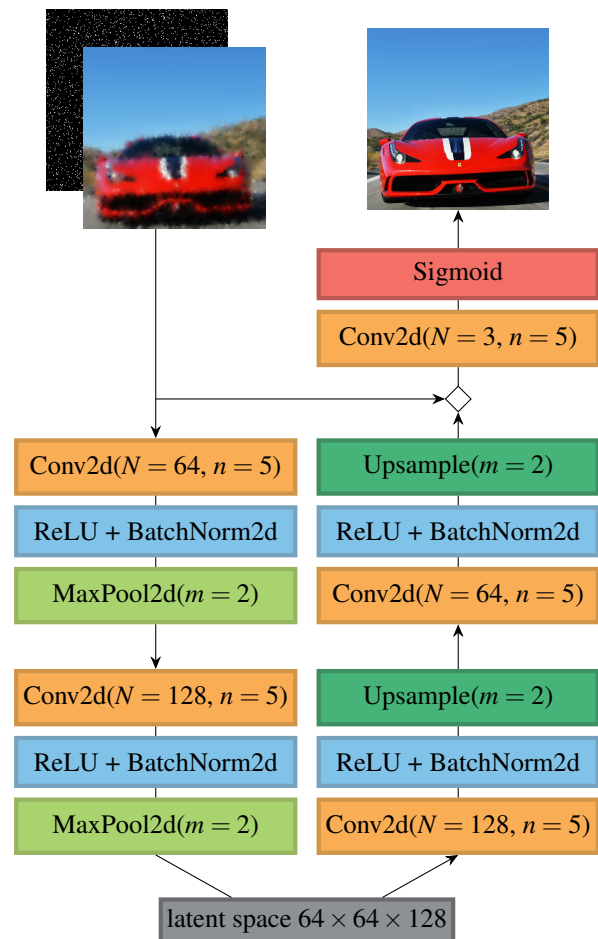


Figure 3: Architecture of the implemented neural network.



Figure 4: Outputs of model trained using (a) L1, (b) L2, (c) SSIM and (d) LPIPS loss.

For the training and evaluation we used the *CompCars Dataset* [Yan+15]. Out of all the available images, 27 thousand were used. They were cropped and scaled to size 256×256 . Then we selected 5% of pixels for each image and generated the reconstructed image as well. An example of images used can be seen in Figure 3. The input into the network is the reconstructed image with an additional fourth channel – a binary mask representing the pixels selected as boundary conditions. On the output we put the original image.

For the loss function we chose the L1 loss, because it produced the least amount of image artifacts among of the loss functions considered. The comparison of four loss functions is shown in Figure 4. The last two loss functions, SSIM [Wan+04] and LPIPS [Zha+18], are actually perceptual metrics, but they do not provide better reconstruction results.

The dataset of images was split into train (80%), validation (10%) and test (10%) sets. The sets were then further divided into batches of 64 images. The Adam optimizer was used for model parameters optimization and the training lasted for 100 epochs. The validation set was used to select the best model, and the evaluation was performed on the test set.

5 RESULTS

We present the reconstruction and detail recovery results separately. First we evaluate the process of reconstruction of basic iterative methods in comparison with the multigrid solver and then show the capability of the neural network. At the end we also present a few examples of the full pipeline – reconstruction and detail recovery.

5.1 Reconstruction

We evaluated all reconstruction methods using the baboon image and two types of boundary conditions (see Figure 5). For the metric we used the *relative residual* defined as $\|\mathbf{r}_k\|/\|\mathbf{r}_0\|$.

We only use the baboon image, because we are only interested in the convergence process. All methods are solving the same linear system, which means they all converge to the same solution. The quality of the reconstruction is only dependent on the boundary conditions.

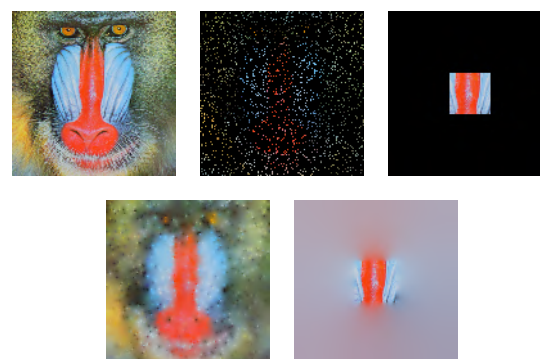


Figure 5: Images used for the evaluation of reconstruction methods: original (top left), corrupted with random boundary conditions (top middle), corrupted with center boundary conditions (top right), reconstructed with random, with center boundary conditions.

As shown in Figures 6 and 7, the multigrid method reduces the reconstruction error the fastest out of all compared methods. We evaluated the methods using two different boundary condition configurations, as shown in Figure 5, which results in vastly different performance. In both configurations, the multigrid method outperforms the other reconstruction methods.

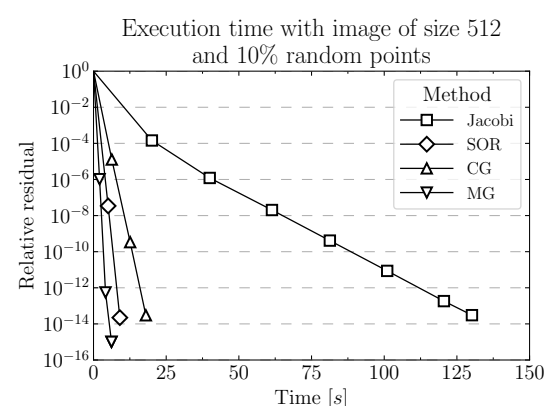


Figure 6: Comparison of reconstruction error reduction w.r.t. time, evaluated on random boundary conditions.

Another improvement given by the multigrid method is that the convergence is much less dependant on the image size when using center boundary conditions, which can be seen in Figure 8. Using other methods the recon-

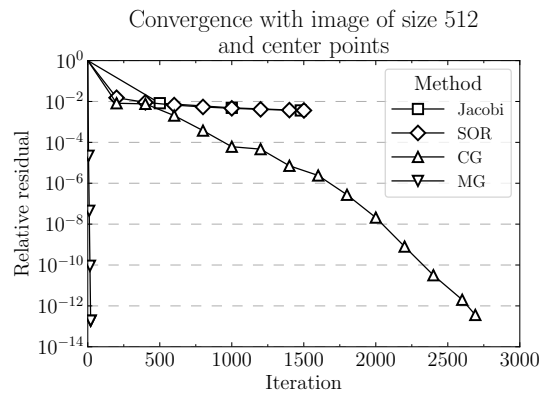


Figure 7: Comparison of reconstruction error reduction w.r.t. time, evaluated on center boundary conditions. Jacobi and SOR reconstruction were stopped early because of their slow convergence.

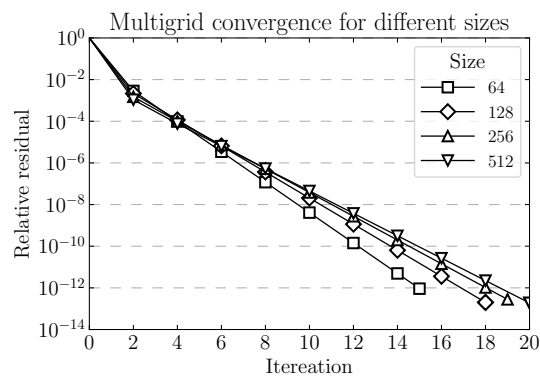


Figure 8: Comparison of multigrid reconstruction error reduction on different image sizes.

struction is propagated outward starting at the boundary conditions, but multigrid updates the whole image in one iteration.

5.2 Detail recovery

The results of the evaluation for the model trained on the reconstructed images are shown in Table 1 and its learning curve in Figure 9. Additionally, we trained the model directly on the corrupted images, but it achieved worse performance compared to training on the reconstructed images.

Metric	Reconstructed			Corrupted	
	Input	Prediction	Change	Input	Prediction
L1	0.066	0.048	27%	0.434	0.051
L2	0.013	0.009	31%	0.238	0.010
SSIM	0.575	0.679	18%	0.030	0.654
LPIPS	0.514	0.308	40%	1.082	0.339

Table 1: Values of four different metrics for the input images (reconstructed and corrupted) and predictions of both models.

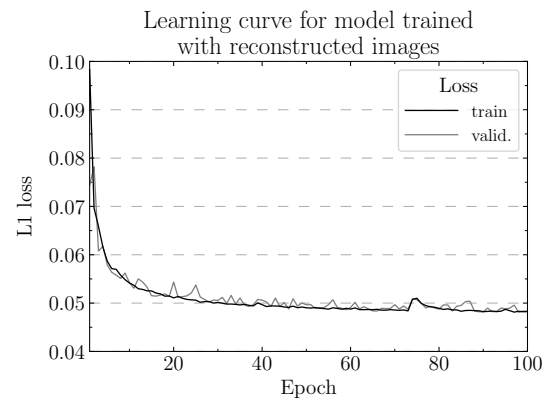


Figure 9: Model learning curve.

This can also be seen from the outputs of the models in Figure 10. Model trained on the reconstructed images produces images with less noise, better edge definition and localization. It also improves all images in the test set regarding all four used metrics.



Figure 10: Outputs of model trained on corrupted (top) and reconstructed (bottom) images.

In Figure 12 results for two more instances are shown. They represent the reconstructed images which have been most and least improved by the model. As it can be seen, both predicted images show an improvement over the reconstructed ones which were input into the model.

At the end we show seven examples of the complete reconstruction and details restoration process in Figure 11. The corrupted images are first reconstructed and then fed into the model to produce the predicted images with restored details – these can then be compared to the original images. We show the outputs of both models, trained on corrupted and reconstructed images.



Figure 11: Comparison between reconstruction and models, from top to bottom: reconstruction, model trained with corrupted images, model trained with reconstructed images, original image.

6 CONCLUSION

In this paper we dealt with the problem of reconstruction from scattered data. We focused on images and presented usage of the multigrid method to solve the differential equation used to model this problem. This method is built upon the more basic iterative methods and improves their convergence rate, which becomes much less dependent on the image size.

Since the corrupted images contain only a small fraction of the original points, the reconstructed images contain less details. Because of this we also employed a neural network model, which is capable of restoring

lost details. It is based on the autoencoder architecture and trained using a dataset of reconstructed images. The combined multigrid and neural network methods outperformed the individual methods in terms of reconstruction quality.

Using both of these methods, we can generate reconstructed images faster and improve the quality of the reconstruction itself, but we are not able to use the neural network on general images, because it was trained on only one domain. This is a classic problem of convolutional neural networks, which could be resolved by using a larger dataset containing images from multiple domains. This small dataset of cars was used only for illustration purposes. A further study using more diverse dataset like ImageNet [Den+09] would be beneficial.



Figure 12: Reconstructed and predicted images for best (top) and worst (bottom) model improvement.

7 REFERENCES

- [Den+09] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- [DF18] Matías Di Martino and Gabriele Facciolo. “An Analysis and Implementation of Multigrid Poisson Solvers With Verified Linear Complexity”. In: *Image Processing On Line* 8 (2018), pp. 192–218. URL: <https://doi.org/10.5201/ipol.2018.228> (visited on 01/15/2023).

- [GQ20] Jean Gallier and Jocelyn Quaintance. *Linear Algebra and Optimization with Applications to Machine Learning: Volume I: Linear Algebra for Computer Vision, Robotics, and Machine Learning*. English. New Jersey: WSPC, Jan. 2020. Chap. 9. ISBN: 9789811207716.
- [GV96] Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)* USA: Johns Hopkins University Press, 1996, pp. 177–180. ISBN: 978-0-8018-5414-9.
- [GT11] Thomas Guillet and Romain Teyssier. “A simple multigrid scheme for solving the Poisson equation with arbitrary domain boundaries”. In: *Journal of Computational Physics* 230.12 (June 2011). arXiv:1104.1703 [astro-ph, physics:physics], pp. 4756–4771. ISSN: 00219991. DOI: 10.1016/j.jcp.2011.02.044. URL: <http://arxiv.org/abs/1104.1703> (visited on 01/15/2023).
- [Jab+22] Susmija Jabbireddy et al. *Foveated Rendering: Motivation, Taxonomy, and Research Directions*. 2022. DOI: 10.48550/ARXIV.2205.04529. URL: <https://arxiv.org/abs/2205.04529> (visited on 01/15/2023).
- [QSS07] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. “Iterative Methods for Solving Linear Systems”. en. In: *Numerical Mathematics*. Ed. by Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. Texts in Applied Mathematics. Berlin, Heidelberg: Springer, 2007, pp. 126–132. ISBN: 978-3-540-49809-4. DOI: 10.1007/978-3-540-49809-4_4.
- [Ram+22] Aditya Ramesh et al. “Hierarchical text-conditional image generation with clip latents”. In: *arXiv preprint arXiv:2204.06125* (2022).
- [Rom+21] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. arXiv: 2112.10752 [cs.CV].
- [Sah+22] Chitwan Saharia et al. “Photorealistic text-to-image diffusion models with deep language understanding”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 36479–36494.
- [Str07a] Gilbert Strang. *Computational Science and Engineering*. English. 1st edition. Wellesley, MA: Wellesley-Cambridge Press, Nov. 2007, pp. 283–284, 568, 571–583. ISBN: 978-0-9614088-1-7.
- [Str07b] Walter A. Strauss. *Partial Differential Equations: An Introduction*. English. 2nd edition. New York: Wiley, Dec. 2007, pp. 165–172. ISBN: 978-0-470-05456-7.
- [Wan+04] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.
- [Wen17] Holger Wendland. *Numerical Linear Algebra: An Introduction*. Cambridge Texts in Applied Mathematics. Cambridge: Cambridge University Press, 2017. Chap. 3, 4, 6. ISBN: 978-1-107-14713-3. DOI: 10.1017/9781316544938.
- [Yan+15] Linjie Yang et al. “A large-scale car dataset for fine-grained categorization and verification”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3973–3981. DOI: 10.1109/CVPR.2015.7299023.
- [Zha+18] Richard Zhang et al. “The unreasonable effectiveness of deep features as a perceptual metric”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 586–595.

Blocky Volume Package: a Web-friendly Volume Storage and Compression Solution

Žiga Lesar
University of Ljubljana
Faculty of Computer and
Information Science
Večna pot 113
1000 Ljubljana, Slovenia
ziga.lesar@fri.uni-lj.si

Ciril Bohak
King Abdullah University
of Science and
Technology
Visual Computing Center
23955 Thuwal, Saudi
Arabia
ciril.bohak@kaust.edu.sa

Matija Marolt
University of Ljubljana
Faculty of Computer and
Information Science
Večna pot 113
1000 Ljubljana, Slovenia
matija.marolt@fri.uni-lj.si

ABSTRACT

The Blocky Volume Package (BVP) format is a distributed, platform-independent and API-independent format for storing static and temporal volumetric data. It is designed for efficient transfer over a network by supporting sparse volumes, multiple resolutions, random access, and streaming, as well as providing a strict framework for supporting a wide palette of encoding formats. The BVP format achieves this by dividing a volume or a volume sequence into blocks that can be compressed and reused. The metadata for the blocks are stored in separate files so that a client has all the information required for loading and decoding the blocks before the actual transmission, decoding and rendering take place. This design allows for random access and parallel loading and has been specifically designed for efficient use on the web platform by adhering to the current living standards. In the paper, we compare the BVP format with some of the most often implemented volume storage formats, and show that the BVP format supports most major features of these formats while at the same time being easily implementable and extensible.

Keywords

Volume storage, volume compression, block-based format.

1 INTRODUCTION

Volumetric data are a type of data that describe the properties or characteristics of a three-dimensional space. Such data are used to represent a wide range of physical phenomena, including but not limited to density, temperature, pressure, and composition of a particular region of space. Volumetric data play a significant role in various fields such as medical imaging, scientific visualization, and computer graphics.

Volumetric data can be represented in a number of different ways, depending on the specific application and the type of data being represented. Some common methods of representation include voxel grids, point clouds, and signed distance fields. In this paper, we will focus on storing volumetric data as voxel grids, which we will refer to as volumes. Volumes are a popular representation of volumetric data due to their sim-

plicity, flexibility, and ease of use. They can be acquired through a variety of techniques, including computer simulations, crystallography, electron microscopy (transmission tomography [KM86], cryo-electron tomography [KK09]), computed tomography [KSKV90], positron emission tomography [BMTV05], magnetic resonance imaging [Fos84], and 3D ultrasound [NE93, HZ17]. These techniques allow for the creation of highly detailed and accurate data, making them a rich source of volumes for a wide range of applications. However, volumes also have a number of disadvantages that limit their potential applicability. One major disadvantage, which we address in this paper, is the high memory usage required for storage and manipulation, especially in applications where high resolution is required. Additionally, operations on volumes, such as rendering, filtering, segmentation, and registration, can be computationally expensive and time-consuming. In certain cases, alternative methods of representation, such as surface meshes or point clouds, may be more appropriate, but conversions to such representations inherently result in information loss, which is often unacceptable.

For volume storage, compression plays a vital role in managing large amounts of data by reducing storage requirements while retaining the quality of the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

data [RGG⁺13, BRGIG⁺14]. In computer graphics, it alleviates the communication bottleneck problem, resulting in faster and more efficient visualizations, as well as reducing energy consumption which is critical on mobile devices. While storage and compression methods for two-dimensional images and videos have been subject to considerable research and innovation, which pushed the technology closer to theoretical limits, the technology for compressing and storing volumes has, in comparison, remained rather under-developed [SW03, BRLP20], despite the significantly higher storage requirements.

Existing volumetric data storage and compression formats have several shortcomings. The first issue is the lack of simple and effective compression solutions. Existing formats are either extremely simplistic and do not offer any compression capabilities, or they are featureful but difficult to understand and implement, making them less accessible to a wider range of users. Additionally, there is poor support for different platforms, as data types and endianness are often not specified, causing compatibility issues. Finally, poor extensibility is a major concern, as there is a lack of extension mechanisms, support for only a limited set of data formats, and a lack of support for GPU-accelerated formats. These problems highlight the need for better and more accessible solutions for volumetric data storage and compression.

As a potential solution to the listed shortcomings of the existing formats, this paper presents the Blocky Volume Package (BVP) format, a novel block-based format for distributed volume storage, compression and transmission. Compared with the existing formats, it includes a wide array of features and comes with a comprehensive list of advantages:

- it supports distributed volume storage, compression and transmission;
- it is agnostic to the underlying platform, storage medium, and graphics APIs;
- it is geared towards applications that require fast lookups and previews (e.g., rendering applications);
- it unifies concepts from several existing formats and generalizes them to create a more comprehensive and versatile format;
- it is simple to implement, integrate and extend.

2 RELATED WORK

There is a vast body of research on volumetric data storage and compression formats [RGG⁺13, LMG⁺18, LM14], covering a wide range of applications and use cases. Due to the scope and breadth of this field, we will focus specifically on formats for storing and compressing dense, bounded volumes. This includes formats such as DICOM [GPS05, MDG08, Clu21], and

NIfTI¹, which are commonly used in medical imaging, as well as formats such as VTI [SML06, HMCA15], HDF5 [FHK⁺11, KR18], NRRD² and Zarr³, which are used in scientific computing and other fields. While these formats may not be applicable to all types of volumetric data, they represent a significant portion of the volumetric data storage landscape and are relevant to a wide range of applications.

Volumes have historically been stored in raw format, which is a file format that stores data in its raw, unprocessed state, voxel by voxel. It is often used during data acquisition, where large amounts of data prohibit the use of on-the-fly compression. Because of this, raw format files are typically very large and can be difficult to work with. Raw format files do not include any information about the data type or endianness, which can make them incompatible with different platforms or software. Metadata for raw format files is usually stored separately (MHD⁴, MRC [CHS96, CHM⁺15], NRRD, PVM⁵, VFF⁶, VOL⁷), and includes, among other information, the resolution of the volume, its orientation and data type. However, the lack of standardization makes such metadata formats a poor choice for interoperability. Despite these limitations, raw format is often used as an intermediary format and as a format for offline use. Some of the listed formats (MHD, MRC, NRRD, VFF) include basic compression support, which is applied directly to the raw data without any transformations or spatial data structures.

In recent years, several new formats have been developed that address some of the limitations of raw format. One such format is OpenVDB [MLJ⁺13, MAB19], which is a hierarchical representation of sparse volumetric data. OpenVDB uses a voxel grid data structure and a tree-based topology to efficiently store and manipulate sparse data. This allows for efficient data access and manipulation, and makes it well-suited for visual effects and simulation applications. However, it may not be well-suited for dense volumetric data, as it is optimized for sparse data, and can be significantly more difficult to implement compared to other common formats. NeuralVDB [KLM22, Cla22] is an extension of OpenVDB that leverages the hierarchical data structure of OpenVDB and enhances it with efficient deep neural network compression capabilities. This makes it possible to store and manipulate large volumes in a way that

¹ <https://nifti.nimh.nih.gov>

² <https://teem.sourceforge.net/nrrd/>

³ <https://zarr.dev>

⁴ <https://itk.org/Wiki/ITK/MetaIO/Documentation>

⁵ <http://paulbourke.net/dataformats/pvm/>

⁶ <https://www.ventuz.com/support/help/latest/DevelopmentVFF.html>

⁷ <http://paulbourke.net/dataformats/vol/>

	RAW	BVP	MHD	NRRD	Zarr	VDB	MRC	NIfTI	DICOM	VTI	HDF5
Standard data types	no	yes	no	yes	yes	yes	no	yes	no	yes	yes
Extensible formats	no	yes	no	no	yes	yes	no	no	no	no	yes
Platform-independent	no	yes	yes	yes	yes	yes	no	no	yes	yes	yes
Simple to implement	yes	yes	yes	yes	no	no	yes	yes	no	no	no
Distributed storage	no	yes	no	yes	yes	yes	no	no	no	yes	yes
Storage alternatives	no	yes	no	no	yes	no	no	no	no	no	no
Format-agnostic operations	no	yes	no	yes	yes	no	no	no	no	no	no
Physical dimensions	no	yes	no	yes	no	no	yes	yes	yes	no	no
Extensions	no	yes	no	no	no	no	no	no	no	no	no
Multiresolution	no	yes	no	no	no	no	no	no	no	no	no
Animations	no	yes	no	yes	no	yes	no	no	yes	no	no
Supercompression	no	yes	no	yes	yes	yes	yes	yes	yes	yes	yes
GPU compression formats	no	yes	no	no	no	no	no	no	no	no	no
Higher-dimensional arrays	yes	no	no	yes	yes	no	no	no	no	no	yes
Sparse data	no	no	no	no	no	yes	no	no	no	no	yes
Transformations	no	no	no	yes	no	yes	no	yes	yes	no	no
General-purpose data	yes	no	no	yes	yes	no	no	no	no	no	yes

Table 1: Features supported in different formats.

is both memory-efficient and fast. Since NeuralVDB is built on top of OpenVDB, it shares many of its advantages as well as downsides, especially implementation complexity. NanoVDB [Mus21] is a lightweight GPU-accelerated version of OpenVDB, which primarily targets rendering applications.

VTI is the image format of the Visualization Toolkit (VTK). VTI is a general-purpose format that supports a wide variety of data types and primitives, including structured grids, unstructured grids, and polygons. As a result, it shares some of the downsides with other formats that are not optimized for volume storage. Newer versions of VTI support parallel I/O along with several compression options, including LZ4, LZMA, and ZLIB, enabling efficient reading and writing of large datasets. VTI supports only simple data formats, and GPU-accelerated compression formats are not supported. The data in a VTI file is described using an XML document, which can be complex to parse. VTI has the advantage of being widely supported and used in the scientific and research communities, and there are many libraries and tools available for working with VTI data. However, due to the broad feature set, implementing VTI can be a challenging task.

Another format that is gaining popularity is HDF5, which stands for Hierarchical Data Format version 5. HDF5 is a general-purpose data format designed to store large, complex data sets in a hierarchical format. It provides a rich feature set, including support for complex data structures, metadata, and parallel I/O. HDF5 is widely used and supported in many scientific and research communities, and has a wide range of libraries and tools available for working with HDF5 data. However, it is more complex and can be significantly more difficult to implement than other formats, and may not be as well-suited for storing volumes because the format is not aware of their specific spatial structure.

Zarr is a similar format as HDF5, it is also a general-purpose data format designed to store large, multi-dimensional arrays in a hierarchical structure. Zarr provides a simple, easy-to-use Python API, and can be used with a wide range of compression and encoding techniques. It has been designed to support distributed storage solutions, such as Amazon S3. However, it supports only a limited set of data formats, which does not include GPU-accelerated formats.

DICOM is a standard for storage, communication and management of medical images and related information, such as MRI and CT scans. DICOM was created to facilitate the exchange of information between different medical imaging devices and provide a standard format for storing and transmitting images and metadata. It is primarily a container format, and storage and processing information for each specific use case is provided in a separate specification document (e.g. there is a separate specification for 3D ultrasound images). DICOM is a large and complex standard that encompasses a wide range of use cases, with volume storage being only a small part of it. Unsurprisingly, implementing DICOM requires a significant investment of time and resources.

NIfTI is a file format for storing medical images and data, particularly in the field of neuroimaging. NIfTI was created as an alternative to the popular Analyze format and was designed to overcome some of the limitations of Analyze. NIfTI provides basic compression support through the use of DEFLATE, and it includes spatial transformation information, such as the orientation and size of the data. This information is stored in the header of the NIfTI file, along with information about the data type, data range, and measurement units. The use of physical units ensures that the data is correctly scaled and can be used for quantitative analysis. However, despite its advantages, NIfTI also has some limitations. The set of data formats is limited

and does not include GPU-accelerated compression formats. NIFTI is also not a hierarchical format, which prohibits efficient random access to the underlying data.

The BVP format addresses the downsides of the existing formats by unifying their advantages while following a simplistic design. One of the main advantages of BVP is its simplicity, which greatly facilitates integration and extension (e.g., as a plugin) of existing software. Unlike other formats, BVP additionally supports GPU-accelerated compression formats, which motivates its usage in rendering applications. For an overview of the features of different formats, refer to Table 1.

3 BLOCKY VOLUME PACKAGE

The main motivation behind BVP is its use on the web platform, where platform independence, memory safety, and efficient compression are crucial factors. The web presents additional restrictions, such as limited memory usage and restricted external file access. However, it simplifies some tasks, such as parsing JSON documents and manipulating arrays.

Drawing from the best features of the existing formats, we designed the BVP format with the following goals in mind:

- (G₁) the format must enable efficient parallel random access;
- (G₂) the format must support distributed storage and access;
- (G₃) metadata must be stored separately to enable clients to have all information about a volume available before the actual transmission, decoding and rendering take place;
- (G₄) data types have to support common use cases in a wide range of applications;
- (G₅) common GPU-accelerated compression formats, such as S3DC [YNV08], ETC [SAM05] and ASTC [NLP⁺12], must be supported;
- (G₆) the format must support compression in the form of reuse of parts of a volume or general-purpose compression;
- (G₇) execution of common operations, such as cropping and concatenation, must be possible without knowledge of the data format;
- (G₈) the format must be able to store multiple modalities, e.g. raw data and segmentation;
- (G₉) multiple resolutions of the same volume may optionally be added to allow fast previews;
- (G₁₀) physical units must be used to scale the volume correctly and enable quantitative analysis;
- (G₁₁) established data representation standards, such as IEEE 754, UTF-8 and JSON, must be respected throughout the format for it to be completely platform-independent; and
- (G₁₂) an extension mechanism must be available to allow future enhancements.

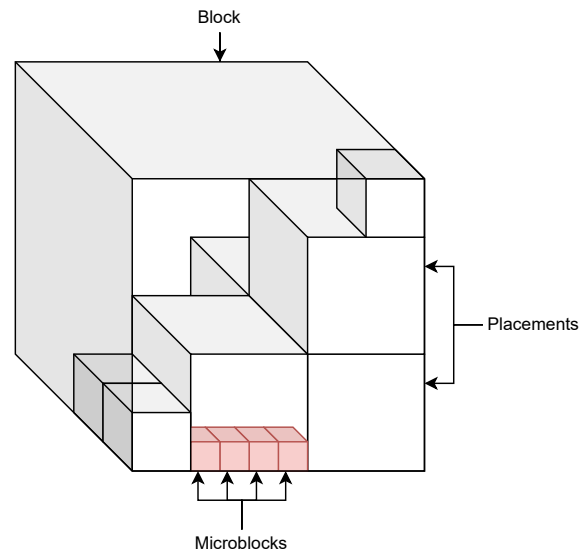


Figure 1: Blocks, microblocks and placements. A block is represented as a 3D array of microblocks. The microblocks come from external sources or other blocks copied into placements.

3.1 Overview

A BVP asset is built around 4 fundamental concepts: formats, microblocks, blocks, and modalities. A *microblock* is a fixed and indivisible block of voxels, represented by a specific number of bytes. The dimensions of a microblock, the number of bytes representing it, and the interpretation of these bytes are described by a *format*. Microblocks are assembled together to form a *block* as shown in Fig. 1. The data for a block may come from different sources, including external files and other blocks (G₂). Finally, blocks may be referenced by *modalities*, which equip a block with metadata and physical characteristics, such as its physical size. A BVP asset may contain multiple modalities such as raw data and different segmentations. Transformations are not described by a BVP asset, as it is only a storage format, not a scene description format.

All information about modalities, formats, blocks, how the blocks are structured in a hierarchy, and how to access the external microblock data, is supplied in a separate (G₃) JSON (G₁₁) document called the *manifest*.

3.2 Microblocks and formats

The concept of microblocks has been introduced to allow different block-level encoding schemes, such as S3DC, ETC, ASTC, etc., which are common in computer graphics and often hardware-accelerated (G₅). For example, a microblock may be as simple as a single voxel with a 32-bit floating-point value or more complex such as a $4 \times 4 \times 4$ RGBA volume represented by 16 bytes in ASTC format. There exist many encod-

ing schemes and choosing any fixed set of them for inclusion in BVP would make it inflexible and inextensible, as it would not be able to accommodate the diversity and evolution of encoding schemes. Instead, we chose to generalize the concept of a block-level encoding scheme as a format family, and a specific instance of that scheme as a format.

A format defines the dimensions of a microblock, the number of bytes representing it, and the semantics of these bytes when extracting voxel values. The dimensions and the number of bytes must be specified in a BVP asset so that even if an implementation encounters an unknown format (possibly added to BVP as an extension), it can still perform basic operations, such as block assembly, cropping and concatenation without having to decode a single microblock (G_7).

Formats that share similar characteristics are organized into format families, which can be added to BVP through extensions (G_{12}). The most basic format family is the “mono” format family, which describes single-voxel microblocks storing vectors of a single primitive type, for example one 8-bit unsigned integer or four 32-bit floating-point numbers. A format belonging to the “mono” format family must specify the primitive type and its size in bytes, along with the number of vector components. The primitive types are unsigned integers, signed integers represented as two’s complements, or floating-point numbers conforming to the IEEE 754 standard, all of which are stored in little-endian format when endianness is relevant (G_{11}). Microblock byte size is the product of the number of components and the byte size of the primitive type. A conformant implementation must support at least vectors with 1 to 4 components, signed and unsigned integers represented with 1, 2, or 4 bytes, and single- and double-precision floating point numbers, which is adequate for most practical applications (G_4). Note that not all of these formats have equivalents in common graphics APIs, although they are commonly used in practice (e.g. doubles in scientific simulations and 16-bit integers in medical imaging), rationale being that BVP should not be limited by such APIs. In such cases, the data must be converted to an appropriate representation. It is important to note that data conversions are specific to pairs of formats and are therefore not included in the BVP specification.

3.3 Blocks

Blocks are cuboid regions of space formed by microblocks (see Fig. 1) of a specific format, which can be hierarchically assembled together to form volumes. Microblocks in a block adhere to a right-handed coordinate system and are ordered lexicographically in \mathbb{R}^3 , which aligns with most modern graphics APIs. The data for a block can come from various sources

such as files, web servers, or other blocks positioned in designated areas called placements within the parent block. This structure enables block reuse, resulting in a compressed volume (G_6), and additionally allows distributed storage (G_2).

An assembly process is required to fill a block with data. It is initialized with zeros, then placements are filled with referenced blocks, and finally, external data is copied into the block if available. Either a block has placements, or it is defined by external data. With future extensions in mind, we have decided not to explicitly disallow having both placements and external data present in a single block. This, however, necessitates an order of operations. Since the choice is largely arbitrary, we opted for placements first and external data second. The designated format of the parent block must be matched by all referenced sources.

External data can also be differential, in which case the existing microblocks are modified rather than overwritten, which can significantly improve compressibility. The placements must not be overlapping to allow parallel assembly (G_1). Additionally, circular dependencies between blocks are not allowed and an implementation should be robust to malformed assets. As a result, the blocks form a directed acyclic graph (DAG). Such a structure enables fast queries if an application chooses to store the blocks in unassembled form, or in out-of-core scenarios (G_1). The use of DAGs is not new in computer graphics [KSA13, DKB⁺16, vdLSE20], but to our knowledge, BVP is the first format to use DAGs for dense volume compression.

A block may reference a lower-resolution block representing the same data, which enables representing the same data in multiple resolutions, allowing fast previews and accelerated rendering (G_9). Additionally, rudimentary support for animated volumes is provided by allowing blocks to reference subsequent blocks as frames of an animation. Video compression is possible by sharing constituent blocks between frames. It is important to note that while this feature is available, the BVP format is not primarily designed to target volumetric video.

3.4 Modalities

Blocks are referenced by modalities which provide semantic information about the blocks, such as the physical size and acquisition method (G_{10}). A BVP asset can store multiple modalities, such as raw data and segmentation volumes, or even scans of the same subject using different methods like magnetic resonance imaging and computed tomography (G_8). This allows for a comprehensive representation of the data and enables the storage of multiple views and interpretations of the same underlying data. Unlike blocks in a single hierarchy, different modalities can use different formats.

3.5 Manifest

The manifest is a UTF-8-encoded JSON document (G_{11}), which may be split into several parts, that describes all the information stored in a BVP asset (G_3). It includes information about modalities, formats, blocks, the structure of blocks in a hierarchy, and how to access the external microblock data. This design draws inspiration from the glTF format. Furthermore, the manifest stores metadata about the BVP asset, such as copyright information, acquisition methods, checksums, and various timestamps.

External microblock data is referenced from the manifest by a URL (G_{11}), which may locate a file in a local file system, an archive, or a web server. Consequently, the BVP format is agnostic of the storage medium, at the same time allowing distributed storage and single-file volume archives (G_2). External data can optionally be supercompressed to further improve the compression ratio. A modified version of the LZ4 compression scheme, known for its simplicity and high decompression speeds, is a core component of the BVP format. Deflate, in contrast to LZ4, is much more commonly used and achieves better compression ratios at the cost of simplicity and decompression speed. For these reasons it is included in the BVP format as an extension.

3.6 Extension mechanism

The BVP format has a built-in extension mechanism, similar to the one used by the glTF format (G_{12}). This mechanism allows the core format to remain simple to implement, while still providing the ability to extend certain functionalities, for example by adding new format families, compression methods, and block placement capabilities. Extensions are categorized into two types: those that allow a BVP asset to be decoded even if the extension is not supported and those that require specific decoding logic, such as compression methods. Both types of extensions must be listed in the manifest so that they are readily available to the BVP reader.

The core BVP format, without any extensions, only describes the block assembly process, the “mono” format family, and modalities. A simplified LZ4 supercompression is also included. Everything else, such as compressed formats, multiple resolutions, animations, and differential blocks, is provided by extensions.

4 RESULTS

We compared the following existing formats in terms of file size: RAW (baseline), BVP, MRC, NRRD, VTI, NIFTI, Zarr, HDF5, OpenVDB, and DICOM. Since Zarr does not prescribe a storage medium, we used an uncompressed ZIP archive. Similarly, we used a simplified form of ASAR⁸ for BVP assets. In the first

set of tests we disabled supercompression and relied only on block reuse to reduce the file sizes. In the next set of tests, we enabled LZ4 supercompression where it was available. The dataset was comprised of 40 single-channel 8-bit integer volumes from various fields⁹, adding up to approximately 1424 MB. As a practical use case, we also used BVP to store an ASTC-compressed RGBA volume to show that BVP can effectively handle GPU-accelerated compression formats. Finally, to show the potential for compression with DAGs, we evaluated BVP on a 32-bit instance segmentation volume. The dataset and the scripts used for storing it in different formats are available on github¹⁰.

For the BVP assets we used a two-level block hierarchy with the parent block in the first level and subblocks of size $32 \times 32 \times 32$ in the second level. Equal blocks were found using the xxHash32 hash function¹¹.

Figure 2 reveals that BVP with block reuse outperforms other formats, which do not include this feature. This is most evident in volumes with lots of empty space. When we enabled LZ4 supercompression, we found that only the BVP format could produce a smaller file than LZ4 alone, as shown in Figure 3. Note that in Figures 2 and 3 we colored the baseline (the raw format without and with supercompression) yellow, the BVP format red, and other formats blue. Light blue was used in Figure 3 for the formats that are not directly comparable to BVP due to e.g. unsupported data types or unsupported compression algorithms. Some formats outperformed BVP, but they relied on other compression algorithms, such as DEFLATE (MRC, NRRD, NIFTI) or JPEG2000 (DICOM), as they do not support LZ4. OpenVDB does not support 8-bit integers and stores values as 32-bit floating-point numbers, which performs well without supercompression, but poorly with it. The only formats with comparable features were VTI, Zarr, and HDF5, which produced 4-8 % larger files and are significantly more complex in design and implementation. Additionally, HDF5 provides LZ4 only as a plugin and it does not allow arbitrary subblock sizes.

Compression ratios expectedly varied throughout the dataset depending on the amount of noise in the volumes. Where the amount of noise was reasonable (in around half of the volumes), we achieved reductions in file sizes from 10 % to 70 % without the use of LZ4 supercompression. With LZ4 supercompression, reductions varied more. However, when comparing LZ4-compressed raw volumes and LZ4-compressed BVP

⁸ <https://github.com/electron/asar>

⁹ <https://klacansky.com/open-scivis-datasets/>

¹⁰ <https://github.com/UL-FRI-LGM/wscg2023-bvp>

¹¹ <http://www.xxhash.com/>

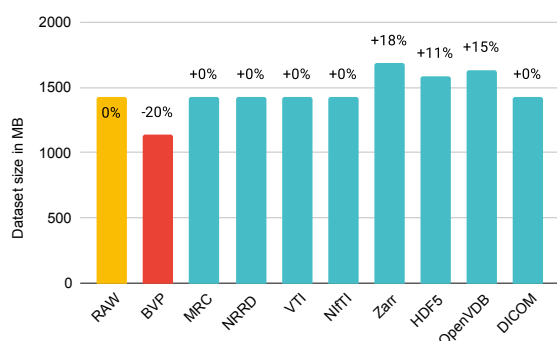


Figure 2: Dataset sizes without supercompression. The RAW baseline and BVP were colored yellow and red, respectively, for emphasis.

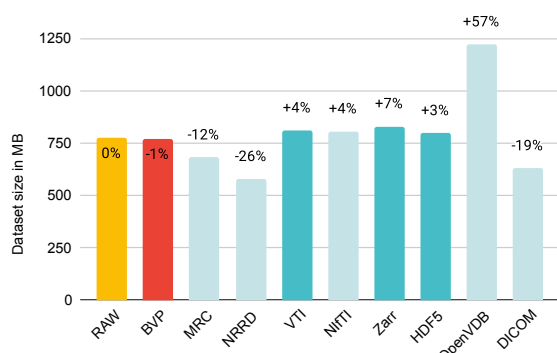


Figure 3: Dataset sizes with supercompression. The RAW baseline is a LZ4-compressed RAW dataset. Light blue formats are not directly comparable to BVP.

volumes, the resulting file sizes were very similar, with differences of up to 5 %.

Smaller blocks could be used to improve the compression ratio at the expense of a bigger manifest and a lower compression speed. However, due to diminishing returns, we found that blocks of size $32 \times 32 \times 32$ were a good compromise.

Note that these results are for lossless compression only. Extending the condition for block reuse from equality to similarity would result in lossy compression, which would significantly increase the compressibility of the volumes due to the use of DAGs. Even though this feature has not yet been implemented in our compressor, the BVP format readily supports it. Similarly, differential blocks were not included in the evaluation. We conjecture that the benefit would be marginal in the static case and more pronounced in animations.

In addition to raw volume compression, we evaluated BVP on an ASTC-compressed RGBA volume of size $768 \times 768 \times 360$. We used microblocks of dimensions $4 \times 4 \times 4$ and $6 \times 6 \times 6$ and a similar two-level block hierarchy with subblocks of size $48 \times 48 \times 48$. Both cases were evaluated with and without LZ4 supercompression. Figure 4 shows that BVP with LZ4 achieves

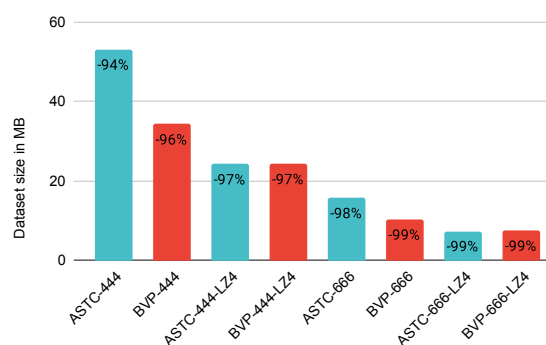


Figure 4: Dataset size using ASTC compression. Percentages are calculated with respect to the RAW file size (approximately 850 MB).

approximately equal final asset size as raw ASTC with LZ4 supercompression, but with added metadata and hierarchical structure.

To show the compression capabilities of DAGs, we used BVP to compress a 32-bit instance segmentation volume of size $512 \times 512 \times 512$. We used the mono format and a two-level block hierarchy with subblocks of size $32 \times 32 \times 32$. The raw volume of 512 MB was reduced to 259 MB, which corresponds to a compression ratio of 50 %. After enabling LZ4 compression, the raw volume was reduced to 6.1 MB, while the BVP-compressed volume was reduced to 6.9 MB. The overhead was almost exclusively due to the manifest.

5 CONCLUSION

The BVP format offers many advantages for storing and distributing volumes. It is a simple-to-implement platform-independent format that can be used to store large volumes from a variety of different fields. BVP draws inspiration from many existing formats, consolidating ideas such as multiresolution representation, hierarchical storage, parallel I/O, and supercompression. It readily supports GPU-accelerated compressed formats for efficient use in computer graphics. Contrary to the existing formats, the BVP format respects the widely-accepted engineering standards, facilitating data exchange and interoperability.

However, there are some limitations to BVP. BVP by design only supports dense, bounded 3D volumes. Specifically, it does not support unlimited indexing (such as in OpenVDB), it does not support general-purpose data storage (such as in HDF5), and it does not support higher-dimensional arrays (such as in NRRD and Zarr). Additionally, a BVP asset does not include information about the scene, such as transformations, transfer functions, and illumination, so a separate format, such as USD¹² [CMMLB22] or

¹²<https://graphics.pixar.com/usd/>

glTF [RAPC14], must be used for that purpose. Furthermore, BVP deliberately excludes information about sampling and data conversions, as these operations are format-specific and often domain-specific.

There are also some technical limitations that have been deliberately introduced to simplify the implementation process. BVP has some restrictions on block and placement dimensions and formats. For instance, blocks and placements must be completely contained within the parent block, and the size of a block and placement must be an integer multiple of microblock size. Consequently, block-level encoding schemes prohibit arbitrary block sizes. Additionally, the format of placements must match the format of the block, and format mixing is only allowed on modality level.

Despite the above limitations, the BVP format provides a simple and comprehensive representation of volumes from various fields. It is a reasonable and featureful alternative to the existing formats.

6 REFERENCES

- [BMTV05] Dale L Bailey, Michael N Maisey, David W Townsend, and Peter E Valk. *Positron emission tomography*, volume 2. Springer, 2005.
- [BRGIG⁺14] M. Balsa Rodríguez, E. Gobbetti, J.A. Iglesias Guitián, M. Makhinya, F. Marton, R. Pajarola, and S.K. Suter. State-of-the-art in compressed gpu-based direct volume rendering. *Computer Graphics Forum*, 33(6):77–100, 2014.
- [BRLP20] Rafael Ballester-Ripoll, Peter Lindstrom, and Renato Pajarola. Tthresh: Tensor compression for multidimensional visual data. *IEEE Transactions on Visualization and Computer Graphics*, 26(9):2891–2903, 2020.
- [CHM⁺15] Anchi Cheng, Richard Henderson, David Mastronarde, Steven J. Ludtke, Remco H.M. Schoenmakers, Judith Short, Roberto Marabini, Sargis Dalakyan, David Agard, and Martyn Winn. Mrc2014: Extensions to the mrc format header for electron cryo-microscopy and tomography. *Journal of Structural Biology*, 192(2):146–150, 2015.
- [CHS96] R.A. Crowther, R. Henderson, and J.M. Smith. Mrc image processing programs. *Journal of Structural Biology*, 116(1):9–16, 1996.
- [Cla22] Ronald Clark. Volumetric bundle adjustment for online photorealistic scene capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6124–6132, June 2022.
- [Clu21] David A. Clunie. Dicom format and protocol standardization—a core requirement for digital pathology success. *Toxicologic Pathology*, 49(4):738–749, 2021.
- [CMMLB22] Jon-Patrick Collins, Romain Maurer, Fabrice Macagno, and Christian Lopez Barron. Usd at scale. In *The Digital Production Symposium*, DigiPro ’22. Association for Computing Machinery, 2022.
- [DKB⁺16] Bas Dado, Timothy R. Kol, Pablo Bauszat, Jean-Marc Thiery, and Elmar Eisemann. Geometry and attribute compression for voxel scenes. *Computer Graphics Forum*, 35:397–407, 5 2016.
- [FHK⁺11] Mike Folk, Gerd Heber, Quincey Koziol, Elena Pourmal, and Dana Robinson. An overview of the hdf5 technology suite and its applications. page 36–47, New York, NY, USA, 2011. Association for Computing Machinery.
- [Fos84] Margaret A Foster. Magnetic resonance in medicine and biology. *Radiology and Nuclear Medicine*, 1984.
- [GPS05] R.N.J. Graham, R.W. Perriss, and A.F. Scarsbrook. Dicom demystified: A review of digital file formats and their use in radiological practice. *Clinical Radiology*, 60(11):1133–1140, 2005.
- [HMCA15] Marcus D. Hanwell, Kenneth M. Martin, Aashish Chaudhary, and Lisa S. Avila. The visualization toolkit (vtk): Rewriting the rendering code for modern graphics cards. *SoftwareX*, 1-2:9–12, 2015.
- [HZ17] Qinghua Huang and Zhaozheng Zeng. A review on real-time 3d ultrasound imaging technology. *BioMed research international*, 2017, 2017.
- [KK09] Roman I Koning and Abraham J Koster. Cryo-electron tomography in biology and medicine. *Annals of Anatomy-Anatomischer Anzeiger*, 191(5):427–445, 2009.
- [KLM22] Doyub Kim, Minjae Lee, and Ken Museth. Neuralvdb: High-resolution sparse volume representation using hierarchical neural networks, 2022.
- [KM86] Satoshi Kawata and Shigeo Minami. The principle and applications of optical microscope tomography. *Acta his-*

- tochemica et cytochemica*, 19(1):73–81, 1986.
- [KR18] Quincey Koziol and Dana Robinson. Hdf5. [Computer Software] <https://doi.org/10.11578/dc.20180330.1>, mar 2018.
- [KSA13] Viktor Kämpe, Erik Sintorn, and Ulf Assarsson. High resolution sparse voxel dags. *ACM Transactions on Graphics*, 32:1–13, 7 2013.
- [KSKV90] Willi A Kalender, Wolfgang Seissler, Ernst Klotz, and Peter Vock. Spiral volumetric CT with single-breath-hold technique, continuous transport, and continuous scanner rotation. *Radiology*, 176(1):181–183, 1990.
- [LM14] Michele Larobina and Loredana Murino. Medical image file formats. *Journal of Digital Imaging*, 27(2):200–206, 2014.
- [LMG⁺18] S. Li, N. Marsaglia, C. Garth, J. Woodring, J. Clyne, and H. Childs. Data reduction techniques for simulation, visualization and data analysis. *Computer Graphics Forum*, 37(6):422–447, 2018.
- [MAB19] Ken Museth, Nick Avramoussis, and Dan Bailey. Openvdb. In *ACM SIGGRAPH 2019 Courses*, SIGGRAPH ’19, New York, NY, USA, 2019. Association for Computing Machinery.
- [MDG08] Mario Muštra, Kresimir Delac, and Mislav Grgic. Overview of the dicom standard. In *2008 50th International Symposium ELMAR*, volume 1, pages 39–44, 2008.
- [MLJ⁺13] Ken Museth, Jeff Lait, John Johanson, Jeff Budsberg, Ron Henderson, Mihai Alden, Peter Cucka, David Hill, and Andrew Pearce. Openvdb: An open-source data structure and toolkit for high-resolution volumes. In *ACM SIGGRAPH 2013 Courses*, SIGGRAPH ’13, New York, NY, USA, 2013. Association for Computing Machinery.
- [Mus21] Ken Museth. Nanovdb: A gpu-friendly and portable vdb data structure for real-time rendering and simulation. In *ACM SIGGRAPH 2021 Talks*, SIGGRAPH ’21, New York, NY, USA, 2021. Association for Computing Machinery.
- [NE93] T.R. Nelson and T.T. Elvins. Visualization of 3d ultrasound data. *IEEE Computer Graphics and Applications*, 13(6):50–57, 1993.
- [NLP⁺12] Jorn Nystad, Anders Lassen, Andy Pomianowski, Sean Ellis, and Tom Olson. Adaptive scalable texture compression. In *Proceedings of the Fourth ACM SIGGRAPH / Eurographics Conference on High-Performance Graphics*, EGGH-HPG’12, page 105–114, Goslar, DEU, 2012. Eurographics Association.
- [RAPC14] Fabrice Robinet, Rémi Arnaud, Tony Parisi, and Patrick Cozzi. gltf: Designing an open-standard runtime asset format. 5:375–392, 2014.
- [RGG⁺13] Marcos Balsa Rodríguez, Enrico Gobbetti, José A. Iglesias Guitián, Maxim Makhinya, Fabio Marton, Renato Pajarola, and Susanne K. Suter. A Survey of Compressed GPU-Based Direct Volume Rendering. In M. Sbert and L. Szirmay-Kalos, editors, *Eurographics 2013 - State of the Art Reports*. The Eurographics Association, 2013.
- [SAM05] Jacob Ström and Tomas Akenine-Möller. Ipackman: High-quality, low-complexity texture compression for mobile phones. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, HWWS ’05, page 63–70, New York, NY, USA, 2005. Association for Computing Machinery.
- [SML06] Will Schroeder, Ken Martin, and Bill Lorensen. *The visualization toolkit an object-oriented approach to 3D graphics, 4th Edition*. Kitware, 2006.
- [SW03] J. Schneider and R. Westermann. Compression domain volume rendering. In *IEEE Visualization, 2003. VIS 2003.*, pages 293–300, 2003.
- [vdLSE20] Remi van der Laan, Leonardo Scandolo, and Elmar Eisemann. Lossy geometry compression for high resolution voxel scenes. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3:1–13, 4 2020.
- [YNV08] Héctor Yela, Isabel Navazo, and Pere-Pau Vazquez. S3Dc: A 3Dc-based Volume Compression Algorithm. In Luis Matey and Juan Carlos Torres, editors, *CEIG 08 - Congreso Espanol de Informatica Grafica*. The Eurographics Association, 2008.

Generating Realistic River Patterns with Space Colonization

Haoran Feng

University of Auckland
School of Computer
Science

Auckland, New Zealand
hfen962@aucklanduni.ac.nz

Burkhard C. Wünsche

University of Auckland
School of Computer
Science

Auckland, New Zealand
burkhard@cs.auckland.ac.nz

Alex Shaw

University of Auckland
School of Computer
Science

Auckland, New Zealand
l.shaw@auckland.ac.nz

ABSTRACT

River generation is an integral part of realistic terrain generation, since rivers shape terrains and changes in terrain, e.g., due to tectonic movements can change the path of rivers. Fast existing terrain generation methods often result in non-realistic river patterns, whereas physically-realistic techniques, e.g., building on erosion models, are usually slow. In this paper we investigate whether the space colonization algorithm can be modified to generate realistic river patterns. We present several extensions of the space colonization algorithm and show with a user study with $n = 55$ participants that some variants of the algorithm are capable of generating river patterns that are indistinguishable from real river patterns. Although our technique can not generate all types of natural river patterns, our results suggest that it can prove useful for developing plausible 2D maps and potentially can form the basis for new terrain generation techniques.

Keywords

Space Colonization, river pattern, procedural generation, realism

1 INTRODUCTION

Rivers are essential for realistic terrains since rivers effect the shape of terrains via erosion, and vice versa terrain changes can influence the flow of rivers. Most countries contain rivers within their boundaries [Way19], and for some countries, such as New Zealand, rivers cover a large part of the land surface [NIWA18].

There are many existing realistic terrain generation algorithms [Arc11, CBCB+16, DP10, BF02]. However, they generally focus on the generation of landforms, instead of the rivers within the landforms. They are suitable for the generation of terrain in mountainous areas, but bodies of water are missing from the generation, which lowers the degree of realism in the terrain. Although the area of river generation is underexplored, there also exist some specialized river generation algorithms [GGGP+13, PDGC+19]. However, all algorithms listed above generate rivers or terrain by simulating erosion using physical simulation engines, hence the generation process is computationally intensive, and large-scale terrain generation may be impractical for commercial uses. An alternative method of terrain generation that produces more realistic river patterns is one that first generates realistic river patterns, then generates the terrain surrounding the patterns accordingly.

An efficient method of pattern generation is the space colonization algorithm [RLP07], which has been used originally for the generation of tree structures. In this paper, we explore how to extend the algorithm for river

pattern generation. We aim to answer the following research question:

Can the space colonization algorithm be modified to efficiently generate realistic river patterns?

2 BACKGROUND

2.1 Types of River Patterns

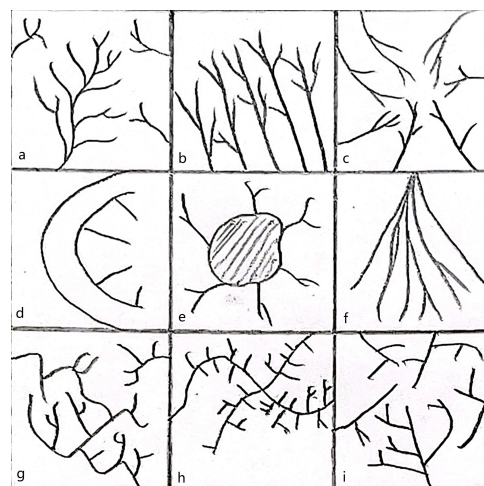


Figure 2.1: Different types of river patterns, adapted from Twidale's 2004 paper [Twi04].

Realistic river generation is not a trivial task, as there are many factors influencing the formation of natural rivers, such as the slope of the terrain, the material of the terrain, and the rate of erosion. This can result in

many types of river patterns [Twi04], as shown in *Figure 2.1*.

2.2 Space Colonization

Space colonization is a procedural content generation algorithm for generating realistic tree structures. It was created and published by Runions et al. in 2007. In their research paper, an overview of the algorithm is given, and the procedure of the algorithm is shown in detail. The paper also outlines the effects on the resultant tree structure if some parameters are changed. From the examples given in the paper, it is shown that the tree branches can be made denser or sparser, and also thicker or thinner, depending on the environment and the usage of the trees.

The procedure of the algorithm is outlined below, where black nodes are the nodes that are already in the tree structure, and blue nodes are the “attraction points” that control the generation and structure of the tree.

1. Start with a predetermined set of black nodes (i.e. start nodes) and a random set of blue nodes spread across the generation space.
2. Find the closest black node for each blue node under a distance and assign each of the blue nodes to the closest black node.
3. Determine for each black node the normalized directional vectors to its assigned blue nodes. If a black node is not assigned any blue nodes, then it skips step 4.
4. Sum all normalized directional vectors for each black node then normalize the resultant vector. Generate a new black node towards that direction a unit length away.
5. Determine if any blue nodes are within the vicinity of a black node, and remove the blue node if it is within a distance from a black node.
6. Return to step 2 until no new black nodes can be added.

3 RELATED WORK

Rivers are usually generated as by product of terrain generation, i.e., a terrain is generated first and rivers are formed using flow simulations. Only a few authors generate explicitly, i.e., that the rivers are formed first and influence the terrain shape.

Génevaux et al. devised a terrain generation model based on hydrology [GGGP+13]. The authors take a boundary for the terrain and initial nodes on the said boundary as input, expand the river structure from the nodes, modify parts of the structure based on the Rosgen river classifications [Ros94], and generate the 3D model according to the resultant structure. The model generates realistic terrain and naturalistic river patterns

with reasonable efficiency. However, the node expansion and structure generation processes are convoluted and difficult to understand and implement, so there is room for simplifications or the usage of more straightforward algorithms for potential accelerations in terrain generation.

A more recent model developed by Peytavie et al. [PDGC+19] follows a similar pipeline as the model of Génevaux et al., as it also constructs river networks following the Rosgen river classifications. Then the model derives the multiple aspects of the terrain from the constructed patterns (e.g. drainage, slope), and it follows an amplification-combination procedure to simulate water flow along the rivers. The model results in an efficient method of generating terrain with realistic fluid flow animations, but the paper does not focus on the patterns of the generated rivers, so the realism of the generated rivers, hence also the generated terrain, may be further improved.

Van den Hurk et al. present a sketch-based terrain modelling techniques where rivers are manually inserted into the terrain using sketching [vHYW11].

4 DESIGN

As discussed in Subsection 2.2, the space colonization algorithm succeeds in generating realistic tree structures. However, due to the difference between tree structures and river structures, the algorithm does not generate satisfactorily realistic river patterns. For the output patterns to more closely resemble the structure of natural rivers [Twi04], several modifications to the algorithm can be made, and they are outlined and discussed in this subsection.

4.1 River Pattern Requirements



Figure 4.1: Real river patterns: Godavari River (left), Mississippi River (middle), and Yangtze River (right).¹

Figure 4.1 shows three examples of river maps from famous rivers across the world. Comparing the maps with the patterns generated from space colonization, the following dissimilarities may be observed (they are referred to as Dissimilarity 1, 2, and 3 respectively).

1. Curvature: The real river patterns tend to be curvier than the generated patterns.

¹ All patterns are listed in *Table 6.8*

2. Parallelism: The river branches in real river patterns are generally not parallel as opposed to the generated patterns.
3. Lateral branches: Short and perpendicular branches, known as lateral branches, from generated patterns are uncommon to occur in natural rivers.

Accounting for these differences between real river patterns and generated patterns, we devise a range of modifications to reduce the dissimilarities, such that the generated patterns also contain similar features as real river patterns, hence are considered more realistic.

4.2 Space Colonization Momentum

To reduce Dissimilarity 1 and introduce more curvature into the structures, the simplest approach is to divert the course of node generation. Our proposed Momentum variant of the space colonization algorithm slightly alters the direction of node generation using a measure of momentum, inspired by the concept of momentum in the field of physics.

In physics, if an object contains momentum in a direction, then it tends to continue moving toward that direction until an external force is acted on the object. This concept is adapted to generate nodes, however in a slightly different manner. If momentum is applied directly, where the direction of the next node tends to stay in the same direction as the previous node, then the resultant patterns become even less curvy than those generated from the original space colonization. Since we want to introduce more curvature into the patterns, we make a modification that acts the opposite as described above. The procedure of the Momentum variant is shown in *Figure 4.2 (left)*, and this modification takes place at step 4, as described in Subsection 2.2.

In *Figure 4.2 (left)*, (M1) shows a structure after step 4 of the space colonization procedure outlined above, where A, B, and C are nodes in the structure, v_1 is the direction vector of the generation of B from A, and v_2 is the direction vector of the generation of C from B. (M2) shows the introduction of a momentum vector, v_1 , shown in red. The momentum vector belongs to B, as it is the direction of its generation, and hence it is applied to the next generation to alter its direction. For the new node stemming from B, if original space colonization is used, then C is generated, however, for the Momentum variant, the direction vector for the new node is subtracted by the momentum vector, multiplied by a multiplier k , as shown in (M3). The multiplier k can be interpreted as the strength coefficient of momentum, or essentially the size of effect the momentum vector (i.e. the previous direction vector) has on the new direction vector. In (M4), the subtraction results in the new direction vector $u = v_2 - kv_1$, which is then normalized to \hat{u} in (M5), where the tip lies the generated node under the

momentum variant, C' . Comparing the generation of C and C' as shown in (M1) and (M5), it is evident that the three nodes have produced a larger angle in the Momentum variant than the original approach, thus successfully introducing more curvature on a local scale.

It is noteworthy that the addition of momentum may affect different nodes differently. This difference becomes evident when there exists a large difference of angles between nodes, as shown in *Figure 4.2 (middle)*. (M6) shows a moderate angle between the three nodes, hence the shift in angles, θ_1 , is moderate, whereas (M7) and (M8) show the effects of the shift in angles when the angle between the three nodes is smaller and larger, θ_2 and θ_3 respectively, when using the same strength coefficient k . From the comparisons, we learn that with a constant k , the size of the angle made by the three nodes is directly proportional to the shift in angles from the application of momentum.

Figure 4.3 (middle) shows an example pattern after applying momentum to space colonization. The difference between before and after the modification is evident, as the structure includes more curvature. The curvatures of some branches have a higher resemblance to the curvatures of natural rivers, but some extremely sharp turns and zig-zag patterns are also introduced in some branches as byproducts, which are rare in natural river patterns, hence lowering the overall degree of realism for an artificial river pattern.

4.3 Space Colonization Curl

To alleviate the problem of zig-zag patterns in the structure, we propose an alternative modification to the space colonization algorithm coined Curl. The Curl variant can be considered as a generalized version of Momentum with a randomization element, as the principles are similar, such as the modification occurring at the same step in space colonization, and it also alters the angle of the next node generated. The difference between Curl and Momentum is that Curl does not rely on the direction vector of the previous node to determine the shift in the next node, but it simply uses a random variable.

Essentially, before the node is generated, it is rotated at a random small angle either clockwise or anticlockwise, in which the direction is predetermined depending on the node. The direction of the rotation for the first node is preset, then the direction is changed after each set of c new nodes, where c is the length of each section of the pattern before bending toward the other direction. Then the angle of rotation is randomized to a value between 0 and θ_c radians.

The parameter θ_c can be interpreted as the degree of shift in angles allowed. The higher is θ_c , the greater shifts are allowed, and more curvature is introduced,

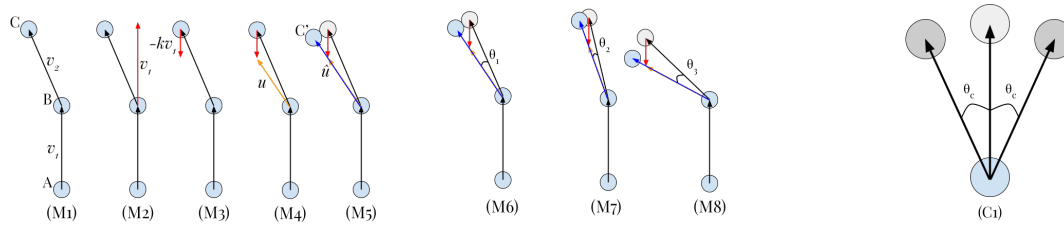


Figure 4.2: The effects of the proposed variants on the generated patterns. Left: The process of determining the position of C' , i.e. the node generated instead of C using Momentum. Middle: The different effects of Momentum on generating nodes at different angles. Right: The possible range of nodes to be generated, where the light grey node signifies the next node if the original space colonization is applied, and the dark grey nodes signify the left and right extremes of the possible new nodes.

and vice versa. Although choosing too high of a value for θ_c may result in unpredictable and chaotic patterns. The effects of Curl on the node generation are illustrated in Figure 4.2 (right).

As shown in Figure 4.3 (right), the patterns have become more unpredictable, and more curvature and a higher complexity have been introduced to the structure, the number of extremely sharp turns and zig-zag patterns has also been reduced when compared with the Momentum variant. Dissimilarity 2 has also been reduced, as the generated branches are not parallel with one another. However, the resultant patterns are still significantly unrealistic for river patterns, as there is too much detail at the ends of the structure, and Dissimilarity 3 still occurs in the patterns, as there exist short, perpendicular branches across long branches, which is unlikely to occur naturally.

4.4 Trimmed Space Colonization

To reduce Dissimilarity 3 and remove details at the ends of structures, a postprocessing step, namely trimming, is devised to remove these detailed endings and short branches, hence to further improve the degree of realism. Before the procedure of trimming is outlined, definitions of *children*, *parents*, and *depth* of nodes need to be introduced. For the purpose of trimming, we say that node Y is a child of node X , and X is the parent of Y , if Y is generated from the basis of X , and the *depth* of a node is defined recursively as the following.

- If a node has no children (i.e. is a leaf), then it has a depth of 0.
- If a node has one child of depth d , then it has a depth of $d + 1$.
- If a node has n children of depths $\{d_0, \dots, d_{n-1}\}$, then it has a depth of $\max(\{d_0, \dots, d_{n-1}\}) + 1$.

Now the procedure of trimming may be introduced. Trimming takes place after the generation of the entire

tree structure, firstly the depths of all nodes in the structure are computed, then the nodes X that satisfy both *Inequality 1* and *Inequality 2* are removed from the structure, where *Inequality 1* and *Inequality 2* are outlined below, and t is a preset variable, named the trimming threshold.

$$\text{depth}(X) < t \quad (1)$$

$$\text{depth}(X) < \text{depth}(\text{parent}(X)) - 1 \quad (2)$$

We require both inequalities to hold for the removal of the node. The reasoning for the first inequality is intuitive, as all nodes that are “close” (defined by t) to endings are removed. The second inequality is not necessary if only one structure is generated, however, if multiple river patterns are generated simultaneously on the same surface, then the exclusion of the second inequality would cause gaps between the structures. To prevent this, we include the second inequality, such that only small, perpendicular branches and ending branches with excessive detail are removed from the structure, whereas regular endings are kept in place.

Since trimming is a postprocessing step, hence the modification can be applied in parallel with each of the Momentum and Curl modifications. This results in a total of six variants of space colonization, i.e., original, momentum, curl, trimmed original, trimmed momentum, and trimmed curl. Some patterns from the trimmed variants are shown in Figure 4.4. Comparing the patterns, it is evident that the amounts of detail in the structures have decreased from the “untrimmed” variants, as there exist fewer shorter branches in the patterns.

Referring back to the desired qualities outlined for generated patterns, the degree of curvature has increased in the patterns, and some parameters (k , c , θ_c) have been introduced to control the degree of curvature; as a result of the enhanced curvature, the branches in generated patterns are no longer parallel, and the patterns

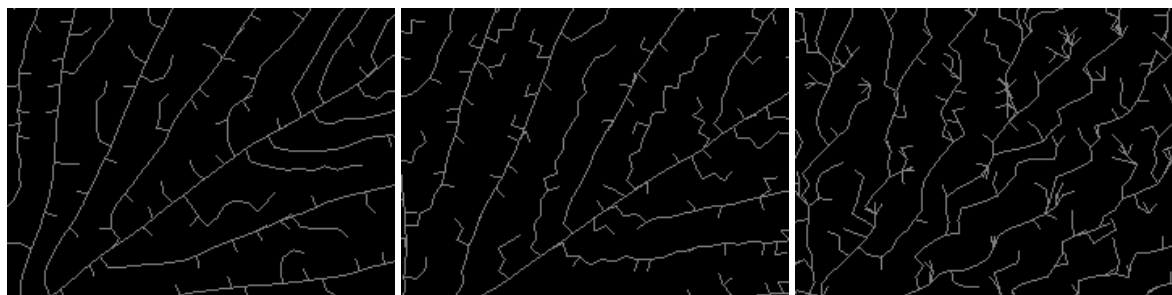


Figure 4.3: Example patterns generated from different variants of space colonization: Original (left), Momentum (middle), Curl (bottom). As shown in the patterns, Momentum introduces more curvature into the patterns but also introduces unrealistic zig-zag patterns, and Curl also introduces more curvature and adds more complexity to the patterns.

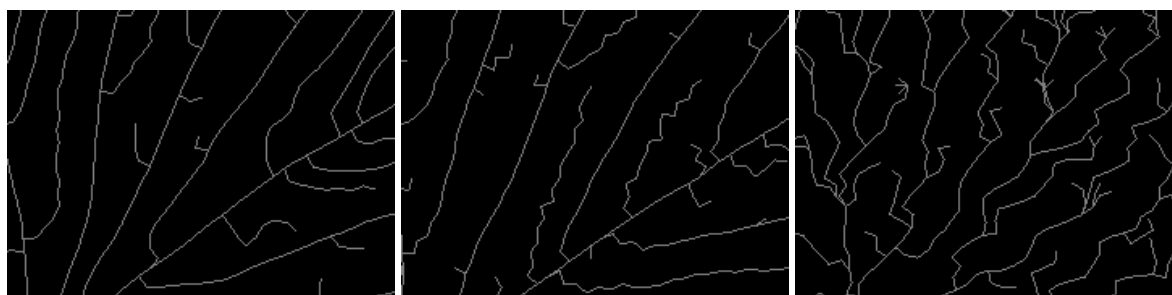


Figure 4.4: Example patterns generated from different variants of trimmed space colonization: Trimmed Original (left), Trimmed Momentum (middle), Trimmed Curl (right). Compared to the patterns shown in Figure 4.3, short branches at the ends of patterns are pruned after trimming, and the amount of detail in the patterns reduces.

are also less predictable; the option to remove lateral branches has also been introduced using trimming. In summary, the dissimilarities outlined in Subsection 4.1 have been reduced. To further examine the degree of realism each space colonization variant provides when generating river patterns, a user study is conducted, and further details are discussed in Subsection 6.1.

5 IMPLEMENTATION

To produce results from the generation algorithms outlined above, a system incorporating the algorithms was developed using C++14 and OpenGL3.3. The IDE (Integrated Development Environment) used for system development is Microsoft Visual Studio Community 2022. GitHub was also used for version control of the written code. The system was built using a 64-bit Windows 10 Home operating system, on a Dell Inspiron 15-3568 with 8GB RAM, and an Intel(R) Core(TM) i5-7200U CPU.

During development, several headers in the C++ Standard Library were used, which are chrono, cmath, iostream, random, and vector. No external libraries were used for C++. Some commonly used external libraries for OpenGL were also used, which are Glad (OpenGL Loading Library), GLFW (Graphics Library

Framework), GLM (OpenGL Mathematics), and KHR (Khronos).

Some algorithms were implemented and tested with Python before being implemented with C++, for ease of development and debugging. The version of Python used is Python 3.7.2. Some modules and libraries were used during the process, which are Matplotlib, NumPy, OpenCV, random, and time.

In the evaluation process (further discussed in Section 5), some image processing was conducted. The image processing program for raw images was written with Python 3.7.2 using the IDE Visual Studio Code, with the assistance of the libraries Matplotlib, NumPy, OpenCV, and Skimage. Software Microsoft Paint and Microsoft Paint 3D were also used for the viewing and cropping of the images.

Lastly, the processing of raw data received from responses to the user study was carried out using Microsoft Excel.

6 EVALUATION

To answer our research question whether the space colonization algorithm can be used to generate realistic river patterns, we conducted a user study comparing the

realism of real river patterns with river patterns generated using the algorithms outlined in Section 4.

6.1 Methodology

To evaluate the realism of river patterns generated from algorithms outlined in Subsection 4.1, a user study was carried out in the form of an online questionnaire, and it was propagated through private messaging among friends and classmates. The study was anonymous and participation was voluntary.

The questionnaire begins with a list of background questions to determine the demographic groups of the participant. The information on the gender, age, and visual conditions (if any) of the participants is collected, as well as their level of experience in the fields of geography, map-reading, and visual arts.

The demographic information of the participants is collected, in confidence, for future reference, such that any potential hidden relationships between demographic variables and observed variables may be discovered if relevant.

The questionnaire then asks the participant to rate the degree of realism of 30 river patterns on a 7-point Likert scale. Each river pattern is presented with an image, where the areas within a river are labeled white, and the areas outside are labeled black, some examples are shown in *Figure 6.1*.



Figure 6.1: Examples of patterns presented in the user study questionnaire.

There exists a mix of real and generated river patterns within the 30 river patterns shown to users, 15 of which are collected from real, natural rivers, and the remainder 15 are patterns generated from variants of space colonization. Out of the 15 generated patterns, 3 are generated from the original space colonization algorithm, 1 from Momentum, 1 from Curl, 4 from Trimmed original, 3 from Trimmed Momentum, and 3 from Trimmed Curl. All patterns used in the user study are shown in *Table 6.8*.

The information of whether a pattern is from a real river or generated is withheld from the participant, hence blinding is used to reduce bias and ensure that participants do not rate patterns from different origins differently.

To ensure that only the images presented only contain information about the patterns of the real/generated

ivers, some preprocessing is necessary. Online images of river maps often contain information irrelevant to patterns, such as labels and colors, and it needs to be removed before being presented to the participants, hence a 4-step preprocessing procedure is applied to all river maps retrieved through online sources. The procedure is outlined below.

1. Color extraction: Choose a range of RGB colors that include all pixels relevant to the river patterns.
2. Manual label removal and resizing: Use image editing software to erase any remaining text, and resize the image to an appropriate scale for cropping.
3. Skeletonization: Apply a skeletonization algorithm, such as the Zhang-Suen Thinning algorithm [ZS84], such that the information on the widths of rivers is removed.
4. Snippet extraction: Crop the image to a preset, fixed size. (The images used in this user study are 100 pixels high and 100 pixels wide.)

To ensure that the generated patterns are representative, a range of parameters are used across generations of the same variant, and the snippets of patterns are cropped using a randomly generated set of image coordinates. Furthermore, to keep the colors and scales in the images consistent between real and generated patterns, steps 3 and 4 of the preprocessing procedure are also applied to the generated patterns.

One potential problem with this questionnaire is that each participant has a different view on the “realism” of river patterns. Some participants may generally rate patterns higher on the Likert scale than others, and vice versa, this may skew the overall distribution of the ratings. But since we are comparing the distributions of two subgroups of data, each subgroup of data is likely skewed to a similar degree, hence this does not pose a problem in the evaluation process.

6.2 Results

Out of the 52 participants in this user study, there are 34 males, 16 females, and 2 preferred to not specify their gender. The majority of participants (49 participants) are between 18 and 25 years of age, with a mean of 19.94 years and a standard deviation of 1.30 years, and there are 3 participants outside of this range, who are 45, 52, and 52 years of age. 7 participants reported having myopia (short-sightedness). The results of the other background questions are shown in *Table 6.2*.

Table 6.2: Distribution of experience levels on different skills of participants in the user study.

Skill	None	Little	Some	Much
Geography	34	8	10	0
Map-reading	7	22	20	0
Visual Arts	22	26	3	1

Regarding the questions for rating the realism of the river patterns, the average realism scores² across multiple meaningful groups of patterns are calculated for each participant, such as the average score for all real patterns given by a participant. This produces a list of average scores for each participant. Then the mean and median scores are calculated across all participants, and the results are shown in *Table 6.3-6.5*.

Table 6.3: The mean and median realism score between real patterns and generated patterns across all participants.

Pattern Type	Mean	Median
Real	4.41	4.47
Generated	3.94	4.03

Table 6.4: The mean and median realism score between untrimmed generated patterns and trimmed generated patterns across all participants.

Pattern Type	Mean	Median
Untrimmed	3.53	3.60
Trimmed	4.15	4.20

Table 6.5: The mean and median realism score between all variants of space colonization across all participants.

Variant	Mean	Median
Original	3.31	3.00
Momentum	4.67	5.00
Curl	3.02	3.00
Trimmed Original	3.85	3.75
Trimmed Momentum	4.19	4.00
Trimmed Curl	4.53	4.67

From the results shown in *Table 6.3-6.5*, we observe that for the participants in the user study, on average, real patterns achieved a higher realism score than generated patterns, trimmed generated patterns achieved a higher realism score than untrimmed patterns, and out of all space colonization variants, momentum achieved the highest realism score, with trimmed curl achieving the second highest, and both achieved a higher realism score than real patterns.

6.3 Significance Testing

The research question asks whether space colonization and its variants can generate realistic river patterns, we

can reframe the question to whether space colonization and its variants can generate patterns that are indistinguishable from real river patterns, and that question can be answered from the data collected from the user study, by comparing the realism scores between groups of data. This can be achieved by using a series of significance tests.

There are several options of significance tests for Likert scale data (i.e. ordinal categorical data), some commonly used ones are the Mann-Whitney U test, Wilcoxon Signed-Rank test, and paired t-test. Different tests differ in requirements within the data and their strictness. Given the nature of our dataset, the Mann-Whitney U test is unsuitable, since two groups of compared data are collected from the same set of participants, hence are dependent [Gle22]. Thus Wilcoxon Signed-Rank test and paired t-test are used for significance testing. Although some commonly used requirements for the paired t-test are that each group of data needs to be normally distributed, and the data needs to be numerical, which does not comply with the collected Likert scale data. However, there has been research showing that the paired t-test is robust even on skewed data and ordinal categorical data [Nor10]. Hence despite the requirements stated above, the paired t-test is used on the collected data.

For significance testing, each variant or each collection of variants is treated as a group, and all groups, except for the Real group, are compared to the Real group for any significant difference using both the two-sided Wilcoxon Signed-Rank test and the two-sided paired t-test, with 95% confidence. The null hypothesis (H_0) and alternative hypothesis (H_1) are listed below, and the results of the tests are shown in *Table 6.6-6.7*.

H_0 : There does not exist a significant difference between the groups, i.e. the patterns in the two groups are indistinguishable and the generated patterns are considered realistic.

H_1 : There exists a significant difference between the groups, i.e. the patterns in the two groups are distinguishable and the generated patterns are considered unrealistic.

Table 6.6: The p-values of the two-sided Wilcoxon Signed-Rank tests (95% confidence) comparing the Real group with other groups.

Group	p-value
Generated	0.000
Untrimmed	0.000
Trimmed	0.003
Original	0.000
Momentum	0.087
Curl	0.000
Trimmed Original	0.000
Trimmed Momentum	0.073
Trimmed Curl	0.238

² A "realism score" is measured on a 7-point Likert scale, where 1 indicates highly unrealistic, and 7 indicates highly realistic.

Table 6.7: The p-values of the two-sided paired t-tests (95% confidence) comparing the Real group with other groups.

Group	p-value
Generated	0.000
Untrimmed	0.000
Trimmed	0.002
Original	0.000
Momentum	0.186
Curl	0.000
Trimmed Original	0.000
Trimmed Momentum	0.044
Trimmed Curl	0.316

The common usage of significance tests is to reject the null hypothesis, and the notion of accepting the null hypothesis is discouraged. However, there has been research showing that it is adequate to accept the null hypothesis for certain scenarios [Fri95]. We aim to investigate whether a group of patterns are realistic, hence the goal of the significance tests is to accept the null hypothesis. If the p-value of a significance test is greater than 0.05, then the significance test is considered successful, and vice versa.

6.4 Discussion

From the results of the significance tests, we may conclude that the Trimmed Curl variant produces patterns that are indistinguishable from real river patterns, as the null hypothesis is accepted in both scenarios, also notably with relatively high p-values.

The variants Original, Curl, and Trimmed Original are deemed unsuitable for river pattern generation, as the null hypothesis is rejected with minuscule p-values.

The Trimmed Momentum variant passes one of the two significance tests, which implies that it is on the borderline to being suitable for river pattern generation, but since Trimmed Curl outperforms Trimmed Momentum significantly, the former is preferred over the latter.

Surprisingly, the null hypothesis for the Momentum variant is accepted, as the variant is not expected to perform well. However, since there is only one pattern out of the 30 that is from the Momentum variant, we cannot conclude that it produces convincing river patterns in general, as the one pattern used does not nearly cover the large range of possible patterns across different parameters. Whereas there are three patterns used for Trimmed Curl across ranges of parameters, covering a large range of possible patterns. At this stage, we cannot conclude whether the Momentum variant produces realistic river patterns, and another user study may need to be conducted to reach a more conclusive outcome.

For larger groups (Generated, Untrimmed, Trimmed), the null hypotheses are all rejected. This is to be

expected since if at least one algorithm in the collection of algorithms produces unrealistic patterns, then the collections of patterns cannot be realistic altogether.

From the results of the user study, we can conclude that the trimmed curl variant of space colonization generates local patterns that are indistinguishable from real river patterns. This means that it can be used to reliably generate realistic local river patterns. Secondary choices are the momentum and trimmed momentum variants. However, since no advantages have been discovered using the secondary variants, hence the trimmed curl variant is the optimal approach for generating realistic river patterns.

6.5 Limitations


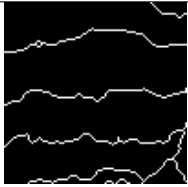




























There are several limitations to the evaluation of this research, some with potentially significant impact on the conclusion are outlined and discussed in this subsection.

There is one major limitation regarding the user study. The patterns shown to the participants only capture local snippets of the structures, whereas, for larger areas, or global snippets with a wider view of the structures, the degree of realism that algorithms deliver may differ. However, for the purpose of this research project, local snippets are sufficient to capture the essence of the patterns and provide meaningful results.

Another limitation of the user study, as discussed briefly above, is the limited number of patterns from each variant. Considering that six variants were evaluated simultaneously, the number of patterns for each variant needed to be controlled, as participants may lose interest, hence reducing the quality of responses, if a multitude of patterns is shown. To resolve this, we assigned more patterns to variants that are expected to perform well, and fewer to those not, so that there is more supporting evidence for a well-performing variant if it receives positive results. However, this results in the lack of evidence for the variants that are not expected to perform well, such as the Momentum variant as seen in the results, and another user study may need to be conducted, incorporating more patterns from that variant, to more conclusively evaluate its performance.

The evaluation of the river pattern generation component of the research has proven to be quite successful. However, the generated river patterns are limited to be local, i.e. on a small scale, whereas for more general uses of terrain generation, realistic global patterns may be more beneficial. A possible extension of the research is to conduct another user study comparing real patterns and generated patterns on a larger scale, such that the

Table 6.8: 30 patterns shown (in order) in the user study and their origins.

					
Pattern ID	1	2	3	4	5
Group(s)	G0	G0	G1, G3, G9	G1, G3, G8	G0
Origin	Ohio River	Orinoco River	Trimmed Curl	Trimmed Momentum	Indus River
					
Pattern ID	6	7	8	9	10
Group(s)	G1, G3, G9	G1, G3, G7	G1, G3, G8	G0	G1, G3, G7
Origin	Trimmed Curl	Trimmed Original	Trimmed Momentum	Congo River	Trimmed Original
					
Pattern ID	11	12	13	14	15
Group(s)	G1, G2, G6	G1, G3, G8	G1, G2, G5	G0	G0
Origin	Curl	Trimmed Momentum	Momentum	Yangtze River	Volga River
					
Pattern ID	16	17	18	19	20
Group(s)	G1, G3, G7	G0	G1, G3, G9	G0	G0
Origin	Trimmed Original	Amur River	Trimmed Curl	Irrawaddy River	Amazon River
					
Pattern ID	21	22	23	24	25
Group(s)	G0	G0	G0	G0	G1, G2, G4
Origin	Thames River	Murray River	Mississippi River	Tigris River	Original
					
Pattern ID	26	27	28	29	30
Group(s)	G0	G1, G2, G4	G1, G3, G7	G1, G2, G4	G0
Origin	Zambezi River	Original	Trimmed Original	Original	Godavari River

performance of the devised algorithms for global patterns may also be determined. Another possible extension is to extend the proposed methods, such that they are also capable of generating a wider range of river patterns, as described in Subsection 2.1.

7 CONCLUSION AND FUTURE WORK

We explored the usefulness of the space colonization algorithm [RLP07] for river pattern generation, and we devised several variants of the algorithm to achieve higher measures of realism for this purpose. We showed that one of the variants, Trimmed Curl, can generate patterns that are indistinguishable from real patterns extracted from natural rivers. This has proven to be novel, as the original space colonization algorithm did not return promising results from the user study, similarly for several other innovated variants.


The invention of the Trimmed Curl variant broadened the usage of space colonization to be not limited to tree generation, but also river generation. It also serves as a more computationally inexpensive alternative to existing river generation algorithms, such that rivers may be artificially generated on a larger scale, at a faster rate.


In future work we would like to explore how synthesised river patterns can be used for map and terrain generation. Since terrains in game engines are usually represented as greyscale maps, a possible solution might be an exemplar-based texture synthesis technique finding for river sections patches of matching terrain information and then completing the greyscale texture using a texture infilling method [NWDL14].

8 REFERENCES

- [Arc11] Archer, T. (2011). Procedurally generating terrain. In *44th annual midwest instruction and computing symposium, Duluth (pp. 378–393)*.
- [BF02] Benes, B., & Forsbach, R. (2002). Visual Simulation of Hydraulic Erosion. *Journal of WSCG*, 10(1), 79–86.
- [CBCB+16] Cordonnier, G., Braun, J., Cani, M.-P., Benes, B., Galin, E., Peytavie, A., & Guérin, E. (2016). Large Scale Terrain Generation from Tectonic Uplift and Fluvial Erosion. *Computer Graphics Forum*, 35(2), 165–175. doi:10.1111/cgf.12820
- [DP10] Doran, J., & Parberry, I. (2010). Controlled Procedural Terrain Generation Using Software Agents. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(2), 111–119. doi:10.1109/tciaig.2010.2049020
- [Fri95] Frick, R. W. (1995). Accepting the null hypothesis. *Memory & Cognition*, 23(1), 132–138. doi:10.3758/bf03210562
- [GGGP+13] G enevaux, J., D., Galin,  ., Gu erin, E., Peytavie, A., & Bene s, B. (2013). Terrain generation using procedural models based on hydrology. *ACM Transactions on Graphics*, 32(4), 1. doi:10.1145/2461912.2461996
- [Gle22] Stephanie Glen. (2022). *Mann Whitney U Test: Definition, How to Run in SPSS*. From StatisticsHowTo.com: Elementary Statistics for the rest of us!. Accessed 9 Nov 2022. <https://www.statisticshowto.com/mann-whitney-u-test>
- [NIWA18] National Institute of Water and Atmospheric Research. (2018). *Map of New Zealand Rivers* | NIWA. Accessed 2 October 2022. <<https://niwa.co.nz/freshwater/nzffd/NIWA-fish-atlas/map-of-NZ-rivers>>.
- [Nor10] Norman, G. (2010). Likert scales, levels of measurement and the “laws” of statistics. *Advances in Health Sciences Education*, 15(5), 625–632. doi:10.1007/s10459-010-9222-y
- [NWDL14] Nguyen, H. M., W unsche, B. C., Delmas, P., Lutteroth, C. (2014). Poisson Blended Exemplar-based Texture Completion. *Proceedings of the Thirty-Seventh Australasian Computer Science Conference*. 2014, 147, 99–104.
- [PDGC+19] Peytavie, A., Dupont, T., Gu   rin, E., Cortial, Y., Benes, B., Gain, J., & Galin, E. (2019). Procedural Riverscapes. *Computer Graphics Forum*, 38(7), 35–46. doi:10.1111/cgf.13814
- [RLP07] Runions, Adam & Lane, Brendan & Prusinkiewicz, Przemyslaw. (2007). Modeling Trees with a Space Colonization Algorithm. *Natural Phenomena*. 63–70.
- [Ros94] Rosgen, D. L. (1994). A classification of natural rivers. *CATENA*, 22(3), 169–199. doi:10.1016/0341-8162(94)90001-9
- [Twi04] C.R. Twidale (2004). River patterns and their meaning. *Earth-Science Reviews, Volume 67, Issues 3-4, Pages 159–218*. ISSN 0012-8252.
- [vHYW11] van den Hurk, S., Yuen, W., & W unsche, B. (2011) Real-time Terrain Rendering with Incremental Loading for Interactive Terrain Modelling. *Proc. of GRAPP*, 181–186.
- [Way19] Wayback Machine/Central Intelligence Agency. (2019). *Field Listing: Waterways* | *The World Factbook - CIA*. <https://web.archive.org/web/20190111040953/https://www.cia.gov/library/publications/the-world-factbook/fields/386.html>.
- [ZS84] Zhang, T. Y., & Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3), 236–239. doi:10.1145/357994.358023

AutogrASPing pose of virtual hand model using the Signed Distance Field real-time sampling with fine-tuning

Marcin Puchalski 
Jan Dlugosz University
Faculty of Science & Technology
Armii Krajowej 13/15
42-200 Częstochowa, Poland
m.puchalski@ujd.edu.pl

Bożena Woźna-Szcześniak 
Jan Dlugosz University
Faculty of Science & Technology
Armii Krajowej 13/15
42-200 Częstochowa, Poland
b.wozna@ujd.edu.pl

ABSTRACT

Virtual hands have a wide range of applications, including education, medical simulation, training, animation, and gaming. In education and training, they can be used to teach complex procedures or simulate realistic scenarios. This extends to medical training and therapy to simulate real-life surgical procedures and physical rehabilitation exercises. In animation, they can be used to generate believable pre-computed or real-time hand poses and grasping animations. In games, they can be used to control virtual objects and perform actions such as shooting a gun or throwing a ball. In consumer-grade VR setups, virtual hand manipulation is usually approximated by employing controller button states, which can result in unnatural final hand positions. One solution to this problem is the use of pre-recorded hand poses or auto-grasping using physics-based collision detection. However, this approach has limitations, such as not taking into account non-convex parts of objects, and can have a significant impact on performance. In this paper, we propose a new approach that utilizes a snapshot of the Signed Distance Field (SDF) of the area below the user's hand during the grab action. By sampling this 3D matrix during the finger-bending phase, we obtain information about the distance of each finger part to the object surface. Comparing our solution to those based on discrete collision detection shows better visual results and significantly less computational impact.

Keywords

Grasping; Virtual Reality; Visualization; Interaction; Animation

1 INTRODUCTION

The Virtual Reality (VR) technology has shown significant potential in studying disabilities, injuries, and rehabilitation. It can be used to help patients with motor impairments or neurological disorders regain functions and improve their quality of life [COC22]. By providing a virtual representation of patients' hands, therapists can design tailored rehabilitation programs that allow patients to practice movements and exercises in a safe and controlled environment [IWL⁺22, FSY⁺19].

VR technology has demonstrated great promise for use in education. One study [ŽCJ18] shows how a prototype application can be used to teach mathematics, while the feedback received reveals new potential research problems and, for example, the importance of virtual hands.

Visualizing virtual hands in VR has a significant impact on how users perceive the overall experience [JYM⁺20]. Studies [AHTL16, DMF⁺19, LKRI20] have shown that, while the resemblance of human hands is not important for achieving a sense of agency (the quality of tracking is more important), it is crucial for achieving a sense of ownership.

Research [EWB20] has shown that even with passive haptic feedback, low-cost visualization of virtual hands and snapping them to predefined positions, rotations, and poses on manipulated objects significantly increase presence and user experience. Another study [KSF⁺18] has revealed that introducing virtual hands while using a keyboard paired with a VR headset raises typing speed.

The recent study [CLL22] has demonstrated that providing feedback using a virtual hand and a virtual targeted object in a virtual environment produces kinematic patterns similar to those observed in physical environments.

To prevent the interpenetration of two dynamic objects moving in space (such as a virtual hand and the object one interacts with), the term God-object was introduced [ORC07]. It can be generalized to individual

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

phalanges [JSF12]. The solution assumes that the visual representation of the hand pursues the target which is the position of the user's hand, as read by any sensors while maintaining collision detection with other objects and taking into account constraints between objects.

Another way to increase the realism of a VR experience is to depict the grasping of virtual objects in a way that appears more lifelike. One effective technique for achieving this is to use a realistic hand pose which shows the hand holding the object as it would be in the real world [LC21].

Earlier research [BI05] described a spring model in which the finger parts and the base of a virtual hand follow the movement of tracked hand parts. This model includes collision detection, which prevents the spring hand from penetrating grasped objects. It can also produce forces and torques for a physically realistic response to the grasped objects. Authors assumed that visual feedback, due to the psychological phenomenon of visual capture [WW80], which is not an exact representation of the real hand configuration is less distracting to users than an exact representation that exhibits the visual interpenetration artifacts. This was confirmed by the later study [CNS⁺19], where the more general term of outer hand (OH) was used instead of spring hand.

Other research [LGAMB06] proposes, in addition to a rigid model for dynamic simulation, using a deformable model to resolve local contact with friction and surface deformation of fingertips.

These types of auto-grasping solutions are often referred to as hybrid solutions with physics hands in VR and are still widely researched with different approaches, for example, using machine-learning [TWMZ19]. In the latter by using discrete collision detection, the preprocessing algorithm is sampling multi-dimensional grasp space using random configurations of hand poses. This approach requires offline preprocessing for each object but produces plausible online grasp poses after initial user-guided movement and pre-grasp pose from tracked hand.

The auto-grasping problem also extends to other fields including robotics where it is important to properly position and move the robotic arm so that it can achieve a stable grip and avoid collisions with the environment.

In [BCO⁺21] authors propose a Volumetric Grasp Network (VGN) - a 3D convolutional neural network that can detect the pose of a grasp in real-time with 6 degrees of freedom (DOF). The network takes in a Truncated Signed Distance Function representation of a scene and produces a volume of the same spatial resolution as the output. Each cell in the volume contains information about the predicted quality, orientation, and width of a grasp that would be executed at the center of the voxel. The network has been trained

on a synthetic grasping dataset created using physics simulation.

A more recent paper [TWH⁺22] proposes a method of synthesis of grasping poses with a machine learning-enabled gradient method using dense 256^3 precomputed Signed Distance Field (SDF) of the grasped object as an input. Although the proposed method synthesizes stable poses with a high contact surface, it is too slow for online usage.

Another recent paper [SHX⁺22] proposes using a machine learning approach with novel sampling of the Voronoi diagram between two close 3D geometric objects. Although the resulting method yields grips with high success rate, it is still prone to unnatural final poses.

In many cases VR environments are implemented using game engines such as Unity [Uni05] with commercial autograsping libraries ([Clo20], [Ear20]). For performance reasons, these libraries rely on a method using discrete physics collision and overlap detection based on approximating the physical collision of each grasped object using one or many simple shapes or proxy convex meshes. For more complex concave objects, for example, introducing holes, it is necessary to decompose their surface into f.e. a set of convex shapes [Uni16]. This preprocessing step is required for each interactive object in the virtual environment. Due to its offline nature, this method is not suitable for animated skinned meshes.

Taking all of this into consideration, we ask whether it is possible to skip the preprocessing stages and use an online method that will work with animated skinned meshes while still working in a real-time environment.

In this paper, we propose an approach to autograsping that, unlike existing methods that use discrete physical collision and overlap detection, uses a snapshot of the SDF of the grasped object. In addition, unlike other methods using the SDF, we propose to use only a snapshot of a closed region under the virtual hand. We compare our method with existing methods based on discrete physical collision and overlap detection. We also indicate possible future work that can be developed based on our approach.

The paper is organized as follows. In Section 2, we define the research problem and existing physics-based method; in Section 3, we describe the proposed method and its implementation; and in Section 4, we report on the simulation results. Section 5 discusses possible limitations of proposed method and current implementation. Finally, in Section 6 we conclude the paper.

2 BACKGROUND

2.1 Visual representation of hands

Several software platforms are commonly used to create VR experiences, including Unity [Uni05] and Un-

real Engine. Virtual hand presence is widely implemented in most of them using popular VR frameworks like SteamVR [Val20] and OculusSDK [Met22b].

Virtual hands are displayed instead of or along with controller models. The movement of the controllers is translated into the approximate movement of the hand relative to the position and orientation of the controller in question. The pose of the virtual hand when the controller itself is also displayed is set to resemble the pose of the hand wrapping around the given controller. However, in order to increase the immersion, the display of the controller is usually omitted while using the display of the virtual hand in a neutral pose, such as an open palm. This is somewhat counter-intuitive as the user's hand is clamped around the controller. After a while, the impression of detachment from the experience blurs, but VR hardware manufacturers are making attempts to create controllers that would solve this problem. An example of this is the Valve Index Controller [Val19], which, with the help of a suitable hand strap, allows the user to relax and open his hand.

2.2 Physics Based Hands

Physics-based hands are used in VR applications to prevent virtual hands from intersecting with virtual objects [PZ05]. These hand models consist of a visual model that is visible to the user, and a physics model that is made up of simple 3D shapes (such as capsules and boxes) implemented into the physics engine. The physics model follows the movements of the tracked hand (i.e. controller). Simulated as a non-kinematic object it is not penetrating static objects while exerting forces on other dynamic objects upon collision.

2.3 Hand poses

Visualization of the selected hand pose is done with the help of a rigged skinned mesh hand model. The rig usually consists of 3-4 bones for each finger (one for each phalanx and metacarpal bone) and at least one bone for the base of the hand. As for the movement proximal phalanges joints' have 2DOF, while intermediate and distal phalanges have 1DOF.

Commonly used VR libraries [Val20, Met22b] also allow us to customize the appearance and animating poses of the virtual hand. These poses must be predefined for a given 3D hand model and its skeleton. The defined poses most often include closed and open hand poses, between which one blends the hand configuration. For example, depending on the state readings of the controller's buttons and of touchpads, we simulate the user's hand poses such as fist clenching, finger pointing, and palm waving. These libraries also allow for the creation of predefined poses used when detecting the action of grasping various objects. However, these poses must be predefined for each

object with similar shapes and sizes (e.g., a sphere and an apple). It involves setting the hand model in the target position and orientation relative to the object and then adjusting the rotation of the individual finger bones to reflect the grasping pose. In some cases, we can achieve mirror poses for the left and right hands through symmetry.

The OculusSDK library [Met22a] provides a tool for recording predefined poses for grasping objects using hand tracking, which has been implemented in the Meta Quest software. Thanks to it developers can quickly define natural hand configuration for the grasping pose of selected objects without the need for their manual rig setup.

Several possible grasping positions and poses can be defined for a single object but in each case, the entire process of manual posing must be repeated. Further, when grasping a virtual object in runtime, the hand pose is adjusted, with the position and orientation of the object snapped to the hand to match the defined pose (e.g., the handle of a cup should wrap around the index finger) or the hand to the object when the object is static or attached to environmental elements (e.g., a door handle). Snapping a dynamic object to the hand may result in the abrupt movement of other objects in clutter due to the immediate change in position and orientation of the grasped object.

2.4 Virtual hand auto-grasping

Auto-grasping is a method that allows the automation of grasping tasks, eliminating the need for manual preparation of specific poses for different types of objects in a variety of grasping configurations. It can heavily speed up the creation of predefined grasping poses and, more importantly, enable the online generation of the poses. It also means that virtual objects can be grasped from any position, eliminating the need to align them with predetermined positions and orientations, which can be cumbersome in cluttered environments.

The auto-grasping technique can be found in commercial toolkits for Unity [Uni05] such as Auto-Hand [Ear20] and HurricaneVR [Clo20]. These implementations require specifying only two hand configurations: an open, slightly exaggerated pose (like with muscles of the back of the hand clenched) and a fully closed one - opposite from the former one (like a clenched fist).

To enable the auto-grasping of a selected virtual object, it is necessary to define one or more colliders for the grasped object. These colliders can be simple shapes such as spheres, cubes, or proxy convex mesh colliders. It is not possible to use a convex mesh collider for skinned meshes, like those of animated characters.

Algorithm 1: Physics-based auto-grasping pose estimation

Input : *fingers* list of finger objects. Each object contains *Squish* property taking values 0..1 and is responsible for bending a given finger from open to closed pose, and *Tip* property referencing to finger's tip object $\Delta\alpha$ step from open to closed pose in normalized time
r_{tip} - radius of the area around fingertip in game units, typically in meters
foreach *finger* in *fingers* **do**
 for $\alpha \leftarrow 0$ to 1 **step** $\Delta\alpha$ **do**
 // bend current finger to α value
 finger.Squish $\leftarrow \alpha$;
 // Check if tip overlaps the grasped object
 if *OverlapSphere*(*finger.Tip.Position*, *r_{tip}*) **then**
 break
 end
 end
end

Additionally, single convex mesh colliders do not accurately reflect the details of virtual objects, such as cavities and holes.

Algorithm 1 illustrates the simplified process of the grasping phase of the virtual object within the hand range using physics-based method. During this phase, an interactive blending between the open and closed poses occurs for each finger. This blending is mapped to $\alpha \in [0, 1]$, where 0 means open pose, and 1 means closed pose. We change this value by a small step $\Delta\alpha \in (0, 1)$, typically 0.1, and using discrete overlapping detection, we check if the target object enters the sphere around the fingertip, then we stop bending the finger. It is repeated for each finger individually. Intuitively, we can identify two factors affecting performance - finger pose blending and subsequent collision detection.

The method proposed in this paper does not rely on physics, discrete overlapping detection, iterative pose blending, or multiple proxy convex meshes to model mesh holes. It can also work with animated skin meshes as shown in one of the attached video files.

3 THE SDF SAMPLING-BASED METHOD

In contrast to the physics-based approach (Algorithm 1) we do not rely on discrete overlap detection. Instead, we sample the SDF of the grasped object at voxel positions mapped to fingers' tips positions during the bending of each finger. Subsection 3.1 describes the initial

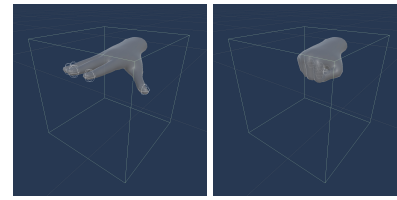


Figure 1: Hand model with an area of SDF generation and with fully closed pose

setup and prerequisites for the hand model we use for our method. At the start of the grasp phase, we compute the SDF (Subsection 3.2) of the grasped object but, for performance reasons, only in the clipped area around the virtual hand and in relatively low resolution. For optimizations, we also cache all possible fingertip positions mapped to the SDF texture space (Subsection 3.3). Values sampled from SDF at these positions (Subsection 3.4) are the Euclidean distances to the object's surface, and by comparing them with a minimum threshold we can determine when each fingertip penetrates the object and stop bending the finger. This is further described in Subsection 3.5. However, as opposed to the discrete overlap detection the sampled distance values can be used in a broader way to fine-tune the bending of the finger (Subsection 3.6).

3.1 Virtual hand configuration

For visualization, the 3D model of the hand provided with the OculusSDK library was used and further configured (Figure 1). Although we can use any rigged hand model even with a different number of fingers than the human hand. There is an added component on each finger responsible for bending the finger from the open to close position and for storing the finger's tip position.

Around the hand, we have defined an area with two purposes in mind. First, it acts as a trigger to detect a virtual object under the palm and to start interactive SDF computation of that object relative to hand position and orientation. Secondly, it is the clipping area of the computed SDF.

3.2 SDF computation

SDFs and implicit surfaces have been used for many years in computer graphics for creating and displaying shapes [Bli82, BW97, Har95, WGG99]. SDF can be described by the following mathematical function:

$$\phi(x) : \mathbb{R}^3 \rightarrow \mathbb{R} \quad (1)$$

that maps a point x in 3D space (represented in some predefined coordinate frame) to a real number. The value of $\phi(x)$ at a given point is the signed Euclidean distance from that point to the nearest point on a surface, where $\phi(x) = 0$. Points that are outside of the object being represented by the SDF have a value of

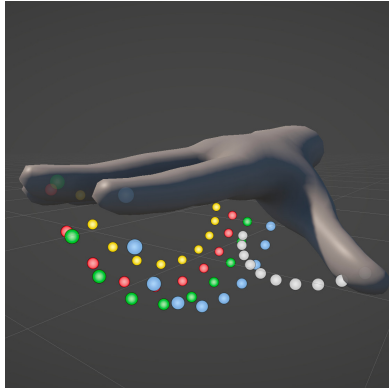


Figure 2: Hand model with all possible fingers' tip positions during auto-grasping

$\phi(x) > 0$, while points inside the object have a value of $\phi(x) < 0$.

In this paper, we've used Unity's light and fast real-time SDF generator Mesh-to-SDF [Uni22] that implements an algorithm from AMD TressFX library [Adv20]. Its main optimizations rely on taking into account the area near each triangle and then filling the rest of the SDF using linear or jump flood algorithm [RT06]. It is fast enough for real-time, especially for meshes with less than 10k triangles. In our implementation, we have assumed the SDF volumes size to be 16^3 , which in tests turned out to be sufficient and resulted in a fast computation.

3.3 Fingertip positions cache

During the auto-grasp, each finger's configuration blends from a fully open to a closed pose. The blend happens on each part of the finger by changing its local positions and orientation relative to the parent bone in the hierarchy. As a result of this blending, the finger's tip travels on a constant curve. For our implementation, we sample the tip positions on that curve by constant step $\Delta\alpha = 0.1$, where $\alpha = 0$ is the start of the curve, and $\alpha = 1$ is its end. All tip positions are stored in a 1D vector of positions. Each position in this vector is then transformed using Equation 2 from the hand local space to the SDF 3D texture space, where $x, y, z \in \mathbb{R}$ are the coordinates of transformed position, $bbox$ is the local bounding box size reflecting the area from which the SDF is created. This results in the constant cache of all possible fingertips' positions in SDF texture space.

$$T: \mathbb{R}^3 \rightarrow \mathbb{R}^3, T\left(\begin{pmatrix} x \\ y \\ z \end{pmatrix}\right) = \begin{pmatrix} \frac{2x}{bbox_x} \\ \frac{2y}{bbox_y} \\ \frac{2z}{bbox_z} \end{pmatrix} + \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix} \quad (2)$$

3.4 SDF sampling

To retrieve values stored in the SDF created by the Mesh-To-SDF algorithm we use a simple

multi-threaded GPU compute shader described by Algorithm 2. It takes a vector cache of possible tips' positions in 3D texture space and returns a 1D vector of sampled SDF values. By running this code on GPU we do not need to transfer the whole SDF texture from GPU memory to computer memory, and then do the linear sampling for non-integer texture coordinates on CPU. Additionally, we can sample many SDF values parallel using multiple GPU threads. After dispatching SDF sampling on GPU we use Unity's *AsyncGPUReadback* class allowing the asynchronous readback of sampled values without any stall on the CPU or GPU side.

Algorithm 2: Multithreaded SDF Sampling Compute Shader

Input : *tex* - SDF 3D texture;
 input_buffer - holding texture coordinates to sample;
 threadID - vector of three unsigned integer components x, y, z with thread indexes.
Output : *output_buffer* - holding sampling results
id \leftarrow *threadID* _{x}
coord \leftarrow coordinate at(*id*) in *input_buffer*
texel \leftarrow Sample3DTexture(*tex*, *coord*)
output_buffer[*id*] \leftarrow *texel*

3.5 SDF based auto-grasping

Algorithm 3: SDF based auto-grasping

Input : *resultArr* - result of the SDF sampling compute shader;
 $\Delta\alpha$ - the predefined step of finger on bending curve;
 minTipDistance - threshold value of SDF sample;
 fingers - list of finger objects.
α \leftarrow 0
stopped \leftarrow false
eFinger \leftarrow *fingers*.GetEnumerator()
eFinger.MoveNext()
foreach *result* in *resultArr* **do**
 if !*stopped* & *result* < *minTipDistance* **then**
 stopped \leftarrow true
 eFinger.Current.Squish \leftarrow *α*
 end
 α \leftarrow *α* + $\Delta\alpha$
 if *α* > 1.0 **then**
 if !*stopped* **then**
 eFinger.Current.Squish \leftarrow 1.0
 end
 α \leftarrow 0
 stopped \leftarrow false
 eFinger.MoveNext()
 end
end

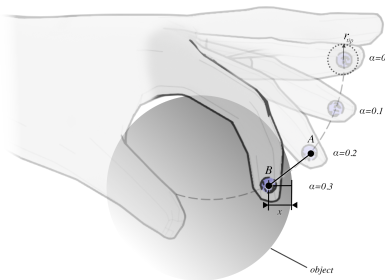


Figure 3: Index finger bending phase before and after penetrating object

After receiving a callback with sampled SDF values they can be processed on the CPU side by Algorithm 3. Its main loop iterates through the *resultArr* array holding the sampled SDF values in discrete positions on each finger's bending curves traversed by the $\Delta\alpha$ step. During this iteration, our algorithm also iterates through each finger component responsible for its bending. If the current SDF value from the *resultArr* is less than the specified minimum threshold value, then the current finger is bent to the corresponding position on the bending curve, any remaining samples for that finger are skipped, and the algorithm iterates to the next finger.

3.6 Fine tuning using SDF sample

The fingertip bend value is stored in the α variable taking a normalized value in the range 0..1 (from a fully open pose to a close one). Assuming the value of $\Delta\alpha = 0.1$ as in Figure 3, it may happen that for successive discrete positions, the fingertip is above the surface of the object (point A), and in the next step it is inside the object (point B). In methods using discrete overlap detection, one retracts the finger to point A by changing the value of the bend $\alpha \leftarrow \alpha - \Delta\alpha$. The finger bending is then repeated but this time for a smaller step, e.g. $\Delta\alpha = 0.01$.

This approach is not suitable when using SDF, as it would require another cache of fingertip positions and another round of SDF sampling. Fortunately, SDF samples give us more information than just discrete collision - specifically the distance to the nearest surface. In the proposed method, we use the SDF value x read at the time of penetration into the object. \overline{AB} is the segment between the last position of the fingertip before penetrating the object and the current position inside the object. \widehat{AB} denotes the arc between two points on the bending curve of the current finger. When $\Delta\alpha \rightarrow 0$ then $|\overline{AB}| = |\widehat{AB}|$, so we can assume for small values of $\Delta\alpha$ that $|\overline{AB}| \approx |\widehat{AB}|$. If $|\overline{AB}|$ is the approximated distance the finger traveled from point A to B on the curve, then the correction of α value to push the finger from the surface can be approximated with the following equation:

$$\alpha_{corr} = \frac{x - r_{tip}}{|\overline{AB}|} \Delta\alpha \quad (3)$$

4 RESULTS

The proposed method was implemented and tested in the Unity Editor environment [Uni05]. Our implementation is available online as a GitHub repository [SDF23]. Screenshots and timing measurements were done on the Apple MacBook Air M1 laptop.

METHOD	Bunny	Armadillo	Dragon	Pixel
Physics	0.08ms	0.04ms	0.1ms	0.08ms
Ours	0.01ms	0.02ms	0.03ms	0.02ms

Table 1: Avg timings of grasping methods for test objects

Figure 4 shows the comparison of the physics-based and the SDF sampling-based grasps on test objects. The first image in each set shows a physics-based grasp relying on overlap detection. The second image shows a simple convex mesh collider used for overlap detection. The third image shows the result of the proposed SDF sampling-based method, and the last one adds a visualization of the SDF sample. As we can infer from examples, the physics-based grasp produces worse grasps than the SDF sampling based. In most distinctive examples, the fingers stop before reaching the object's surface because they overlap already with a simple convex mesh collider.

Table 1 shows the average timings of physics-based and SDF sampling-based grasping methods for each of the tested models. On a note, the latter does not include the timings of SDF computation as it is not a part of the proposed method, depends on the selected implementation, and heavily relies on a specific model's triangle count. For a Pixel test model (with 499,548 triangle count) online SDF computation can result in >0.6ms overhead while for models with $\leq 10k$ triangle count the overhead is insignificant. Additionally, each *GPUAsyncCallback* can stall SDF sampling results for an additional frame.

5 LIMITATIONS

In addition to the last note, it is worth mentioning other limitations of the proposed method. First, the results are highly dependent on the initial position of the virtual hand. In the context of this paper, however, we do not elaborate on the stage of hand approach to the object surface. The proposed method and implementation focus only on detection around the fingertips and is subject to, for example, interpenetration of other parts of the fingers with grasped objects. We can also imagine a situation in which, during the bending phase of a finger with $\Delta\alpha = 0.1$, the fingertip misses a small obstacle on



Figure 4: Example grasps of test objects including Stanford Bunny, Stanford Dragon, Stanford Armadillo, and Pixel mascot presented in four configurations: physics-based grasp, convex collider used for physics-based grasp, SDF sampling based grasp and SDF visualization

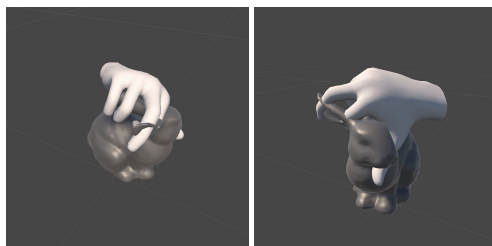


Figure 5: Examples of interpenetration

it's bending curve. This, however can be resolved by using smaller $\Delta\alpha$ values.

Figure 5 shows examples of results failed due to the above limitations.

6 CONCLUSIONS

In this paper, we have proposed the SDF sampling-based auto-grasping method as a replacement for the physics-based one used for example in popular commercial auto-grasping toolkits for the Unity game en-

gine. Using SDF sampling produces better visual results and the sampling method implementation is faster than the one based on discrete sphere overlap detection. The trade-off is that we also need to implement a selected SDF computation algorithm that for 3D models with bigger triangle counts can have an impact on performance. But taking into account the dynamic development of new SDF computation methods and that SDF computation can be tailored for our specific usage (low resolution and small clipped area of SDF computation) we can assume future improvements in this field.

Regarding possible interpenetration, in future work, we can focus on sampling SDF values for multiple contact points on the hand. We could also benefit from using different bending strategies like bending parts of each finger separately, using SDF sample values for wider optimizations, and leveraging more GPU-side computations.

As for the hand approach phase, it can also be solved using sampling the SDF around, for example, the

hand's palm center. Additional calculation of the SDF derivative at the same point will result in a near-surface normal that can be used to determine the orientation of the whole hand. For this reason, we do not present Q-distance comparisons as these depend strongly on the aforementioned stage.

Our auto-grasping method is not the first one using SDFs, but in opposition to other methods [BCO⁺21, TWH⁺22], it does not need any precomputations. It targets online usage, can be used with dynamic skinned meshes, and focuses only on the small clipped area for the SDF computations, not the entire object or environment. In future work, we would like to study the contextless clipped SDF samples as an input for a generalized machine learning approach to auto-grasping. Additionally, it can leverage from using imitation learning with grasping poses obtained using, i.e., Meta Quest hand tracking.

7 REFERENCES

- [Adv20] AMD, Inc. TressFX (2020). URL: <https://github.com/GPUOpen-Effects/TressFX>.
- [AHTL16] Argelaguet, F., Hoyet, L., Trico, M., and Lecuyer, A. The role of interaction in virtual embodiment: Effects of the virtual hand representation. In *2016 IEEE Virtual Reality (VR)*, pages 3–10. IEEE, New York (2016). DOI: 10.1109/VR.2016.7504682.
- [BCO⁺21] Breyer, M., Chung, J.J., Ott, L., Siegwart, R., and Nieto, J. Volumetric Grasping Network: Real-time 6 DOF Grasp Detection in Clutter. In J. Kober, F. Ramos, and C. Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 1602–1611. PMLR, Cambridge MA (2021). DOI: 10.48550/arXiv.2101.01132.
- [BI05] Borst, C. and Indugula, A. Realistic virtual grasping. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, pages 91–98. IEEE, New York (2005). DOI: 10.1109/vr.2005.1492758.
- [Bli82] Blinn, J.F. A Generalization of Algebraic Surface Drawing. *ACM Trans. Graph.*, 1(3):235–256 (1982). DOI: 10.1145/357306.357310.
- [BW97] Bloomenthal, J. and Wyvill, B. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., San Francisco (1997). ISBN 155860233X.
- [CLL22] Cai, Q., Li, J., and Long, J. Effect of Physical and Virtual Feedback on Reach-to-Grasp Movements in Virtual Environments. *IEEE Transactions on Cognitive and Developmental Systems*, 14(2):708–714 (2022). DOI: 10.1109/TCDS.2021.3066618.
- [Clo20] Cloudwalkin Games. Hurricane VR - Physics Interaction Toolkit (2020). URL: <https://assetstore.unity.com/packages/tools/physics/hurricane-vr-physics-interaction-toolkit-177300>.
- [CNS⁺19] Canales, R., Normoyle, A., Sun, Y., Ye, Y., Luca, M.D., and Jörg, S. Virtual Grasping Feedback and Virtual Hand Ownership. In *ACM Symposium on Applied Perception 2019, SAP '19*. ACM, New York (2019). ISBN 9781450368902. DOI: 10.1145/3343036.3343132.
- [COC22] Chen, J., Or, C.K., and Chen, T. Effectiveness of Using Virtual Reality-Supported Exercise Therapy for Upper Extremity Motor Rehabilitation in Patients With Stroke: Systematic Review and Meta-analysis of Randomized Controlled Trials. *J Med Internet Res*, 24(6):e24111 (2022). DOI: 10.2196/24111.
- [DMF⁺19] D'Alonzo, M., Mioli, A., Formica, D., Vollero, L., and Di Pino, G. Different level of virtualization of sight and touch produces the uncanny valley of avatar's hand embodiment. *Scientific Reports*, 9(1):19030 (2019). DOI: 10.1038/s41598-019-55478-z.
- [Ear20] Earnest Robot. Auto Hand - VR Interaction (2020). URL: <https://assetstore.unity.com/packages/tools/game-toolkits/auto-hand-vr-interaction-165323>.
- [EWB20] Elbehery, M., Weidner, F., and Broll, W. Haptic Space: The Effect of a Rigid Hand Representation on Presence When Interacting with Passive Haptics Controls in VR. In *19th International Conference on Mobile and Ubiquitous Multimedia, MUM '20*, page 245–253. ACM, New York (2020). ISBN 9781450388702. DOI: 10.1145/3428361.3428388.
- [FSY⁺19] Furmanek, M.P., Schettino, L.F., Yarossi, M., Kirkman, S., Adamovich, S.V., and Tunik, E. Coordination of reach-to-grasp in physical and haptic-free virtual environments. *Journal of NeuroEngineering and Rehabilitation*, 16(1):78 (2019). DOI: 10.1186/s12984-019-0525-9.
- [Har95] Hart, J. Sphere Tracing: A Geometric Method for the Antialiased Ray Tracing of Implicit Surfaces. *The Visual Computer*, 12 (1995). DOI: 10.1007/s003710050084.
- [IWL⁺22] Isenstein, E.L., Waz, T., LoPrete, A., Hernandez, Y., Knight, E.J., Busza, A., and Tadin, D. Rapid assessment of hand reaching using virtual reality and application in cerebellar stroke. *PLOS ONE*, 17(9):1–21 (2022). DOI: 10.1371/journal.pone.0275220.
- [JSF12] Jacobs, J., Stengel, M., and Froehlich, B. A generalized God-object method for plausible finger-based interactions in virtual environments. In *2012 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 43–51. IEEE, New York (2012). DOI:

- 10.1109/3dui.2012.6184183.
- [JYM⁺20] Jörg, S., Ye, Y., Mueller, F., Neff, M., and Zordan, V. Virtual Hands in VR: Motion Capture, Synthesis, and Perception. In *SIGGRAPH Asia 2020 Courses*, SA '20. ACM, New York (2020). ISBN 9781450381123. DOI: 10.1145/3415263.3419155.
- [KSF⁺18] Knierim, P., Schwind, V., Feit, A.M., Nieuwenhuizen, F., and Henze, N. Physical Keyboards in Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–9. ACM, New York (2018). DOI: 10.1145/3173574.3173919.
- [LC21] Lavoie, E. and Chapman, C.S. What's limbs got to do with it? Real-world movement correlates with feelings of ownership over virtual arms during object interactions in virtual reality. *Neuroscience of Consciousness* (2021). DOI: 10.1093/nc/niaa027.
- [LGAMB06] Le Garrec, J., Andriot, C., Merlhiot, X., and Bidaud, P. Virtual Grasping of Deformable Objects with Exact Contact Friction in Real Time. *WSCG '2006: short communications proceedings*, pages 87–92 (2006).
- [LKRI20] Lougiakis, C., Katifori, A., Roussou, M., and Ioannidis, I.P. Effects of Virtual Hand Representation on Interaction and Embodiment in HMD-based Virtual Environments Using Controllers. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 510–518. IEEE, New York (2020). DOI: 10.1109/VR46266.2020.00072.
- [Met22a] Meta. Creating Hand Grab Poses (2022). URL: <https://developer.oculus.com/documentation/unity/unity-isdk-creating-handgrab-poses/>.
- [Met22b] Meta. Interaction SDK Overview (2022). URL: <https://developer.oculus.com/documentation/unity/unity-isdk-interaction-sdk-overview/>.
- [ORC07] Ortega, M., Redon, S., and Coquillart, S. A Six Degree-of-Freedom God-Object Method for Haptic Display of Rigid Bodies with Surface Properties. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):458–469 (2007). DOI: 10.1109/TVCG.2007.1028.
- [PZ05] Pollard, N.S. and Zordan, V.B. Physically Based Grasping Control from Example. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05, page 311–318. ACM, New York (2005). ISBN 1595931988. DOI: 10.1145/1073368.1073413.
- [RT06] Rong, G. and Tan, T.S. Jump Flooding in GPU with Applications to Voronoi Diagram and Distance Transform. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, I3D '06, page 109–116. ACM, New York (2006). ISBN 159593295X. DOI: 10.1145/1111411.1111431.
- [SDF23] SDF2HandPose (2023). URL: <https://github.com/nosferathoo/SDF2HandPose>.
- [SHX⁺22] She, Q., Hu, R., Xu, J., Liu, M., Xu, K., and Huang, H. Learning High-DOF Reaching-and-Grasping via Dynamic Representation of Gripper-Object Interaction. *ACM Trans. Graph.*, 41(4) (2022). DOI: 10.1145/3528223.3530091.
- [TWH⁺22] Turpin, D., Wang, L., Heiden, E., Chen, Y.C., Macklin, M., Tsogkas, S., Dickinson, S., and Garg, A. Grasp'D: Differentiable Contact-Rich Grasp Synthesis for Multi-Fingered Hands. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, page 201–221. Springer-Verlag, Berlin, Heidelberg (2022). ISBN 978-3-031-20067-0. DOI: 10.1007/978-3-031-20068-7_12.
- [TWMZ19] Tian, H., Wang, C., Manocha, D., and Zhang, X. Realtime Hand-Object Interaction Using Learned Grasp Space for Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2623–2635 (2019). DOI: 10.1109/TVCG.2018.2849381.
- [Uni05] Unity Technologies. Unity Real-Time Development Platform (2005). URL: <https://unity.com>.
- [Uni16] Unity Technologies. The V-HACD library decomposes a 3D surface into a set of "near" convex parts. (2016). URL: <https://github.com/Unity-Technologies/VHACD>.
- [Uni22] Unity Technologies. Mesh-To-SDF (2022). URL: <https://github.com/Unity-Technologies/com.unity.demoteam.mesh-to-sdf>.
- [Val19] Valve Corporation. Valve Index - Upgrade your experience (2019). URL: <https://www.valvesoftware.com/en/index/controllers>.
- [Val20] Valve Corporation. Skeleton Input | SteamVR Unity Plugin (2020). URL: https://valvesoftware.github.io/steamvr_unity_plugin/articles/Skeleton-Input.html.
- [ŽCJ18] Žilak, M., Car, v., and Ježić, G. Educational Virtual Environment Based on Oculus Rift and Leap Motion Devices. *WSCG '2018: short communications proceedings*, pages 143–151 (2018). DOI: 10.24132/CSRN.2018.2802.18.
- [WGG99] Wyvill, B., Guy, A., and Galin, E. Extending The CSG Tree. Warping, Blending and Boolean Operations in an Implicit Surface Modeling System. *Comput. Graph. Forum*, 18:149–158 (1999). DOI: 10.1111/1467-8659.00365.
- [WW80] Welch, R. and Warren, D. Immediate perceptual response to intersensory discrepancy. *Psychological bulletin*, 88(3):638–667 (1980). URL: <http://europemc.org/abstract/MED/7003641>.

Temporal segmentation of actions in fencing footwork training

Filip Malawski
Institute of Computer
Science
AGH University of
Science and Technology
Krakow, Poland
fmal@agh.edu.pl

Marek Krupa
Institute of Computer
Science
AGH University of
Science and Technology
Krakow, Poland
mkrupa@agh.edu.pl

ABSTRACT

Automatic analysis of actions in sports training can provide useful feedback for athletes. Fencing is one of the sports disciplines in which the correct technique for performing actions is very important. For any practical application, temporal segmentation of movement in continuous training is crucial. In this work, we consider detecting and classifying actions in a sequence of fencing footwork exercises. We apply pose estimation to RGB videos and then we perform per-frame motion classification, using both classical machine learning and deep learning methods. Using sequences of frames with the same class we find data segments with specific actions. For evaluation, we provide extended manual labels for a fencing footwork dataset previously used in other works. Results indicate that the proposed methods are effective at detecting four footwork actions, obtaining 0.98 F1 score for recognition of action segments and 0.92 F1 score for per-frame classification. In the evaluation of our approach, we provide also a comparison with other data modalities, including depth-based pose estimation and inertial signals. Finally, we include an example of qualitative analysis of the performance of detected actions, to show how this approach can be used for training support.

Keywords

Temporal segmentation, action recognition, sports analysis, fencing, pose estimation, motion analysis.

1 INTRODUCTION

Due to recent advances, human action recognition has found applications in areas such as human-computer interaction, assisted living systems, rehabilitation support, entertainment, surveillance, and sports analysis [KF22, BNSH20]. Supporting sports training with the information provided by various devices becomes more and more popular, not only in professional but also in amateur sports. In highly technical sports disciplines, such as fencing, it is crucial to get proper feedback on exercises in order to improve the performance of different actions. While this task is typically realized by a coach, it is possible to automatically measure several motion parameters during training and provide this information to the person performing the actions. Temporal segmentation is a crucial element of motion analysis,

as it enables the automatic detection of actions that can then be evaluated.

In this work, we consider temporal segmentation of actions in fencing footwork, in which a number of relevant motion parameters can be measured. We detect and classify four relevant actions in recordings of continuous training. Our goal is to obtain a solution that provides useful feedback based on RGB video data. We employ a pose-based action detection, therefore, variations in environment conditions are handled by a state-of-the-art RGB-based pose estimation algorithm, and our models need only to focus on the patterns of motion in actions. This enables us to train the models on a relatively small dataset. Since other modalities are also commonly used for similar tasks, we compare our methods on depth-based pose estimation and inertial data as well. For evaluation we obtained extended expert manual labeling for a dataset used in previous works. We also show how pose estimation and action detection can be used to obtain specific action performance parameters. In this work we: provide expert, multi-class labeling for fencing footwork dataset, compare classical and deep learning approaches for temporal segmentation using multiple modalities, propose a proof-of-concept action performance analysis.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2 RELATED WORK

Sports support is a very promising application of automatic human motion analysis. A variety of methods, with several data modalities, have been proposed in the literature to provide automated detection, classification and evaluation of actions in a variety of sports disciplines [WWB⁺22, PZW⁺22].

2.1 Data modalities

Analysis of actions in sports can be performed with multiple different modalities [SKR⁺23]. An obvious and popular approach is to employ RGB videos, as those are easy to acquire. Moreover, automatic analysis based on videos mimics typical workflow of a human coach. Convolutional neural networks are currently the most common approach for spatio-temporal action recognition when using RGB videos directly [CPR⁺21, LLZ⁺20]. Inertial measurement units (IMU) are small devices mounted on a person, that can measure acceleration, angular velocity and magnetic field, as well as estimate orientation based on data fusion. IMUs are widely used in action recognition, particularly in sports, as they do not suffer from occlusions, which is a relevant problem in vision based approaches. IMUs are employed, among others, for analysis of swimming [WPTM18] and combat sports [WEST19]. Another popular approach is to employ so-called skeleton data modality - an estimation of human pose, provided as coordinates of the most relevant joints. A large number of methods took advantage of skeleton data provided by the Kinect depth sensor [ZLO⁺16, RLDL20]. Recently, pose estimation from RGB videos has become increasingly popular and effective [BJ21, BGR⁺20], allowing to obtain reliable skeleton data with a typical smartphone, without the need to use a dedicated depth sensor [MJ22].

2.2 Action recognition in sports

Depending on the considered type of sports, different problems are relevant for extracting meaningful information from sports actions recordings. In team sports spatio-temporal event detection is of particular interest [YLH19], as well as tracking of players [FSY⁺20] and ball [YHC⁺19]. In analysis of individual sports the focus is more on detection, classification and evaluation of specific actions [HIK22]. Those, however, vary greatly between disciplines, therefore automatic analysis methods are often difficult to generalize. Classification of manually segmented fragments of signals including a single action was applied, among others, in tennis [SHU⁺22]. Automatic, temporal segmentation of actions is usually more difficult, but necessary in real-world applications. While in some sports it is sufficient to detect subsequent repetitions of the same action, e.g. in swimming [ZXZ⁺17], in other disciplines

a variety of actions, that can occur in almost any order, must be considered for effective analysis. Fencing is one of such disciplines, as combining different techniques in rapid and unpredictable manners is an important part of tactics.

Fencing was previously analyzed in terms of footwork classification [MK18, ZWM22], bladework classification [MRPL10] and also kinematics analysis of motion [GTF08]. In this work, we consider analysis of continuous fencing footwork training. This problem was previously addressed in [Mal20], where a single action (lunge) was detected using rule-based model. In this work, using the same dataset, we extend manual labeling of data to include total of four actions (step forward, step backward, lunge, return from lunge). Next, we propose and evaluate action detection methods based on both classical machine learning and deep learning methods. Finally, we show how the proposed approach can be used to provide useful feedback to fencers.

3 FENCING FOOTWORK

Fencing training includes two main elements - footwork and bladework. Those are practiced separately in specific exercises and then jointly in combined exercises. In this work, we consider only the footwork. The main actions in footwork are steps forward and backward, as well as fencing lunge and return from the lunge. Fencers move in a sideways position (see Fig. 1 left), with the blade always pointed towards the opponent, therefore we can distinguish the front and the back leg. In fencing steps (see Fig. 1 middle), it is important to maintain proper distance between both legs, as well as correct knee bend. Fencing lunge (see Fig. 1 right) is a dynamic forward motion used during offensive actions. Proper lunge action is initiated by lifting the front foot toes, then thrusting the front leg, straightening the knee and finally landing, with knee angle in resting position at least 90 degrees. Proper return to basic fencing pose depends on not relaxing legs muscles between the lunge and the return. It is worth noting, that steps, lunge, and return have some variations, e.g. including small jump motion. In all actions, time and range of performance are also important. Tracking those parameters of performing fencing footwork exercises provides relevant feedback to a fencer, which can aid them to progress faster. Automating this process requires temporal segmentation of continuous training, as well as estimation of specific motion parameters.

4 METHODS

In this section we describe employed data and labeling process, pose estimation, temporal segmentation and performance evaluation.

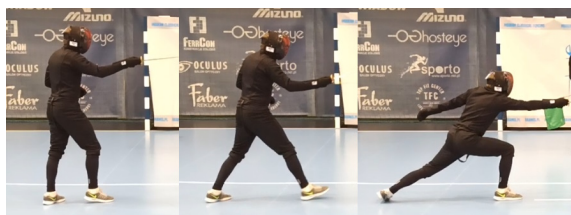


Figure 1: Fencing base pose (left), step forward (middle) and lunge (right).

4.1 Data labeling

We employ a dataset previously used in [Mal20]. It contains recordings of continuous fencing footwork training, acquired with the Kinect sensor and two IMUs, mounted on the chest and on the elbow of the front arm. The Kinect data includes RGB video, depth video, and skeleton estimation, while the two IMUs provide acceleration, angular velocity, and magnetic field. All data are synchronized with a common sampling frequency 30Hz. Aforementioned previous work compared the detection of a single action (fencing lunge) using skeleton data estimated from depth maps and acceleration from IMUs. In this work we provide additional manual labeling of all recordings in the dataset, to include four actions: step forward, step backward, lunge, and return. Manual segmentation is performed by an expert fencer (15 years of experience). A custom tool was developed for labeling, which provided user interface for frame-by-frame viewing of the video and selecting type, start frame and end frame of each action. It is worth noting, that the expert's opinion is that the exact start and end points of actions are sometimes unclear, as the actions may partly overlap or some additional movement between actions may be present.

4.2 Pose estimation

Our goal is to obtain reliable action detection based solely on RGB videos. We use RGB pose estimation as an intermediate representation of motion, therefore variability of environment, lighting, and poses is already captured in the pose estimation model, and our models can focus on the performed actions. This allows us to obtain effective action recognition even though the dataset is relatively small (28 recordings lasting approx. 30 seconds each). We employ BlazePose model included in MediaPipe library [BGR⁺20]. While our main focus is on action detection from RGB videos (using pose estimation as intermediate representation), for comparison, we evaluate our methods on the depth-based skeleton and inertial modalities as well.

Detection of actions in recorded video signal starts with running the BlazePose algorithm from MediaPipe library (see Fig. 2). It provides estimation for 33 landmarks, including 11 face keypoints and 22 most relevant joints (shoulders, elbows, wrists, thumbs, pinky



Figure 2: Fencing lunge - pose estimation (red dots indicate detected joints)

and index fingers, hips, knees, ankles, heels, feet index). Pose is estimated in each frame, using the fastest out of three MediaPipe models, with tracking mode enabled (detection from previous frames is used). While the other two models are larger and therefore more effective, we found that there is little practical difference in accuracy of the models, while the difference in speed is significant. The fastest model is able to run at 15 frames per second on a mid-range smartphone, which enables real-time tracking. It is worth noting, that while BlazePose is able to provide 3D estimation of joint positions, we employ only x and y coordinates, as motion along the z-axis (depth) is not relevant in this scenario.

4.3 Temporal segmentation

We perform temporal detection of actions by classifying each frame based on its context (neighboring frames). We investigate two main approaches: classical machine learning (CML) and deep learning (DL). In the CML method, we extract features in frequency domain and then we apply dimensionality reduction and classification, which is an approach proved to be effective in various motion analysis scenarios [MK18, HJ09]. We first compute per-frame x and y velocities of each joint, using the difference of their positions in neighbouring frames. Then, for each frame, we compute the discrete cosine transform (DCT) in a time window centered on this frame. The length of the window is an important hyperparameter to be selected in the experiments, as it defines the context. From the obtained DCT coefficients we remove the first one, as it corresponds to the constant component and therefore may introduce unwanted bias. DCT coefficients for x and y axes are concatenated and then principal components analysis (PCA) is applied in order to remove redundant information and reduce the number of features. Finally, using obtained feature vectors, we train the support vector machine (SVM) classifier.

In the DL approach, we consider three types of neural networks: long short-term memory (LSTM), gated recurrent units (GRU), and 1-dimensional convolutional neural networks (CNN1D). Those architectures proved to be efficient for human action recognition in different applications [LWW⁺17, MJ22]. The first recurrent

neural network (LSTM) has two bidirectional LSTM layers with 128 units each, followed by two dense layers with 128 and 64 units. The second recurrent network (GRU), has similar architecture, with LSTM layers replaced with GRU layers. CNN1D has three 1-dimensional convolution layers with kernel size 3 and units number set to 32, 64, and 128 respectively. Between convolutions, there are max pooling operations (size 2), and at the end, global average pooling is applied, followed by a dense layer with 128 units. In all networks, there is a final layer with a size equal to the number of classes. In the case of the DL approach the input signal is also a time window of selected length, but rather than computing DCT, we use directly the sequence of joint coordinates, although filtered with Butterworth's filter, with cutoff frequency = 2Hz.

Per frame detection allows to relatively easily find action segments. One common problem that needs to be addressed when merging single frames into segments is the misclassification of single frames or even short sequences of frames during the action. In order to handle such situations, we apply postprocessing, in which short segments of frames with the same class are reclassified if they occur between two segments of another class (which becomes their new class). The maximum length of reclassified segments is set to 10, which corresponds to 330 ms. We found that such length is sufficient to remove such occurrences, while also ensuring that in this time range, there is no actual action of another class. It is also worth noting that in this work we do not address the problem of segmenting subsequent instances of the same action, e.g. multiple steps backward will be treated as a single segment of this class.

As mentioned previously, for comparison we consider also Kinect skeleton modality and IMU data. Since Kinect provides similar data as the BlazePose estimation (only with a smaller number of landmarks), the methods remain the same. IMUs provide a 3-axis measurement of acceleration, angular velocity, and magnetic field. While the nature of these data differs greatly from pose modality, these are also time signals, which can be processed in the same manner, therefore we apply the same approaches. Please note, that the methods were not optimized for the different modalities.

4.4 Performance evaluation

While the main goal of this work was temporal segmentation of actions in fencing footwork, we also include proof-of-concept methods for evaluation of performance, to provide feedback regarding the most common mistakes. We consider two motion parameters:

- Ratio of minimal feet distance to the shoulder distance during step forward and step backward actions
- Maximum angle of the front knee during the lunge action

Regarding the first motion parameter, fencing coaches recommend, that for effective moving, in forward and backward steps, fencers should keep the distance between feet similar to the distance between the shoulders. A common mistake is to have the feet too close to each other after finishing a step. Therefore, we measure minimal distance of feet in steps and compare it to the shoulder distance. Regarding the second motion parameter, the coaches state that the front leg should be fully straightened during the lunge to obtain optimal range and dynamics. Therefore, we measure the maximum knee angle in this action. Both parameters are measured using joint positions from pose estimation.

5 EXPERIMENTS

For experimental evaluation we employ dataset from [Mal20] with additionally added manual labels to include a total of 4 actions: step forward, step backward, lunge, and return, see Sec. 2.1. The dataset was acquired with 9 fencers, and for each fencer, there are 3 or 4 recordings - sequences of continuous fencing footwork training. In all experiments we employ leave-one-subject-out cross-validation, resulting in 9 folds, and the presented results are averaged from all folds. Parameters for feature extraction and classification were determined in a grid search. For the SVM approach, we used window size = 20, number of selected PCA components = 300, and regularization parameter $C = 1$. Neural networks (LSTM/GRU/CNN1D) were trained, respectively, on data with window size = 20/20/15 using Adam optimizer, with learning rate 0.001/0.0005/0.001 and batch size = 32/128/32, for 8/20/20 epochs. It is worth noting that the window size had the most impact on the results.

In the experiments we consider two scenarios: 1) detection of lunge action only, in order to compare with the previous method, and 2) detection of all four actions. For all experiments we measure precision, recall and F1 score. Precision is the ratio of correctly classified frames or actions of given class to all frames or actions classified as this action. Recall is the ratio of correctly classified frames or actions of given class to all actual frames or actions of this class. F1 score is a harmonic mean of precision and recall, which makes it a well balanced metric. Finally, we also present results for the evaluation of performance based on selected parameters.

5.1 Lunge detection

First, we investigate the effectiveness of detecting only the lunge action in order to compare proposed automatic approaches with the previous, rule-based method described in [Mal20]. We present both per-action and per-frame classification results. An action is considered to be detected correctly if the middle frame of the detected segment lies between the start and end frames of

the ground truth segment. We also provide results for finding the first and the last frame of action. While in some actions exact start and end points are not that important, in lunge action start point needs to be detected accurately in order to evaluate correctness in terms of relative motion of body and hand with the weapon. Finally, we depict an example of detection in a plot including ground truth and detected segments.

Modality	Method	F1	Prec.	Recall
RGB pose	SVM	1.00	1.00	1.00
	LSTM	0.99	1.00	0.99
	GRU	1.00	1.00	1.00
	CNN1D	0.97	0.99	0.96
Kinect pose	SVM	0.99	1.00	0.98
	LSTM	1.00	1.00	0.99
	GRU	0.99	1.00	0.98
	CNN1D	0.99	0.99	0.99
	Rules	1.00	1.00	1.00
IMU	SVM	0.94	0.96	0.93
	LSTM	0.91	0.95	0.88
	GRU	0.92	0.97	0.87
	CNN1D	0.83	0.94	0.74
	Rules	0.99	0.99	0.99

Table 1: Single class (lunge) per-action classification results. Results for rule-based method included from previous work [Mal20].

Modality	Method	F1	Prec.	Recall
RGB pose	SVM	0.94	0.96	0.93
	LSTM	0.90	0.94	0.88
	GRU	0.92	0.94	0.92
	CNN1D	0.90	0.94	0.88
Kinect pose	SVM	0.90	0.94	0.87
	LSTM	0.93	0.94	0.92
	GRU	0.91	0.93	0.90
	CNN1D	0.90	0.93	0.88
IMU	SVM	0.82	0.85	0.81
	LSTM	0.80	0.84	0.78
	GRU	0.80	0.86	0.77
	CNN1D	0.73	0.83	0.71

Table 2: Single class (lunge) per-frame classification results.

Our analysis of the results starts with per-action detection, as presented in Table 1. The referenced rule-based method obtained perfect detection of lunge actions using the Kinect pose estimation. The proposed method allowed us to obtain the same result using pose estimation from RGB data and either an SVM classifier or GRU neural network. Other methods also obtain very high results using both RGB and Kinect pose estimations. Interestingly, for the IMU data, learning methods are less effective than the rule-based approach. More specific features may be needed for this modality. Per-frame results (see Table 2) also indicate that RGB pose

Modality	Method	Start err.	End err.
RGB pose	SVM	1.64 ± 2.22	0.99 ± 1.52
	LSTM	2.11 ± 1.32	0.99 ± 0.57
	GRU	1.51 ± 0.75	0.77 ± 0.44
	CNN1D	2.08 ± 2.84	0.92 ± 0.63
Kinect pose	SVM	1.86 ± 1.06	1.36 ± 0.71
	LSTM	1.42 ± 0.74	1.08 ± 0.48
	GRU	1.69 ± 0.81	1.39 ± 1.13
	CNN1D	1.69 ± 1.07	1.15 ± 0.43
	Rules	1.23 ± 1.17	0.66 ± 0.65
IMU	SVM	2.95 ± 1.69	3.51 ± 3.38
	LSTM	3.48 ± 2.63	3.39 ± 3.17
	GRU	3.18 ± 2.28	2.67 ± 1.70
	CNN1D	6.58 ± 9.17	4.97 ± 9.66
	Rules	2.57 ± 1.58	2.49 ± 1.70

Table 3: Single class (lunge) start and end frame detection error given in frames, with mean and standard deviation. Results for rule-based method included from previous work [Mal20].

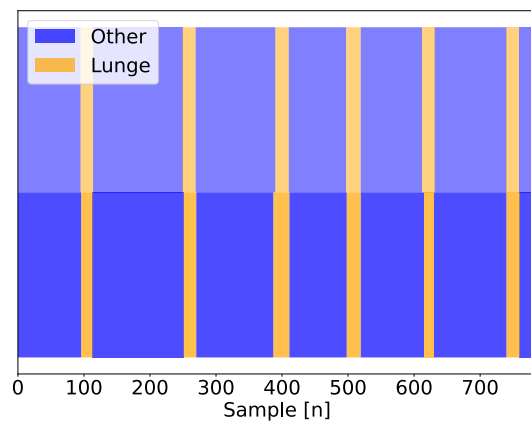


Figure 3: Example of detection of lunge action in continuous recording. Action segments are color-coded. Bottom half represents ground truth, while upper half represents detection results.

estimation combined with SVM or GRU is the most effective approach. In terms of finding exact start and end points (see Table 3), RGB pose estimation with GRU is the most accurate of the proposed methods, while still slightly less effective than the rule-based method. Finally, we can also observe proper detection of the lunge action in the plot in Fig. 3.

5.2 Multi-class detection

One of the key limitations of the previous rule-based method is that it does not generalize well to other actions. Defining manual rules for multiple actions is time-consuming and prone to errors. Therefore we investigate learning approaches for temporal segmentation of four actions using the extended manual labeling provided by an expert fencer.

Modality	Method	F1	Prec.	Recall
RGB pose	SVM	0.98	0.97	0.99
	LSTM	0.92	0.88	0.97
	GRU	0.96	0.94	0.98
	CNN1D	0.94	0.90	0.98
Kinect pose	SVM	0.92	0.88	0.97
	LSTM	0.97	0.95	0.99
	GRU	0.96	0.94	0.98
	CNN1D	0.94	0.91	0.97
IMU	SVM	0.87	0.82	0.93
	LSTM	0.88	0.82	0.94
	GRU	0.87	0.81	0.93
	CNN1D	0.79	0.72	0.89

Table 4: Multi class per-action classification results.

Modality	Method	F1	Prec.	Recall
RGB pose	SVM	0.92	0.92	0.92
	LSTM	0.87	0.87	0.87
	GRU	0.90	0.90	0.90
	CNN1D	0.88	0.88	0.88
Kinect pose	SVM	0.87	0.87	0.87
	LSTM	0.89	0.89	0.89
	GRU	0.89	0.89	0.89
	CNN1D	0.88	0.88	0.88
IMU	SVM	0.75	0.75	0.75
	LSTM	0.75	0.75	0.75
	GRU	0.72	0.72	0.72
	CNN1D	0.66	0.66	0.66

Table 5: Multi class per-frame classification results.

Results in Table 4 indicate that the most effective approach for detection of multiple actions is the SVM classifier applied to pose estimation from RGB video, as it obtained F1 score = 0.98, precision = 0.97 and recall = 0.99. GRU network is a close second for this modality with F1 score = 0.96, precision = 0.94, and recall = 0.98. IMU data provides significantly less accurate detection, with the best F1 score = 0.88 obtained with LSTM neural network. As mentioned before, the

Modality	Method	Start err.	End err.
RGB pose	SVM	1.47 ± 1.15	1.48 ± 1.24
	LSTM	2.11 ± 0.96	2.33 ± 1.28
	GRU	1.74 ± 0.98	1.72 ± 0.79
	CNN1D	1.99 ± 0.95	1.94 ± 1.06
Kinect pose	SVM	2.36 ± 0.78	2.37 ± 0.63
	LSTM	1.77 ± 0.66	1.73 ± 0.67
	GRU	1.87 ± 0.85	2.05 ± 1.18
	CNN1D	1.93 ± 0.78	1.83 ± 0.67
IMU	SVM	3.96 ± 2.83	4.32 ± 3.61
	LSTM	4.34 ± 1.25	4.74 ± 1.72
	GRU	4.12 ± 1.07	4.76 ± 1.29
	CNN1D	4.44 ± 2.61	4.51 ± 2.72

Table 6: Multi class start and end frame detection error given in frames (including mean and standard deviation).

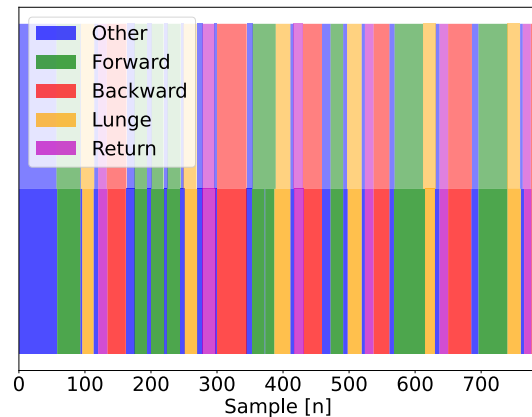


Figure 4: Example of detection of four actions in continuous recording. Action segments are color-coded. Bottom half represents ground truth, while upper half represents detection results.

proposed methods were not optimized for this modality, which may be the reason for lower effectiveness. Results for per-frame detection (see Table 5) indicate the same methods to be the most effective, but also show that for this application RGB pose estimation provides more relevant information than Kinect skeleton data. Error in detecting the start frame (see Table 6) is lower than in the case of a single action, however, end frame error is higher. Both start and end frame errors correspond to approx. 50 ms, which is sufficient for performance analysis. For a visual representation of multi-class temporal segmentation see the plot depicted in Fig. 4.

5.3 Action performance evaluation

While the main goal of this work was to perform temporal segmentation of actions in fencing footwork, we also include a limited action performance analysis in order to show the potential of the final application of the proposed methods. In the detected step forward actions we measure the ratio of the minimum distance of feet to the distance of shoulders. Fig. 5 presents an example of correct (left) and incorrect (right) poses in terms of feet distance. The ratio parameter for the depicted correct pose is 0.96, while for the incorrect pose, it is 0.84. Recommendation from a fencing coach is that the ratio should be close to 1. In Fig. 6 correct lunge is the one with a straight front leg (left image), while the incorrect is the one with a bent knee (right image). The knee angle computed using pose estimation is 177 degrees and 156 degrees respectively. Expected angle for a correct action is approx. 180 degrees (straight leg). As we can see, by using detected actions and dependencies between joints in pose estimation, we can find occurrences of incorrectly performed actions and therefore provide useful feedback to the fencer.

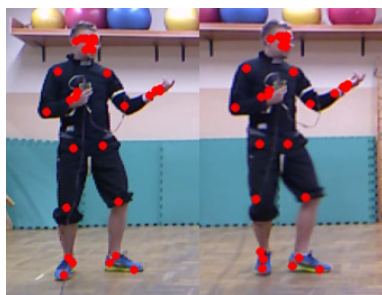


Figure 5: Example of performance analysis in step action. Correct distance between feet (right) vs incorrect (left).

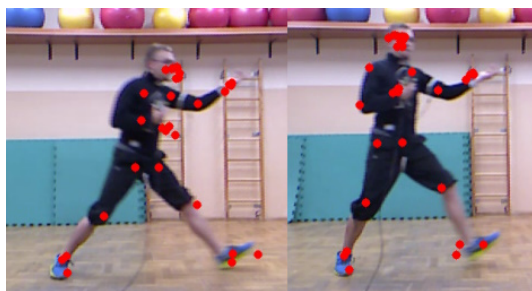


Figure 6: Example of performance analysis in lunge action. Correct knee angle (right) vs incorrect (left).

6 CONCLUSIONS

In this work, we proposed methods for support of fencing footwork exercises. Our approach employs pose estimation from RGB videos, which does not require any devices other than a smartphone, as opposed to previously proposed methods which relied on depth cameras and inertial sensors. This facilitates introducing proposed solution to fencing trainings. We evaluated classical and deep learning methods for the task. Both approaches yielded similar results, with SVM performing slightly better than best neural network architecture (GRU). We expect, that deep learning approaches would be more effective with a larger dataset. Overall, obtained results are very good. Considering best obtained multi-class action classification F1 score = 0.98, proposed method could be used in practical application. Detection of start and end frames, relevant for some actions, is also accurate - average error 1.47 frame and 1.48 frame respectively, which corresponds roughly to 50ms. Also, the proof-of-concept action performance analysis produced promising results, even though it requires more thorough evaluation, for which additional, specific data is needed.

Future works can be realized in multiple directions. First of all, additional, less common footwork actions can be added, such as dodging. Secondly, additional segmentation of sequences of the same actions (e.g. multiple steps forward or backward) can be considered. Moreover, automatic analysis of bladework would be beneficial for the fencers as well, even though it may

prove more difficult to realize. Fusion of visual and inertial data may be useful in this regard. Finally, more qualitative motion parameters can be extracted for the analyzed actions, therefore providing the fencers with additional relevant feedback. However, evaluation of qualitative analysis will require recording additional data with actions performed correctly and incorrectly. Is it also worth noting, that the proposed methods could be used for real-time analysis, which may be used to deliver feedback while training, rather than only when viewing a recording. Such feedback could be delivered e.g. by generating sounds, visual signals or even spoken comments.

7 ACKNOWLEDGMENTS

The research presented in this paper was supported by the National Centre for Research and Development (NCBiR) under Grant No. LIDER/37/0198/L-12/20/NCBR/2021. We also thank Aramis Fencing School (aramis.pl) for providing experts' consultations.

8 REFERENCES

- [BGR⁺20] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. BlazePose: On-device real-time body pose tracking. *arXiv preprint arXiv:2006.10204*, 2020.
- [BJ21] Rishabh Bajpai and Deepak Joshi. Movenet: A deep neural network for joint profile prediction across variable walking speeds and slopes. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2021.
- [BNSH20] Djamila Romaissa Beddiar, Brahim Nini, Mohammad Sabokrou, and Abdenour Hadid. Vision-based human activity recognition: a survey. *Multimedia Tools and Applications*, 79:30509–30555, 2020.
- [CPR⁺21] Chun-Fu Richard Chen, Rameswar Panda, Kandan Ramakrishnan, Rogerio Feris, John Cohn, Aude Oliva, and Quanfu Fan. Deep analysis of cnn-based spatio-temporal representations for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6165–6175, 2021.
- [FSY⁺20] Na Feng, Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, Yizhu Zhao, Yunfeng He, and Tao Guan. Sset: a dataset for shot segmentation, event detection, player tracking in soccer videos. *Multimedia Tools and Applications*, 79:28971–28992, 2020.
- [GTF08] M Gholipour, A Tabrizi, and F Farahmand. Kinematics analysis of lunge fencing using stereophotogrammetry. *World Journal of Sport Sciences*, 1(1):32–37, 2008.
- [HIK22] Kristina Host and Marina Ivašić-Kos. An overview of human action recognition in

- sports based on computer vision. *Heliyon*, 8(6):e09633, 2022.
- [HJ09] Zhenyu He and Lianwen Jin. Activity recognition from acceleration data based on discrete cosine transform and svm. In *2009 IEEE international conference on systems, man and cybernetics*, pages 5041–5044. IEEE, 2009.
- [KF22] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *International Journal of Computer Vision*, 130(5):1366–1401, 2022.
- [LLZ⁺20] Jun Li, Xianglong Liu, Wenxuan Zhang, Mingyuan Zhang, Jingkuan Song, and Nicu Sebe. Spatio-temporal attention networks for action recognition and detection. *IEEE Transactions on Multimedia*, 22(11):2990–3001, 2020.
- [LWW⁺17] Chuankun Li, Pichao Wang, Shuang Wang, Yonghong Hou, and Wanqing Li. Skeleton-based action recognition using lstm and cnn. In *2017 IEEE International conference on multimedia & expo workshops (ICMEW)*, pages 585–590. IEEE, 2017.
- [Mal20] Filip Malawski. Depth versus inertial sensors in real-time sports analysis: A case study on fencing. *IEEE sensors journal*, 21(4):5133–5142, 2020.
- [MJ22] Filip Malawski and Bartosz Jankowski. Depth-based vs. color-based pose estimation in human action recognition. In *Advances in Visual Computing: 17th International Symposium, ISVC 2022, San Diego, CA, USA, October 3–5, 2022, Proceedings, Part I*, pages 336–346. Springer, 2022.
- [MK18] Filip Malawski and Bogdan Kwolek. Recognition of action dynamics in fencing using multimodal cues. *Image and Vision Computing*, 75:1–10, 2018.
- [MRPL10] G Mantovani, A Ravaschio, P Piaggi, and A Landi. Fine classification of complex motion pattern in fencing. *Procedia Engineering*, 2(2):3423–3428, 2010.
- [PZW⁺22] Yiqun Pang, Changnian Zhang, Yibing Wang, Qiurui Wang, and Mingyang Wang. Analysis of computer vision applied in martial arts. In *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 191–196. IEEE, 2022.
- [RLDL20] Bin Ren, Mengyuan Liu, Runwei Ding, and Hong Liu. A survey on 3d skeleton-based action recognition using learning method. *arXiv preprint arXiv:2002.05907*, 2020.
- [SHU⁺22] Anik Sen, Syed Md Minhaz Hossain, Russo-MohammadAshraf Uddin, Kaushik Deb, and Kang-Hyun Jo. Sequence recognition of indoor tennis actions using transfer learning and long short-term memory. In *Frontiers of Computer Vision: 28th International Workshop, IW-FCV 2022, Hiroshima, Japan, February 21–22, 2022, Revised Selected Papers*, pages 312–324. Springer, 2022.
- [SKR⁺23] Zehua Sun, QiuHong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. Human action recognition from various data modalities: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3200–3225, 2023.
- [WEST19] Matthew TO Worsey, Hugo G Espinosa, Jonathan B Shepherd, and David V Thiel. Inertial sensors for performance analysis in combat sports: A systematic review. *Sports*, 7(1):28, 2019.
- [WPTM18] Matthew TO Worsey, Rebecca Pahl, David V Thiel, and Peter D Milburn. A comparison of computational methods to determine intrastroke velocity in swimming using imus. *IEEE Sensors Letters*, 2(1):1–4, 2018.
- [WWB⁺22] Fei Wu, Qingzhong Wang, Jiang Bian, Ning Ding, Feixiang Lu, Jun Cheng, Dejing Dou, and Haoyi Xiong. A survey on video action recognition in sports: Datasets, methods and applications. *IEEE Transactions on Multimedia*, pages 1–25, 2022.
- [YHC⁺19] Young Yoon, Heesu Hwang, Yongjun Choi, Minbeom Joo, Hyeyoon Oh, Insun Park, Keon-Hee Lee, and Jin-Ha Hwang. Analyzing basketball movements and pass relationships using realtime object tracking techniques based on deep learning. *IEEE Access*, 7:56564–56576, 2019.
- [YLH19] Junqing Yu, Aiping Lei, and Yangliu Hu. Soccer video event detection based on deep learning. In *MultiMedia Modeling: 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8–11, 2019, Proceedings, Part II 25*, pages 377–389. Springer, 2019.
- [ZLO⁺16] Jing Zhang, Wanqing Li, Philip O Ogunbona, Pichao Wang, and Chang Tang. Rgb-d-based action recognition datasets: A survey. *Pattern Recognition*, 60:86–105, 2016.
- [ZWM22] Kevin Zhu, Alexander Wong, and John McPhee. Fencenet: Fine-grained footwork recognition in fencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3589–3598, 2022.
- [ZXZ⁺17] Zhendong Zhang, Dongfang Xu, Zhihao Zhou, Jingeng Mai, Zhongkai He, and Qining Wang. Imu-based underwater sensing system for swimming stroke classification and motion analysis. In *2017 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pages 268–272. IEEE, 2017.

Accuracy of Legendre Moments for Image Representation

Cesar Bustacara-Medina

Department of Systems Engineering
Pontificia Universidad Javeriana
Bogotá D.C., Colombia
cbustaca@javeriana.edu.co

Enrique Ruiz-García

Department of Systems Engineering
Pontificia Universidad Javeriana
Bogotá D.C., Colombia
eruiz@javeriana.edu.co

ABSTRACT

Existing works on orthogonal moments are mainly focused on optimizing classical orthogonal Cartesian moments, such as Legendre moments, Gauss-Hermite moments, Gegenbauer moments, and Chebyshev moments. Research in this area generally includes accurate calculation, fast computation, robustness/invariance optimization, and the application of orthogonal moments. This paper presents the inclusion of the integration method proposed by Holoborodko to calculate the Legendre moments. The results obtained are compared with the traditional equation and the methods proposed by Hosny and Pawlak to approximate the integration computation.

Keywords

Image Processing, Image Reconstruction, Legendre Moments, Orthogonal Moments, Moments Computation.

1 INTRODUCTION

Image representation is a principal research area in image processing, pattern recognition, and robotics. In general, there are three types of image representation: the first is developed to support special devices; the second is for the compression of images; and the third is developed to support special image operations [1, 2, 3]. Orthogonal moments, such as Legendre moments and Zernike moments, were first introduced in image processing by Teague [4] and they have been widely used in image analysis and pattern recognition [5, 6] due to their near-zero information redundancy and high discriminative power. Moment-based image representation has been reported to be effective in satisfying the core conditions of semantic description due to its beneficial mathematical properties, especially geometric invariance and independence [1]. For example, moment functions of image intensity values have been successfully used in object recognition [5, 7, 8, 9, 10], image analysis [4, 11, 12, 13, 14], object representation [15], edge detection [16, 17, 18], and texture analysis [19].

Moments and invariant moments were introduced to the pattern recognition community in 1962 by Hu [10]. Since then, after almost 60 years of research, numerous moment-based techniques have been developed for image representation with varying success

degrees. For example, researchers have been introduced Invariant Moments, Rotational Moments, Orthogonal Moments, Complex Moments, and Standard Moments [1, 20]. In 1998, Mukundan and Ramakrishnan [21] surveyed the main publications proposed until then and summarized the theoretical aspects of several classical moment functions. In 2006, Pawlak [12] gave a comprehensive survey on the reconstruction and calculation aspects of the moments with great emphasis to the accuracy/error analysis. In 2007, Shu et al. [22, 23, 24] provided a brief literature review for the mathematical definitions, invariants, and fast/accurate calculations of the classical moments, respectively. In 2009, Flusser et al. [25] presented an overview of moment-based pattern recognition methods with significant contribution to the theory of invariant moments. The substantial expansion [26] of this book includes more detailed analysis of the 3D object invariant representation. In 2011, Hoang [27] reviewed unit disk-based orthogonal moments in his doctoral dissertation, covering theoretical analysis, mathematical properties, and specific implementation. For most of the above reviews, state-of-the-art methods in the past 10 years are not covered. In 2014, Papakostas et al. [28] gave a global overview of the milestones in the 50 years research and highlighted all recent rising topics in this field. However, the theoretical basis for these latest research directions is rarely introduced. In 2019, Kaur et al. [29] provided a comparative review for many classical and new moments. More recently, in 2023, Qi et al. [1] provided a comprehensive survey of the orthogonal moments for image representation, covering recent advances in fast/accurate calculation, robustness/invariance optimization, definition extension, and their application.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Orthogonal moments are shown to be less sensitive to noise and have an efficient capability to feature representation. They allow to reconstruct the image intensity function analytically, from a finite set of moments, using the inverse moment transform [30]. Legendre and Zernike moments are most widely used because their minimum redundancy. Indeed, they can represent the properties of an image with no redundancy or overlap information between the moments. Unfortunately, these approaches are often characterized by a huge computational complexity, making them unsuitable for real-time applications. This lead many researchers to develop faster and accurate algorithms for computing Legendre and Zernike moments [31]. Specifically, the computation of Legendre moments is in general a time consuming process. In many references [32, 33, 34], their computation has been performed using closed form representations for orthogonal polynomials, and taking little care to the accuracy of the quadrature formulas used to approximate integrals.

In general, the calculation of orthogonal moments suffers from geometric error, numerical integration error, and representation error (mainly numerical instability). These errors will severely restrict the quality of image representation, especially when high-order moments are required to better image description. According to Qi et al. [1], the accurate computation strategies of moments are vital for their applicability.

The rest of the paper is organized as follows: In Section 2, an overview of Legendre moments is given. The accuracy computation of Legendre Moments is described in Section 3. Section 4 presents an experimental results of the different methods used to compute Legendre Moments, including the Holoborodko method that is proposed in this paper. Conclusions are presented in Section 5.

2 LEGENDRE MOMENTS

Legendre moments suggested by Teague [4] are one of the important continuous orthogonal moments defined in a rectangular region and have been well researched since the early years of moment-based descriptor studies [4, 14, 35, 36, 37]. In general, Legendre moments form an orthogonal set, defined in Cartesian coordinate space [21, 38] and have been used to analyze and extract features, for example, in facial recognition, image indexing, pattern recognition, etc.

Legendre moments usually contain Legendre polynomials as the kernel which approximated by sampling at fixed intervals, so, the resulted moments have approximated values. In Addition, Legendre moments are orthogonal and scale invariants hence they are suitable for representing the features of the images [39]. In this case, the image intensity distribution can be analytically reconstructed from its orthogonal moments.

2.1 Definitions and Properties

The kernel of Legendre moments are products of Legendre polynomials defined along rectangular image coordinate axes inside a unit circle [3, 4, 21, 38, 40]. The Legendre moments of order $(p + q)$ are defined as

$$L_{pq} = \frac{(2p+1)(2q+1)}{4} \int_{-1}^1 \int_{-1}^1 P_p(x) P_q(y) f(x,y) dx dy \quad (1)$$

where the functions $P_p(x)$ and $P_q(y)$ denote Legendre polynomial of order p and q , respectively. The Legendre moments L_{pq} generalizes the geometric moments m_{pq} , in the sense that the monomial $x^p y^q$ is replaced by the orthogonal polynomial $P_p(x) P_q(y)$ of the same order.

In order to evaluate the Legendre moments, the image coordinate space has to be necessarily scaled so that their respective magnitudes are less than 1. If the image dimension along each coordinate axis is N pixels, and i, j denote the pixel coordinate indices along the axes, then $0 \leq i, j \leq N$, and the discrete version of the Legendre moments can be written as

$$L_{pq} = \frac{(2p+1)(2q+1)}{(N-1)^2} \sum_{i=1}^N \sum_{j=1}^N P_p(x_i) P_q(y_j) f(i, j) \quad (2)$$

where x_i, y_j denote the normalized pixel coordinates in the range $[-1, 1]$, given by

$$x_i = \left(\frac{2i}{N}\right) - 1; \quad y_j = \left(\frac{2j}{N}\right) - 1 \quad (3)$$

The functions $P_p(x)$ form a complete orthogonal basis set inside the unit circle, and the function $f(i, j)$ can be approximated by a truncated series of Legendre moments as:

$$f(i, j) \cong \sum_p \sum_q L_{pq} P_p(x_i) P_q(y_j) \quad (4)$$

The above equation represents the inverse moment transform used for image reconstruction from a finite set of Legendre moments [3, 38].

Teague [4] derived a simple approximation to the inverse transform for a set of moments through order N . Additionally, Teh and Chin [14] indicated that, if only Legendre moments of order $\leq N$ are given, then the function $f(x, y)$ can be approximated by

$$f(x, y) \cong \sum_{p=0}^N \sum_{q=0}^p L_{p-q, q} P_{p-q}(x) P_q(y) \quad (5)$$

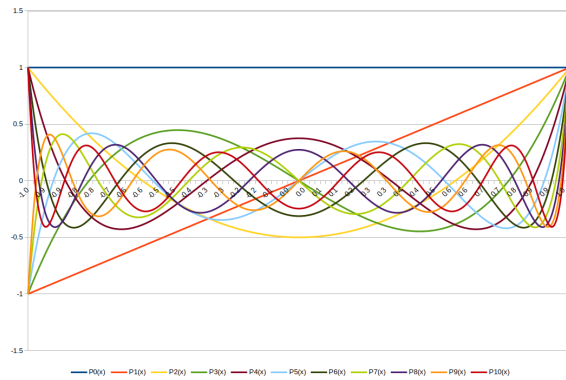


Figure 1: Legendre Polynomials

2.2 Legendre Polynomials

The Legendre polynomials $P_p(x)$ can be given by the equation [40, 41]:

$$P_p(x) = \frac{1}{2^p p!} \frac{d^p}{dx^p} (x^2 - 1)^p \quad (p \geq 0) \quad (6)$$

where $x \in [-1, 1]$, and the Legendre polynomial $P_p(x)$ obeys the following recursive relation [33, 42, 43, 44, 45, 46, 47, 48]:

$$P_{p+1}(x) = \frac{(2p+1)}{(p+1)} x P_p(x) - \frac{p}{(p+1)} P_{p-1}(x) \quad (7)$$

The Eq. (7) also can be rewritten by replacing p with $p-1$ as [2, 21, 35, 49]:

$$P_p(x) = \frac{(2p-1)}{p} x P_{p-1}(x) - \frac{(p-1)}{p} P_{p-2}(x) \quad (8)$$

with $P_0(x) = 1$, $P_1(x) = x$ and $p > 1$. The set of Legendre polynomials $\{P_p(x)\}$ forms a complete orthogonal basis set on the interval $[-1, 1]$.

The plots of the functions $P_p(x)$, with $p = 0, \dots, 10$, are given in Fig. (1).

2.3 Error Analysis

Only in the case of continuous moment functions, their computation over a discrete pixels space (image) is encounters some inaccuracies. These errors are of two types called geometric and numerical errors [1, 50]. The first type of error is caused by the projection of a square discrete image onto the domain (e.g. the unit disc for the radial polynomials) of the polynomial basis, while the numerical error is generated due to the calculation of the double integral over fixed sampling intervals. This case is presented by Papakostas [28] when zeroth-order approximation (ZOA) is applied.

Several approaches have been proposed in the literature towards the minimization of both error types. More

precisely, the geometric errors are minimized by applying specific mapping techniques from the image space to the polynomials domain and appropriate pixels arrangement methodologies [50]. The numerical integration errors are decreased by applying either analytical or approximate iterative integration algorithms (e.g. Simpson, Gauss) [50, 51, 52]. Currently, by using the aforementioned techniques, the values of the derived moments are very close to their theoretical values and, therefore, the level of accuracy achieved is satisfactory [28]. The following section presents the techniques that will be used to analyze the accuracy in the Legendre moments computation and, therefore, in the image reconstruction.

3 ACCURACY COMPUTATION OF LEGENDRE MOMENTS

Liao [36] indicated that the problem of the discretization error for moment computing has been barely investigated though some initial studies into this direction for the case of geometric moments were performed by Teh and Chin [14]. In addition Liao, presents a significant improvement on image reconstructions using Legendre and Zernike Moments. The numerical integration error is decreased by applying either analytical or approximate iterative integration algorithms [40, 50, 51, 52]. Currently, by using classic Newton-Cotes formulas such as Trapezoid, Simpson's, Extended Simpson's, Simpson's 3/8 and Boole's, the derived moment values are very close to their theoretical values and thus, the achieved accuracy level. However, the accuracy in the signals reconstruction (1D, 2D or 3D) is an important challenge today. For example, Table (1) shows the degree, formula, and error term of the classical Newton-Cotes techniques. Specifically, the accuracy degree is the largest positive integer that gives an exact value for x^k , for every k -value.

In addition, approximated Legendre moments defined by Eq. (1) are not accurate, where the double integration is replaced by double summation. Based on the basis of mathematical analysis, double summation is identical to the double integration only when the indices are reaching to infinity. In computing environment, this is not possible [51].

Generalizing, a digital image of size $M \times N$ is an array of pixels. Centers of these pixels are the points (x_i, y_j) , where the image intensity function is defined only for this discrete set of points $(x_i, y_j) \in [-1, 1] \times [-1, 1]$. Where $\Delta x_i = x_{i+1} - x_i$, and $\Delta y_j = y_{j+1} - y_j$ are sampling intervals in the x - and y -directions respectively. In the literature of digital image processing, the intervals Δx_i and Δy_j are fixed at constant values $\Delta x_i = 2/M$, and $\Delta y_j = 2/N$, respectively. Therefore, the points (x_i, y_j) will be defined as follows [40, 51]:

Integration Technique	Degree	Step Size (Δx)	Formula	Error Term
Trapezoid	1	$b-a$	$\frac{\Delta x}{2} (f_0 + f_1)$	$-\frac{1}{12} (\Delta x)^3 f^{(2)}(\xi)$
Simpson's	2	$\frac{b-a}{2}$	$\frac{\Delta x}{3} (f_0 + 4f_1 + f_2)$	$-\frac{1}{90} (\Delta x)^5 f^{(4)}(\xi)$
Simpson's 3/8	3	$\frac{b-a}{3}$	$\frac{3\Delta x}{8} (f_0 + 3f_1 + 3f_2 + f_3)$	$-\frac{3}{80} (\Delta x)^5 f^{(4)}(\xi)$
Boole's	4	$\frac{b-a}{4}$	$\frac{2\Delta x}{45} (7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4)$	$-\frac{8}{945} (\Delta x)^7 f^{(6)}(\xi)$

Table 1: Newton-Cotes techniques

$$x_i = -1 + \left(i - \frac{1}{2}\right) \Delta x_i, \quad y_j = -1 + \left(j - \frac{1}{2}\right) \Delta y_j \quad (9)$$

With $i = 1, 2, 3, \dots, M$, and $j = 1, 2, 3, \dots, N$. Eq. (2) could be rewritten as follows:

$$L_{pq} = \frac{(2p+1)(2q+1)}{(M-1)(N-1)} \sum_{i=1}^M \sum_{j=1}^N P_p(x_i) P_q(y_j) f(i, j) \quad (10)$$

Eq. (10) is so-called direct method for Legendre moments computations, which is the approximated version using zeroth-order approximation (ZOA). As were indicated by Liao and Pawlak [37], Eq. (10) is not a very accurate approximation of Eq. (1).

Therefore, to improve the accuracy, in 1996, Liao and Pawlak [37] proposed to use the following approximated form:

$$L_{pq} = \frac{(2p+1)(2q+1)}{4} \sum_{i=1}^M \sum_{j=1}^N h_{pq}(x_i, y_j) f(x_i, y_j) \quad (11)$$

where

$$h_{pq}(x_i, y_j) = \int_{x_i - (\Delta x_i/2)}^{x_i + (\Delta x_i/2)} \int_{y_j - (\Delta y_j/2)}^{y_j + (\Delta y_j/2)} P_p(x) P_q(y) dx dy \quad (12)$$

Liao and Pawlak proposed the Alternative Extended Simpson's Rule (AESR) method to evaluate the double integral defined by Eq. (12), and then they use it to calculate the Legendre moments defined by Eq. (11). AESR method is shown in Eq. (13).

$$\int_a^b f(x) dx \cong \frac{h}{48} [17f_0 + 59f_1 + 43f_2 + 49f_3 + 48 \sum_{i=4}^{n-4} f_i + 49f_{n-3} + 43f_{n-2} + 59f_{n-1} + 17f_n] \quad (13)$$

Then, in 2005, Yap and Paramesran [39] indicated that the approximation of the integral terms in Eq. (12)

is responsible for the approximation error of Legendre moments. These integrals need to be evaluated exactly to remove the approximation error of the Legendre moments computation and they proposed a method to compute the exact values of the Legendre moments by mathematically integrating the Legendre polynomials over the corresponding intervals of the image pixels. In 2007, Hosny [40] proposed a new accurate and fast method for exact Legendre moments computation. The set of Legendre moment can be computed exactly by:

$$L_{pq} = \sum_{i=1}^M \sum_{j=1}^N I_p(x_i) I_q(y_j) f(x_i, y_j) \quad (14)$$

where

$$I_p(x_i) = \frac{(2p+1)}{(2p+2)} [x P_p(x) - P_{p-1}(x)]_{U_i}^{U_{i+1}} \quad (15)$$

$$I_q(y_j) = \frac{(2q+1)}{(2q+2)} [y P_q(y) - P_{q-1}(y)]_{V_j}^{V_{j+1}} \quad (16)$$

This kernel is independent of the image. Therefore, this kernel can be pre-computed, stored and recalled whenever it is needed to avoid repetitive computation.

In 2011, Holoborodko [53] indicated that there are several ways on how to improve high-order Newton-Cotes formulas. The most obvious is to re-target some of the degrees of freedom in the system from contributing to highest approximating order to regularization and stronger noise suppression. Natural way of doing this is to use least squares approximation instead of interpolation. On the contrary to interpolation, least squares make possible to derive several integration filters of the same approximation order. Table (2) shows the formulas derived by Holoborodko of $O(h^5)$ and $O(h^7)$ [53]. In this paper, the formula of order 9 will be used seeking to obtain a greater accuracy in the numerical integration.

In general, given the large number of possible ways to compute the integrals associated with the Legendre moments, in this paper is proposed to use the Holoborodko formula of order 9. In addition, it is compared with a traditional one (Composite Simpson's 1/3 rule), AESR proposed by Pawlak, and the one proposed by Hosny.

N	Stable/Low Noise Newton-Cotes Formulas of $O(h^5 f^{(4)})$ and $O(h^7 f^{(6)})$	Error Term
5	$\frac{4\Delta x}{105} (11f_0 + 26f_1 + 31f_2 + 26f_3 + 11f_4)$	$\frac{34(\Delta x)^5}{315} f^{(4)}(\xi)$
6	$\frac{5\Delta x}{336} (31f_0 + 61f_1 + 76f_2 + 76f_3 + 61f_4 + 31f_5)$	$\frac{265(\Delta x)^5}{1008} f^{(4)}(\xi)$
7	$\frac{\Delta x}{14} (7f_0 + 12f_1 + 15f_2 + 16f_3 + 15f_4 + 12f_5 + 7f_6)$	$\frac{39(\Delta x)^5}{70} f^{(4)}(\xi)$
7	$\frac{\Delta x}{770} (268f_0 + 933f_1 + 786f_2 + 646f_3 + 786f_4 + 933f_5 + 268f_6)$	$\frac{17(\Delta x)^7}{308} f^{(6)}(\xi)$
8	$\frac{7\Delta x}{31680} [1657(f_0 + f_7) - 5157(f_1 + f_6) + 4947(f_2 + f_5) + 4079(f_3 + f_4)]$	$\frac{11767(\Delta x)^7}{6400} f^{(6)}(\xi)$
9	$\frac{8\Delta x}{6435} [309(f_0 + f_8) + 869(f_1 + f_7) + 904(f_2 + f_6) + 779(f_3 + f_5) + 713f_4]$	$\frac{2696(\Delta x)^7}{1971200} f^{(6)}(\xi)$

Table 2: Newton-Cotes Formulas derived by Holoborodko

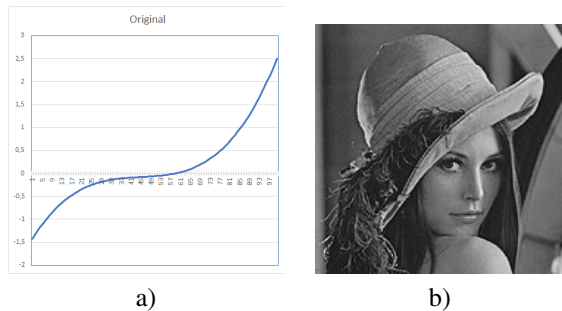


Figure 2: Testing data. a) unidimensional signal b) lena image

4 EXPERIMENTAL RESULTS

To compare the reconstructed signals and images with the originals, here is adopted the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) as the measurements, which are signal or image independent and can be used to evaluate the reconstruction signal performance [36]. PSNR is the ratio between the maximum power of the signal and the affecting noise, and is defined as

$$PSNR = 10 \log_{10} \left(\frac{G_{Max}^2}{MSE} \right) \quad (17)$$

where G_{Max}^2 is the maximum value of the signal or gray level of the image, which is 255 in our case, and MSE is defined by

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |f(x_i, y_j) - \hat{f}(x_i, y_j)|^2 \quad (18)$$

Figure (2) shows the test data (signal and image) used in this paper. The signal is artificially constructed, while the image (Lena) is taken from the traditional images used in digital image processing. The image resolution is 170x170 and has 256 gray levels.

First, the reconstruction of the one-dimensional signal shown in Fig. (2)a is carried out using the different integration techniques mentioned. Signal reconstruction

is generated by varying the number of Legendre Moments and for each of them the MSE and PSNR are calculated respectively. The results obtained from MSE and PSNR for the signal using the four integration techniques to calculate the Legendre moments are shown in Fig. (3). In Fig. 3a can be seen that the values for MSE are very similar and have a decreasing behavior and tend to zero, which implies a good signal reconstruction. Regarding the PSNR, Fig. 3b shows that the best result corresponds to the Hosny technique. Additionally, it can be seen that the four techniques reach a stable state from the fifth order of the Legendre Moments.

Fig. (4) shows the reconstructed signals using five Legendre moments. The original signal is also presented to compare the behavior of the integration techniques and the reconstructed signal from the computation of the Legendre moments. It can be concluded that the reconstructed signals present a high precision with respect to the original signal. However, the biggest error corresponds to the AESR-based technique, which generates differences in the initial part of the reconstruction. The ZOA, Holoborodko and Hosny techniques present better precision in the reconstructed signal.

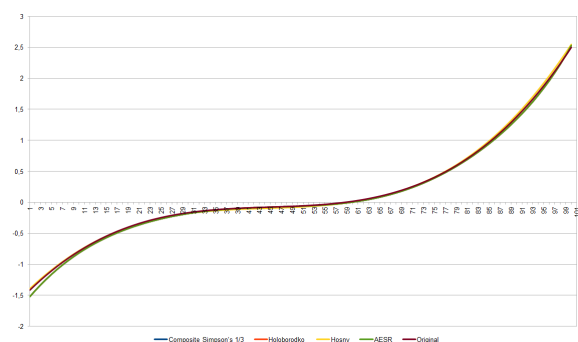


Figure 4: Signal reconstruction.

Regarding image reconstruction, Fig. (5) shows the results of MSE and PSNR. It can be observed that the behavior of the MSE is decreasing for the four techniques applied in the first 90 Legendre Moments, however, this situation begins to change for the Holoborodko, AESR

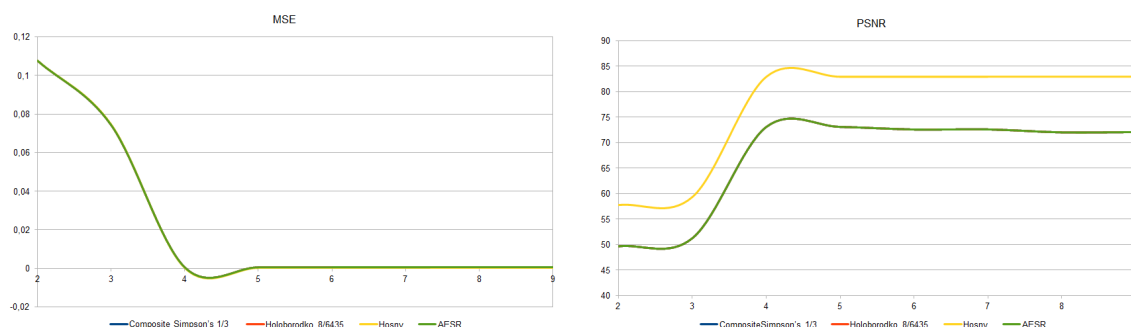


Figure 3: MSE and PSNR for one-dimensional signal.

and Composite Simpson 1/3 techniques. These techniques become unstable and increasing, generating possible results of low precision in the image reconstruction. For its part, the technique proposed by Hosny keeps the MSE decreasing and with a tendency to stabilization, from the Legendre Moment 160, approximately.

Fig. (5) also presents the PSNR behavior for Lena image, where it can be seen that the stability of the technique proposed by Hosny is high, regardless of the order of the Legendre Moments degree. For its part, the PSNR precision using the other techniques (AESR, ZOA and Holoborodko) is reduced from order 90, which implies a possible reduction in the image reconstruction accuracy.

In Fig. (6) three images reconstructed by each of the techniques used (ZOA, AESR, Holoborodko, and Hosny) are presented. Images reconstruction were calculated using Legendre Moments with order between 10 and 300 with steps of 10. Based on the behavior of the MSE and the PSNR, those of order 90, 100 and 110 were selected (see Fig. (5)). As can be seen for the different orders, the images present accuracy problems, mainly at the edges, where pixels with incorrect gray levels appear. Additionally, for the order 110, the accuracy problems are resolved in all cases, but the MSE and PSNR values are better for the Hosny technique. In the particular case of Holoborodko technique, it can be observed that it presents good results, but it does not exceed those obtained using the technique proposed by Hosny.

5 CONCLUSIONS

For the computation of Legendre Moments there are different techniques with different levels of precision. The use of the numerical integration formulas for uniformly spaced data proposed by Holoborodko, derived from the stable Newton-Cotes quadrature rules that are based on the least squares approximation instead of interpolation, was proposed. The Holoborodko technique was compared with respect to techniques such as ZOA, AESR and the one proposed by Hosny, reaching satisfactory results. The results were better than using ZOA

and AESR, but it is necessary to continue exploring the Holoborodko technique using more test images to obtain conclusive information about its strengths and weaknesses. In future works it is necessary to identify the causes of loss of precision after a certain order of the Legendre Moments, because initially it can be seen that precision is lost since many values of the orthogonal product of the legendre polynomials become close to zero. This causes deterioration in the reconstructed images, as can be seen in the MSE and PSNR figures.

6 REFERENCES

- [1] S. Qi, Y. Zhang, C. Wang, J. Zhou, and X. Cao, "A Survey of Orthogonal Moments for Image Representation: Theory, Implementation, and Evaluation," *ACM Computing Surveys*, vol. 55, no. 1, pp. 1–35, jan 2023.
- [2] W. Huang, C. Chen, M. Sarem, and Y. Zheng, "Overlapped rectangle Image representation and its application to exact Legendre moments computation," *Geo-Spatial Information Science*, vol. 11, no. 4, pp. 294–301, dec 2008.
- [3] R. Mukundan and K. R. Ramakrishnan, "FAST COMPUTATION OF LEGENDRE AND ZERNIKE MOMENTS," *Pattern Recognition*, vol. 28, no. 9, pp. 1433–1442, 1995.
- [4] M. R. Teague, "Image Analysis Via the General Theory of Moments," *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, 1980.
- [5] S. H. Abdulhussain, B. M. Mahmmoud, A. Al-Ghadhban, and J. Flusser, "Face Recognition Algorithm Based on Fast Computation of Orthogonal Moments," *Mathematics*, vol. 10, no. 15, aug 2022.
- [6] L. Maofu, H. Yanxiang, and Y. Bin, "Image Zernike moments shape feature evaluation based on image reconstruction," *Geo-Spatial Information Science*, vol. 10, no. 3, pp. 191–195, 2007.
- [7] C.-H. Lo and H.-S. Don, "3-D moment forms: Their construction and application to object

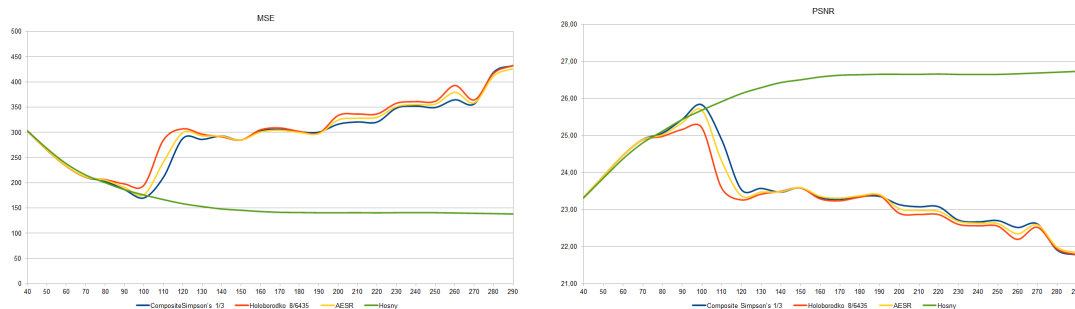


Figure 5: MSE and PSNR for two-dimensional signal (image).

Moment	Comp. Simpson's 1/3	Holoborodko	AESR	Hosny
90				
100				
110				

Figure 6: Reconstructed images from different Legendre moments orders.

- identification and positioning,” *Pattern Analysis and Machine Intelligence*, vol. 11, no. 10, pp. 1053–1064, 2002. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=42836
- [8] J. Flusser and T. Suk, “Pattern recognition by affine moment invariants,” *Pattern Recognition*, vol. 26, no. 1, pp. 167–174, 1993.
- [9] S. A. Dudani, K. J. Breeding, and R. B. McGhee, “Aircraft Identification by Moment Invariants,” *IEEE Transactions on Computers*, vol. C-26, no. 1, pp. 39–46, 1977.
- [10] M. K. Hu, “Visual Pattern Recognition by Moment Invariants,” *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.
- [11] M.W. Nasrudin, N. S. Yaakob, N. A. Abdul Rahim, M. Z. Zahir Ahmad, N. Ramli, and M. S. Aziz Rashid, “Moment Invariants Technique for Image Analysis and Its Applications: A Review,” in *Journal of Physics: Conference Series*, vol. 1962, no. 1. IOP Publishing Ltd, jul 2021.
- [12] M. Pawlak, “Image Analysis by Moments: Reconstruction and Computational Aspects,” Ph.D. dissertation, 2006.
- [13] X. S. Liao, “Image Analysis by Moments,” Ph.D. dissertation, 1993.
- [14] C. H. Teh and R. T. Chin, “On image analysis by the methods of moments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. December, pp. 556–561, 1988.
- [15] R. C. Papademetriou, “Reconstructing with moments,” in *Proceedings - International Conference on Pattern Recognition*, 1992, pp. 476–480.
- [16] L.-M. Luo, X.-H. Xie, and X.-D. Bao, “A Modified Moment-Based Edge Operator for Rectangular Pixel Image,” *IEEE Transactions on Circuits*

- and Systems for Video Technology, vol. 4, no. 6, pp. 552–554, 1994.
- [17] S. Ghosal and R. Mehrotra, “Orthogonal Moment Operators for Subpixel Edge Detection,” *Pattern Recognition*, vol. 26, no. 2, pp. 295–306, 1993.
 - [18] L. M. Luo, C. Hamitouché, J. L. Dillenseger, and J. L. Coatrieux, “A Moment-Based Three-Dimensional Edge Operator,” *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 7, 1993.
 - [19] M. Tuceryan, “Moment based texture segmentation,” in *Proceedings - International Conference on Pattern Recognition*, 1992, pp. 45–48.
 - [20] R. J. Prokop and A. P. Reeves, “A survey of moment-based techniques for unoccluded object representation and recognition,” *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 5, pp. 438–460, 1992.
 - [21] R. Mukundan and K. R. Ramakrishnan, “Legendre Moments,” in *Moment Functions in Image Analysis: Theory and Applications*. World Scientific Publishing, 1998, ch. 4, pp. 49–56.
 - [22] H. Shu, L. Luo, and J.-L. Coatrieux, “Moment-based approaches in imaging part 3: computational considerations,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 27, no. 3, pp. 89–91, 2008.
 - [23] —, “Moment-based approaches in imaging part 2: invariance,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 27, no. 1, pp. 81–83, 2008.
 - [24] —, “Moment-based approaches in imaging part 1, basic features,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 26, no. 5, pp. 70–74, 2007.
 - [25] J. Flusser, T. Suk, and B. Zitová, *Moments and Moment Invariants in Pattern Recognition*. John Wiley Sons, Ltd., 2009.
 - [26] —, *2D and 3D Image Analysis by Moments*. John Wiley Sons, Ltd, 2017.
 - [27] T. V. Hoang, “Image Representations for Pattern Recognition Image,” Ph.D. dissertation, Université Nancy II, 2011.
 - [28] G. Papakostas, “Over 50 Years of Image Moments and Moment Invariants,” in *Moments and Moment Invariants - Theory and Applications*. Science Gate Publishing, 2014, vol. 1, no. July 2014, pp. 3–32.
 - [29] P. Kaur, H. S. Pannu, and A. K. Malhi, “Comprehensive study of continuous orthogonal moments-A systematic review,” *ACM Computing Surveys*, vol. 52, no. 4, 2019.
 - [30] C. D. Ruberto, L. Putzu, and G. Rodriguez, “Fast and accurate computation of orthogonal moments for texture analysis,” *Pattern Recognition*, vol. 83, pp. 498–510, 2018.
 - [31] S. K. Hwang and W. Y. Kim, “A novel approach to the fast computation of Zernike moments,” *Pattern Recognition*, vol. 39, no. 11, pp. 2065–2076, 2006.
 - [32] B. Vijayalakshmi and V. Subbiah Bharathi, “Classification of CT liver images using local binary pattern with Legendre moments,” *Current Science*, vol. 110, no. 4, pp. 687–691, 2016.
 - [33] M. Oujaoura, B. Minaoui, and M. Fakir, “Image Annotation by Moments,” in *Moments and Moment Invariants - Theory and Applications*, jul 2014, ch. 10, pp. 227–252.
 - [34] K. Wu, C. Garnier, J. L. Coatrieux, and H. Shu, “A preliminary study of moment-based texture analysis for medical images,” in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC’10*, no. 1, 2010, pp. 5581–5584.
 - [35] A. N. Hashimi and B. N. Kadhim, “Face recognition based on fusion of SVD and Legendre moment,” in *Journal of Physics: Conference Series*, vol. 1530, no. 1. Institute of Physics Publishing, may 2020.
 - [36] S. Liao, “Accuracy Analysis of Moment Functions,” in *Moments and Moment Invariants - Theory and Applications*. Science Gate Publishing, jul 2014, ch. 2, pp. 33–56.
 - [37] S. X. Liao and M. Pawlak, “On image analysis by moments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 254–266, 1996.
 - [38] R. Mukundan, S. H. Ong, and P. A. Lee, “Discrete vs. continuous orthogonal moments for image analysis,” *CISST’01 International Conference*, no. i, pp. 23–29, 2001. [Online]. Available: <http://ir.canterbury.ac.nz/handle/10092/470>
 - [39] P.-T. Yap and R. Paramesran, “An Efficient Method for the Computation of Legendre Moments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, 2005.
 - [40] K. M. Hosny, “Exact Legendre moment computation for gray level images,” *Pattern Recognition*, vol. 40, no. 12, pp. 3597–3605, dec 2007.
 - [41] C. J. Bustacara Medina, “Evaluación computacional para calcular los polinomios de Legendre de primera clase,” *Revista Avances en Sistemas e Informática*, vol. 7, no. 2, pp. 131–137, 2010.
 - [42] P. Wang, “3-D Image Analysis via Jacobi Moments with GPU-Accelerated Algorithms,” *Tech. Rep.*, 2021.
 - [43] I. Khalil, A. Khalil, S. U. Rehman, H. Khalil, R.

- A. Khan, and F. Alam, "Classification of ECG Signals Using Legendre Moments," *International Journal of Bioinformatics and Biomedical Engineering*, vol. 1, no. 3, pp. 284–291, 2015. [Online]. Available: <http://www.aiscience.org/journal/ijbbehttp://creativecommons.org/licenses/by-nc/4.0/>
- [44] P.-J. A. Chiang, "Legendre Moments Explorations via Image Reconstruction," Ph.D. dissertation, 2014.
- [45] G. B. Arfken, H. J. Weber, and F. E. Harris, *Mathematical Methods for Physicists*, 7th ed. Elsevier Inc., 2013.
- [46] I. A. Selezneva, Y. L. Ratis, E. Hernández, J. Pérez-Quiles, and P. Fernández de Córdoba, "A CODE TO CALCULATE HIGH ORDER LEGENDRE POLYNOMIALS AND FUNCTIONS," *Revista Academica Colombiana de Ciencia*, vol. 37, no. 145, pp. 541–544, 2013.
- [47] G. Yang, H. Shu, C. Toumoulin, G.-N. Han, and L. M. Luo, "Efficient Legendre moment computation for grey level images," *Pattern Recognition*, vol. 39, no. 1, pp. 74–80, 2006. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00135862>
- [48] T. L. Chow, *Mathematical methods for physicists : A concise introduction*. Cambridge University Press, 2000.
- [49] E. Marengo, E. Robotti, and M. Demartini, "The use of legendre and zernike moment functions for the comparison of 2-D PAGE maps," in *Methods in Molecular Biology*. Humana Press Inc., 2016, vol. 1384, pp. 271–288.
- [50] S. X. Liao and M. Pawlak, "On the accuracy of zernike moments for image analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1358–1364, 1998.
- [51] K. M. Hosny, "Image representation using accurate orthogonal Gegenbauer moments," *Pattern Recognition Letters*, vol. 32, no. 6, pp. 795–804, 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.patrec.2011.01.006>
- [52] —, "Efficient Computation of Legendre Moments for Gray Level Images," *International Journal of Image and Graphics*, vol. 7, no. 4, pp. 735–747, 2007.
- [53] P. Holoborodko, "Stable Newton-Cotes Formulas," 2011. [Online]. Available: https://www.researchgate.net/publication/316662645_Stable_Newton-Cotes_Formulas

Optimised Light Rendering through Old Glass

Quentin Huan
Université du Littoral
Côte d'Opale
LISIC
BP 719
France, 62228, Calais
cedex
quentin.huan@univ-littoral.fr

Francois Rousselle
Université du Littoral
Côte d'Opale
LISIC
BP 719
France, 62228, Calais
cedex
francois.rousselle@univ-littoral.fr

Christophe Renaud
Université du Littoral
Côte d'Opale
LISIC
BP 719
France, 62228, Calais
cedex
christophe.renaud@univ-littoral.fr

ABSTRACT

We propose a rendering method for efficiently computing the transmitted caustics produced by a glass panel with arbitrary surface deformations, characteristic of old glass used in 3D reconstructions in virtual heritage. Using Fermat's principle of least time, we generalize the concept of Next Event Estimation to allow light sampling through two displaced refractive interfaces, which amount to numerically finding all stationary points of an objective function. Our work allows for an efficient estimation of the caustic while staying inside a standard Monte Carlo pathtracing framework. Our specific geometrical context allows our solver to converge significantly faster than the more general method Specular Manifold Sampling, while scaling well with the number of panels present in the scene.

Keywords

Ancient glass, Virtual heritage, Caustics, Pathtracing, Sampling, Newton's method

1 INTRODUCTION

Recent progress in lighting simulation offers a true opportunity for recreating the luminous atmosphere that existed in ancient architectures. The virtual restitution of architectural heritage gives to historians a new set of tools allowing a better analysis and understanding of life, work and worship conditions through comparison with the existing written archives.

For centuries, glass has been used in the windows of most buildings. However, the production of high quality glass is relatively recent and old glass, as used in windows and stained glass, has specific visual characteristics that greatly impact the lighting of indoor scenes. These include surface irregularities, or the presence of various bubbles and debris due to the different manufacturing processes (crown or cylinder blown sheet glass), which did not allow for perfectly flat or homogeneous surfaces. These irregularities, however small, produce distortions in the perception of the external environment but also complex lighting patterns

(caustics) on objects illuminated directly by a source (sun, flames) through these types of glass (see Fig. 1). The composition of the glass paste could also lead to slightly colored glass, influencing the color of perceived light.

These effects are still challenging to compute with modern rendering techniques, often requiring prohibitive rendering time to produce noise-free images. On the other hand, they are of great interest to the historian, in order to understand the differences in light atmosphere that may have existed in the past compared to the atmosphere produced by our modern glazing.



Figure 1: Photography of a caustic produced by a slightly irregular panel of glass.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

In this article, we are only considering the lighting effects produced by the irregularities of glass surfaces in windows. By neglecting volume irregularities (bubbles, impurities inclusions e.g.) and the possible colouring of the glass, we can model an ancient glass panel by two parallel and slightly displaced planes. By taking advantage of this particular geometrical context, we can significantly accelerate the rendering time of scenes including these complex objects.

The contributions of this paper are:

- a geometrical framework for modeling the surface irregularities of ancient glass panels ;
- a solver based on Fermat's principle allowing the connection of two points through two refractive interfaces ;
- a robust initialisation strategy for path sampling that scales well with the number of glass panels present in the scene.

In the next paragraph, we review previous works relevant to our problem from the point of view of antique glass and caustics rendering. In paragraph 3, we detail the different steps of our approach. We then present the results obtained in terms of convergence speed of our solver, but also from the point of view of the interest of the initialization strategy that we propose when there are many windows. We discuss in section 5 some residual difficulties of our approach which are also present on the approaches of the state of the art, then conclude this article by proposing some avenues for future research.

2 PREVIOUS WORK

To our knowledge, no previous work specifically tackle the rendering of the caustics produced by ancient glass panels. Kider Jr et al. [Jr+09] recreate the characteristic caustic lighting pattern produced by early Islamic light sources using a specialised technique. This method doesn't take in account the surface imperfections of the glass fixture, resulting in a simplified simulation of the effects. Grobe et al. [GNL20] measure the BSDF of various flat Roman window glass samples. They then proceed to simulate daylight lighting of an interior scene using a data-driven transmission model. While this work captures the small scale in-homogeneity of the material, it doesn't take in considerations the larger surface variations that play an essential role for transparent and homogeneous glass samples that only refract incident light without significantly scattering it.

The rendering of caustics produced by refractive or reflective objects has been a long lasting problem in computer graphics. While standard pathtracing is able to compute caustics generated by a specular object lighted

by an area light, its slow convergence rate makes it difficult to use in practice. Many different rendering techniques have been designed to accelerate the convergence of the standard algorithm.

Pathtracing

Pathtracing (PT) is a popular unbiased estimation method of the rendering equation originally described by Kajiya [Kaj86]. While being able to accurately render scenes with complex light transport (including caustics), the standard algorithm may suffer from impractically slow convergence rates even with GPU acceleration. Although slow, pathtracing can be used to produce accurate ground truth reference images.

The more sophisticated Bidirectional Path Tracing technique [LW93] speeds up convergence for Diffuse-Specular paths (see the A area in Fig. 2b), but remains unable to efficiently deal with Specular-Diffuse-Specular (SDS) paths that are commonly encountered with reflective or refractive objects (see Fig. 2b, where the B area is slower to converge).

Photon Mapping

Techniques based on Photon Mapping [Jen96] are generally well suited for caustics rendering. Photons coming from the light sources are traced around the scene and stored in an auxiliary data structure called a photon map. Since the amount of photons traced is finite, a photon density estimation step is needed in order to compute the rendering integral, which requires a voluminous amount of storage in certain cases.

In related previous work [Shi90], the contribution of illumination rays coming from the light sources were stored in a texture (an illumination map) associated to each diffuse surfaces. This however requires a sufficiently high resolution for the illumination map to avoid artifacts.

The stochastic progressive photon mapping technique (SPPM) proposed by Hachisuka and Jensen [HJ09] allows the progressive construction of the photon map during render time, alleviating the storage issue. While this technique handles SDS paths, it also introduces bias in the final image causing the caustic to appear blurry (see Fig. 2c).

Metropolis Light Transport

Metropolis Light Transport (MLT) of Veach and Guibas [VG97] uses Metropolis-Hasting integration to evaluate the rendering integral. A path space formulation of the rendering equation allows the construction and mutation of a group of bootstrap paths, allowing the

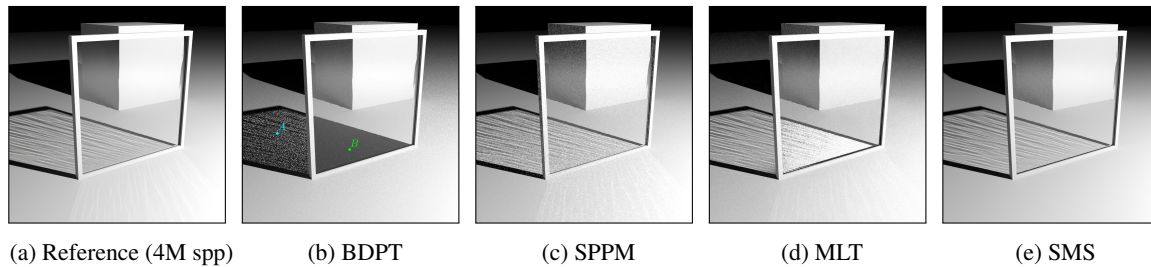


Figure 2: Comparison of different rendering techniques for a simple scene including a glass panel whose surfaces are not smooth (identical number of 200 samples per pixel (spp)). Images were rendered with PBRT-V4 using (a) Pathtracing, (b) Bidirectional Path Tracing, (c) Stochastic Progressive Photon Mapping and (d) Metropolis Light Transport. Image (e) Specular Manifold Sampling was rendered with the Path-SMS-MS integrator (using the author implementation in Mitsuba2 [Nim+19]) with 2 specular bounces (thus lacking the reflected caustic).

local exploration of path space. This approach is particularly efficient for scenes in which the light transport mainly occurs throughout few highly contributing paths that are difficult to sample using traditional techniques. Intrinsically, the convergence process is highly dependent of the random seed used and non-uniform by nature: some paths that are difficult to sample may be explored lately during render time, causing some brightness inconsistencies until the process completely converges (see Fig. 2d where the SDS paths can be alternatively brighter or darker than expected depending on the seed). This may lead to unpleasant flickering artifacts when rendering animations.

Manifold Exploration

Early work from Mitchell and Hanrahan [MH92] makes use of Fermat's principle of least time combined with interval arithmetic for computing the illumination produced by reflective, implicitly defined surfaces. This method is deterministic and is able to compute the whole set of paths connecting two points through a one-bounce perfectly specular reflection.

Similarly with Manifold Exploration [JM12], Jakob and Marschner present a new set of mutations for MLT that are more suited to paths involving purely specular events. This set of paths is a manifold described by a set of specular reflection and refraction constraints C that each path satisfies. The use of the implicit function theorem allows to walk over this manifold by using the gradient of the specular constraint ∇C and a Newton type solver. This action is referred as Manifold Walk and has been applied in a standard pathtracing context by Hanika et al. in Manifold Next Event Estimation (MNEE) [HDF15] in order to construct paths connecting an observed point O to a point S sampled on a light source only visible through specular interactions. This technique is only able to find one solution and is thus limited to specular objects that are regular enough.

The Specular Manifold Sampling technique (SMS) of Zeltner et al. [ZGJ20] generalizes MNEE to the cases

where the geometry of the specular objects become more complex and more than one path between O and S may exist. This stochastic method allows unbiased rendering of caustics while staying in a standard Monte Carlo pathtracing context. Note that this method won't produce more physically accurate caustics than standard pathtracing, but it will converge to the result faster.

Our present work will make use of the same context described in [ZGJ20], with a problem specific solver based on Fermat's principle instead of manifold walk (Fermat Next Event Estimation, FNEE). As illustrated by Fig.2, we chose to compare our method only to Specular Manifold Sampling since this technique seemed to achieve the best balance between performance and fidelity for our use case.

In the following, we will focus on the transmitted caustic, but our work can be adapted to generate the reflected caustics as well.

3 METHODOLOGY

A glass windowpane displaying surface deformations is represented by two parametric surfaces $\{\Gamma_1, \Gamma_2\}$ (Fig. 3), each described by an elevation function $h_i(u, v)$ $i \in \{1, 2\}$ such that $\Gamma_i = \{X \in \mathbb{R}^3 / X = (u, v, h_i(u, v))\}$ with $(u, v) \in [0, 1]^2$. In the following, the h_i functions are considered twice differentiable. The ray - surface intersections are resolved using sphere tracing [Har96] in order to avoid making further assumption on how the h_i are defined (bi-cubic interpolation of an heightmap e.g.).

In the most general case where the two interfaces are not flat (i.e. the $h_i(u, v)$ functions are not constant), several paths connecting O and S may exist. This results in the formation of a caustic that is strongly dependent on the geometry of the two interfaces (Fig. 1).

Our objective is to compute the direct lighting that comes through the panel. This is done by extending the well known Next Event Estimation technique [SWZ96] to the case where two refractive interfaces occlude the

observed point O from the point S sampled on a light source.

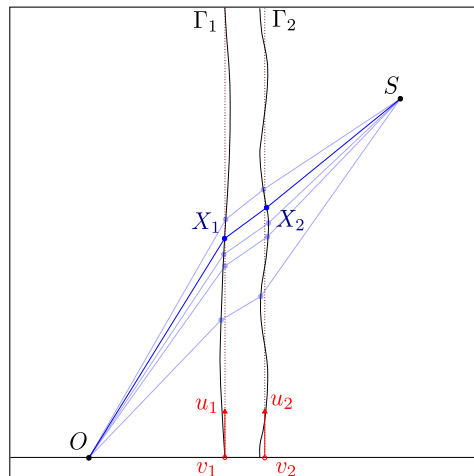


Figure 3: Transmitted ray parametrization. There is in general multiple paths connecting an observed point O to a source S .

3.1 Finding an admissible path

In the following, we will describe a stochastic procedure to construct one valid path and compute its contribution. The procedure can then be re-conducted for each path tracing samples, effectively computing the contribution of all feasible paths when the number of samples is large enough.

Fermat's principle

Fermat's principle (or *principle of least time*) states that for any given two points of space O and S , the optical length of the paths taken by a light ray between O and S is stationary.

Let $L = \int_{\mathbf{P}} \eta(s) ds$ be the optical length of a given path \mathbf{P} , with $\eta(s)$ the index of refraction of the medium at position s . According to Fermat's principle, the set of admissible paths connecting O to S satisfy $dL = 0$.

- In the case of two points in an homogeneous medium of refraction index η , the solution is uniquely given by the straight line connecting the two points.
- With a straight windowpane in between O and S , the solution is still unique, but is now composed of three line segments OX_1 , X_1X_2 and X_2S .
- If the windowpane has irregular surfaces, the solution is no longer unique (see Fig. 3). With the added hypothesis of the h_i being small and regular enough, we can still consider that the solutions are composed of three line segments.

Hence,

$$L(u_1, v_1, u_2, v_2) = \eta_e \|OX_1\| + \eta_i \|X_1X_2\| + \eta_e \|X_2S\|$$

with η_e , η_i the refractive indexes of respectively the exterior or interior medium, and (u_i, v_i) the parametric coordinates of the point X_i . The differential dL is given by:

$$dL = \eta_e \left(\frac{OX_1}{\|OX_1\|} \cdot dX_1 + \frac{X_2S}{\|X_2S\|} \cdot dX_2 \right) + \eta_i \left(\frac{X_1X_2}{\|X_1X_2\|} \cdot (dX_1 + dX_2) \right) \quad (1)$$

Since $dL = 0 \Leftrightarrow \nabla L = 0$, we have to solve a system of 4 non-linear equations of 4 unknowns (u_1, v_1, u_2, v_2) . The solutions of this system will give us the parametric coordinates of X_1 and X_2 , allowing us to construct admissible paths.

Other configurations

Note that we can derive in the same way a system of 2 equations of 2 unknowns (u_2, v_2) for solving caustics generated by light being reflected on one face of the panel (see figure 2a for example). In that case,

$$L(X_2) = \eta_e \|\overrightarrow{OX_2}\| + \eta_e \|\overrightarrow{X_2S}\|$$

$$dL = \eta_e \left(\frac{OX_2}{\|OX_2\|} + \frac{X_2S}{\|X_2S\|} \right) \cdot dX_2$$

We could do the same for an arbitrary number of specular reflections or transmissions (similarly to Manifold walk, the Hessian matrix would have a diagonal block structure). The method used to solve the system of equation and to compute paths contributions are then similar for all configurations.

Solving the system

The system is solved using Newton's method for optimisation. Newton's method is known for converging to stationary points of the objective function regardless of their nature (minimal, maximal or saddle). While that can be an issue in a general optimization context where we want to either maximize or minimize an objective function, it is a useful property in our case. Since h_i are twice differentiable, we can derive an exact analytic expression for the 4×4 Hessian matrix H from the expression of ∇L .

The method is iterative and consists (at step k) in finding a descent direction v_k using curvature information at the point $\theta_k = (u_1^k, v_1^k, u_2^k, v_2^k)$. We can then compute

the next point θ_{k+1} by moving along v_k by a step of size t_k (Eq. 2).

$$\begin{aligned} v_k &= -[H|_{\theta_k}]^{-1} \cdot \nabla L|_{\theta_k} \\ \theta_{k+1} &= \theta_k + t_k \cdot v_k \end{aligned} \quad (2)$$

The step size t_k is found by the Armijo rule [Arm66] using a backtracking strategy (see Algorithm 1) with constants $\alpha = 0.45$ and $\beta = 0.5$.

Algorithm 1 Line search with backtracking

```

 $t_k \leftarrow 1$ 
while  $(L(\theta_k + t_k \cdot v_k) > L(\theta_k) + \alpha \cdot t_k \cdot \nabla L|_{\theta_k} \cdot v_k)$  do
     $t_k \leftarrow \beta \cdot t_k$ 
end while
 $\theta_k \leftarrow \theta_k + t_k \cdot v_k$ 

```

In practice, we limit the number of iterations to 20 and define a convergence threshold to $\|\nabla L|_{\theta_k}\| < \nabla L_\epsilon = 10^{-4}$. If the threshold is not reached after 20 iterations, we consider that the process has failed to converge.

Newton's method convergence behavior is well known to be complex even with relatively simple functions [HSS01]. Since the h_i functions are potentially complex, finding a good initial guess is in general challenging. The initial value θ_0 is thus chosen randomly inside $[0, 1]^4$.

We also found that imposing a minimal value for t_k is generally beneficial to prevent the solver from getting stuck ($t_{min} = 0.004$).

3.2 Path contribution

After finding a potential path, we need to compute its contribution to the lighting of the observed point O .

Let $\bar{X} = (O, X_1, X_2, S)$ be the path found by the solver. The contribution of \bar{X} is given by

$$f(\bar{X}) = L_e \cdot \frac{G(O \leftrightarrow S)}{PDF(\bar{X})} \cdot BSDF_{X_1} \cdot BSDF_{X_2} \cdot V(\bar{X}) \quad (3)$$

with L_e the radiance emitted from the light source and $V(\bar{X})$ being 0 if the path \bar{X} is occluded, 1 otherwise. $BSDF_{X_1}$ is the value of the bidirectional scattering distribution function at vertex X_1 , with incoming light direction $\overrightarrow{X_1 X_2}$ and outgoing light direction $\overrightarrow{X_1 O}$ (respectively, we have for $BSDF_{X_2}$ incoming direction $\overrightarrow{X_2 S}$ and $\overrightarrow{X_2 X_1}$ for outgoing direction).

The ratio $\frac{G(O \leftrightarrow S)}{PDF(\bar{X})}$ is composed of the generalized geometry factor [JM12], and the probability density of finding the solution path \bar{X} using our solver.

Generalized geometry factor

The generalized geometry factor is defined as:

$$G(X_0 \leftrightarrow X_n) = \frac{d\omega^\perp}{dA_n} = \frac{d\omega^\perp}{dA_1} \cdot \frac{dA_1}{dA_n}$$

Intuitively, it represents the tendency of a ray bundle to spread out or focus when subject to a sequence of refraction or reflection.

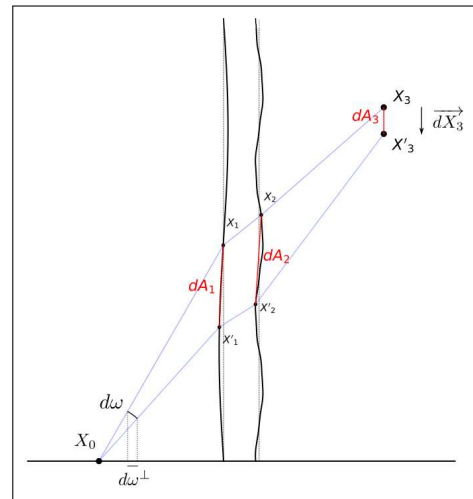


Figure 4: Generalized geometry factor parametrisation in the case of light transmission through a glass panel.

The term $\frac{d\omega^\perp}{dA_1}$ is simply the usual Geometry factor given by:

$$G(X_0 \leftrightarrow X_1) = \frac{d\omega^\perp}{dA_1} = \frac{|N(X_0) \cdot \overrightarrow{X_0 X_1}| \cdot |N(X_1) \cdot \overrightarrow{X_1 X_0}|}{\|X_0 - X_1\|^2}$$

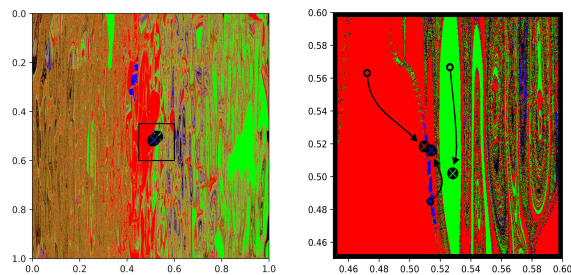
with $N(X_i)$ the normal vector at vertex X_i and ω^\perp the projected solid angle.

The other factor $\frac{dA_1}{dA_n}$ is then computed from ∇C , the derivative of the specular constraint of \bar{X} [JM12].

Alternatively, pencil tracing can also be used to compute G by the mean of the transfer matrices of the optical system [STN87] [KHD14].

Inverse probability estimation

As explained in [ZGJ20], the probability of sampling the solution \bar{X} corresponds to the volume of the convergence basin of this solution. For a given solution \bar{X} , its convergence basin is defined as the set of all the initial values θ_0 that converges to this solution (Fig. 5).



(a) Convergence basins at a point O for a source S for a simple blown glass profile. (b) Zoom on the solution cluster.

Figure 5: The solver converges to 3 solutions colored in red, blue and green. Each point is colored depending on which solution the solver converges to, starting from this point. Their respective convergence basins are highly irregular and nearly cover the whole space. These 4D basins are projected onto the 2D plane for visualization purpose (i.e. we choose $(u_1, v_1) \in [0, 1]^2$ and impose $u_1 = u_2$ and $v_1 = v_2$).

These convergence basins are in general highly irregular and their volume cannot be computed exactly. The inverse of their volume can however be estimated by an iterative process that resorts to counting the number of trials necessary for the solver to converge to \bar{X} from a random initial starting point.

$$\frac{1}{PDF(\bar{X})} \simeq N_{trial}$$

In practice, it is necessary to clamp this estimator to N_{max} to avoid potentially infinite loops if the basins are close to being singular (typical value $N_{max} = 1000$).

3.3 Initialisation strategy

In this section, we discuss the initialisation strategy we use for selecting an appropriate $\{\Gamma_1, \Gamma_2\}$ surface pair given two points O and S that we try to connect. Suppose that multiple specular surfaces are present in the scene. In the general case where the interfaces can have any shape, we can't easily choose an appropriate couple $\{\Gamma_1, \Gamma_2\}$ for initializing the solver. We still can resort to selecting a random pair for each sample, which is the strategy used by Specular Manifold Sampling: select a *caustic caster* marked shape at random, then launch one estimate per *caustic bounce* marked shape. Unfortunately, in practice, most pairs lead to no solutions and are thus bad initialisation choices (see Fig. 6).

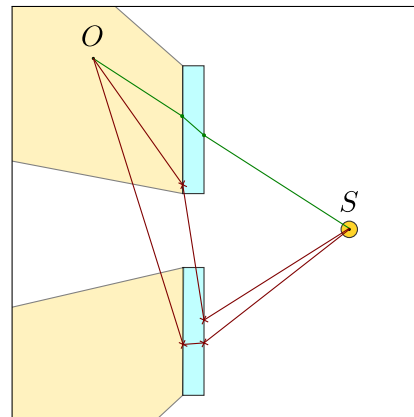


Figure 6: Choice of an interface pair: most combinations lead to a poor initialisation (red paths) that prevents the solver from converging to a solution.

In our case, having to handle many windows in a scene, we can make a better choice.

Consider that we have N glass panes. For each glass pane g_i , the most natural pairing would be to use $\Gamma_1^{g_i}$ and $\Gamma_2^{g_i}$ the front and back faces of the glass pane, thus reducing the number of initialisation choices from $4N^2$ (since there is $2N$ faces) to N .

Since the caustic generated by the window is mostly contained inside the shadow cast by a flat glass pane (that we will refer to as the *approximate caustic area* in the following), we can further reduce the number of possibilities by selecting the glass pane that occluded the shadow ray cast from O in the direction of S (see point O in Fig. 7). That generally leaves us with a unique initialisation possibility. In the contrary case, we resort to standard pathtracing (see point O').

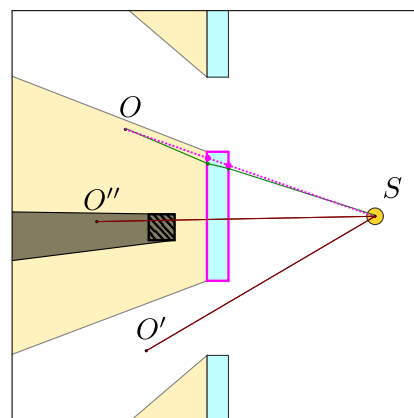


Figure 7: Window selection during light sampling. The *approximate caustic area* is colored in light yellow.

This strategy however produces unnaturally sharp shadows (see Fig. 8b) for occluders situated in front or behind the chosen window (see point O'' in Fig. 7). In practice, the caustic tends to bleed in a small region outside of the approximate caustic area (see Fig. 8 a). We

can easily account for this case by choosing to launch the solver with a random glass pane whenever an object occludes the shadow ray. This may come at the cost of a loss of performance (see Fig. 8 c). This solution isn't entirely satisfactory and more sophisticated approach [WHY20] may be considered in future work.

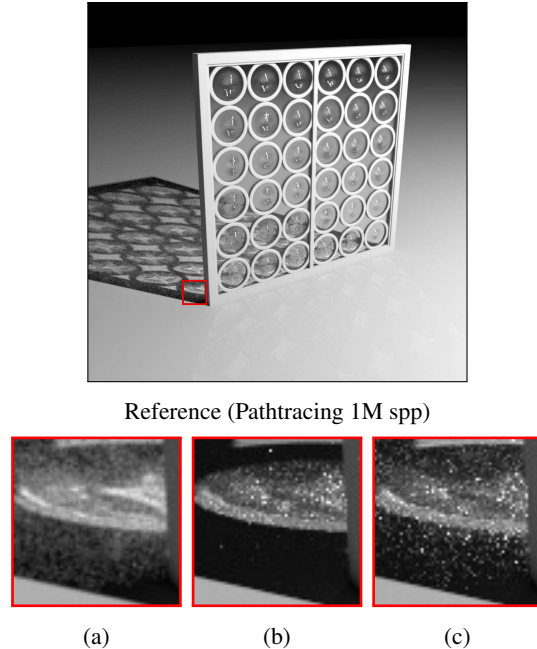


Figure 8: Illustration of the bias introduced by the two initialisation strategies: (a) Reference, (b) FNEE with sharp shadows (1000 sec), (c) FNEE with caustic bleed (1000 sec).

4 RESULTS

Solver comparison

In this section, we compare our solver against the predictor-corrector scheme used by Specular Manifold Sampling. For comparison fairness, the two solvers are implemented inside the same integrator in Mitsuba2 [Nim+19] and thus use the initialisation scheme described in the previous section. Both solvers use double precision arithmetic which is necessary to deal with thin windows (thickness around 1mm).

The images produced by FNEE and SMS were compared against a ground truth image generated by pathtracing. The Mean Squared Error (MSE) metric was used:

$$MSE = \frac{1}{n} \sum_{i=0}^n (x_i - \hat{x}_i)^2$$

with x_i the pixels values of the ground truth image (generated by pathtracing) and \hat{x}_i the pixels values of the image being compared.

Our benchmark scene (Fig. 9) is a window composed of 4 different glass panes with various elevation profiles

on the face facing the camera. These profiles are modeled from typical windows of the XIV to XIX centuries (crown and cylinder blown sheet glass).

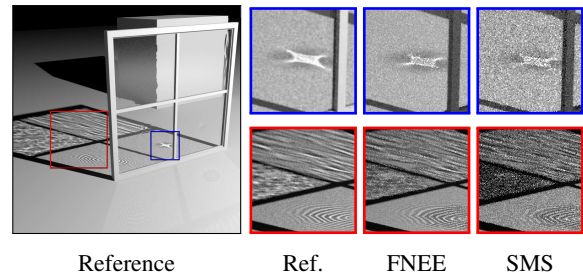


Figure 9: Visual comparison of our method (FNEE) with Specular Manifold Sampling (SMS) on various perturbation profiles (equal time $t = 100s$). Reference computed with Pathtracing with 10M samples per pixels.

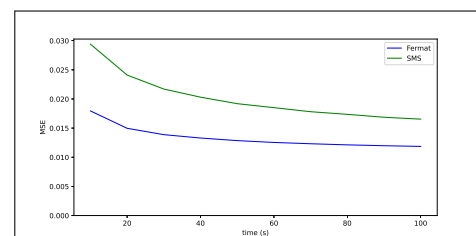


Figure 10: Convergence rate comparison of the two solvers (ours in blue, SMS in green). Using the scene from Fig.9, we compare images computed using each solver to the reference image at various rendering times and report the corresponding MSE error measure.

Our Fermat-based solver achieves noticeably faster convergence on all deformation profiles of the benchmark scene (Fig. 9 and Fig. 10). Both solvers are prone to producing outliers that would need to be eliminated in post treatment, or with a robust Monte Carlo estimator [ZHD18] [BDR21].

Solver	h_{eval}	Solutions	Success
FNEE	956M	76832	3.76%
SMS	8457M	48754	1.87%

Table 1: Comparison of the two solvers for various performance metrics (same scene as Fig. 9 with image size of 256×256 px and 100 spp): number of elevation function evaluations (in millions), number of unique solutions discovered, solver success probability (ratio of the number of solver calls that converges to a solution over the total number of calls).

More in depth comparisons (Tab. 1) reveals that the rendering of caustics produced by thin glass panels is particularly challenging for the manifold walk solver used by SMS. Our formulation displays an around 8 time

smaller number of elevation function calls and a higher success probability (we compute the success probability for each solver as the ratio of the number of solver calls that converges to a solution over the total number of calls registered). This allows FNEE to explore the solution space more efficiently than SMS (larger number of solutions discovered).

Initialisation method comparison

Here, we compare our initialisation strategy to the random specular shape sampling used by SMS on a restitution scene (work in progress) from the Digital Field of Cloth of Gold project. This indoor scene (Fig. 11) is lighted by a distant spherical light source simulating sunlight. The sun shines through many old glass windows, producing characteristic caustics.

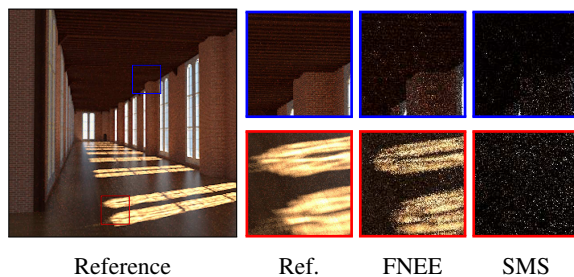
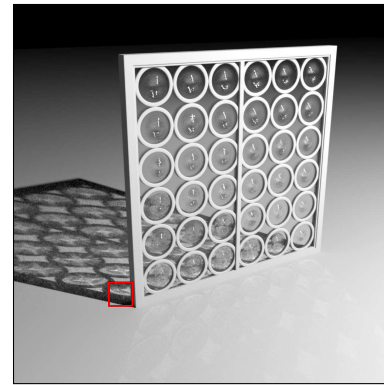


Figure 11: Visual comparison of our method with SMS (equal time $t = 100s$). Reference uses pathtracing with 30M samples per pixel.

The large number of specular interfaces makes the random pairing initialisation particularly inefficient. In contrast, the convergence speed of the proposed method remains practically constant with the number of windows used. This particularly highlights the need for a robust initialisation strategy for these types of scenes.

5 DISCUSSION

Both *FNEE* and *SMS* share the same difficulties dealing with glass panels with intricate, high frequency details, where the invPDF estimation process becomes performance intensive. Using the above parameters ($N_{newton} = 20$, $\nabla L_{\epsilon} = 10^{-4}$, $t_{min} = 0.004$) with finely detailed profiles leads to unnaturally dark caustics (Fig. 12).



Reference (Pathtracing 1M spp)



Ref. $N_{max} = 10k$ $N_{max} = 1M$
(10k sec) (100k sec)

Figure 12: invPDF under-estimation in the case of a finely detailed glass profile (using our method *FNEE*).

Understandably, as the surface gets more detailed, the generalized geometric term $G(O \leftrightarrow S)$ tends to become increasingly small in certain area (being directly dependent to the partial derivatives of the normal vector $\frac{\partial n}{\partial u}$ and $\frac{\partial n}{\partial v}$). As the ray contribution is proportional to the product $G(O \leftrightarrow S) \cdot invPDF(\bar{X})$, if the allowed number of iterations N_{max} during the invPDF estimation process is too low, the invPDF term generally won't be large enough to compensate for the luminosity loss caused by $G(O \leftrightarrow S)$. This results in an underestimation of the illumination.

Since the convergence basins are generally large for smooth perturbation profiles, the estimation can be done in a few iterations. It is however not the case anymore when the perturbations are finer since the basins become increasingly small (Fig. 13) as the number of solutions dramatically increases. In this case, taking a sufficiently large N_{max} may lead to an impractically slow convergence rate for both *SMS* and *FNEE*.

6 CONCLUSION

We have presented a method to compute the transmitted caustic produced by a displaced glass panel lighted by a light source. Our method based on the Fermat's principle displays faster convergence than *SMS* and allows to handle efficiently scenes including many glass panes. These scenes are commons in architecture and represent a situation for which *SMS* initialisation scheme is inefficient. The two methods being closely related, they share the same difficulty dealing with finely detailed surfaces: since many solutions exists, the estimation of

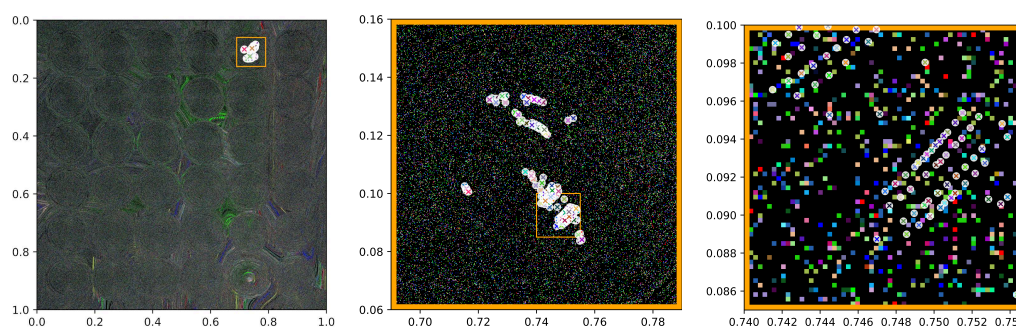


Figure 13: Convergence basins for a crown glass window. The finely detailed surface of the panel gives rise to several clusters of solutions. Each individual solution has a small and irregular convergence basin, making the invPDF estimation process inefficient.

the invPDF term requires many iterations to converge. Finding a better way to estimate this term is an avenue for future work and would greatly benefit the two techniques. Our present work is still a simplification of the real world problem since we neglected the volume irregularities that exist in real ancient glass panels. Finding ways to take these irregularities into account would be a natural direction for future research.

7 ACKNOWLEDGMENTS

We would like to thank the Hauts-de-France region, which is helping to finance this work through Quentin Huan's PhD grant. The Digital Field of Cloth of Gold is a project of the Université Lille Nord-Europe granted by the I-SITE ULNE foundation.

REFERENCES

- [Arm66] Larry Armijo. "Minimization of functions having Lipschitz continuous first partial derivatives". In: *Pacific Journal of Mathematics* 16.1 (Jan. 1966), pp. 1–3. ISSN: 0030-8730, 0030-8730. DOI: 10.2140/pjm.1966.16.1.
- [BDR21] Jérôme Buisine, Samuel Delepoulle, and Christophe Renaud. "Firefly Removal in Monte Carlo Rendering with Adaptive Median of means". In: *Eurographics Symposium on Rendering - DL-only Track* (2021). Artwork Size: 12 pages ISBN: 9783038681571 Publisher: The Eurographics Association Version Number: 121-132, 12 pages. ISSN: 1727-3463. DOI: 10.2312/SR.20211296.
- [GNL20] Lars Oliver Grobe, Andreas Noback, and Franziska Lang. "Data-Driven Modelling of Daylight Scattering by Roman Window Glass". In: *Journal on Computing and Cultural Heritage* 13.1 (Feb. 2020), pp. 1–20. ISSN: 1556-4673, 1556-4711. DOI: 10.1145/3350428.
- [Har96] John C. Hart. "Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces". In: *The Visual Computer* 12.10 (Dec. 1996), pp. 527–545. ISSN: 01782789. DOI: 10.1007/s003710050084.
- [HDF15] Johannes Hanika, Marc Droske, and Luca Fascione. "Manifold Next Event Estimation". In: *Computer Graphics Forum* 34.4 (July 2015), pp. 87–97. ISSN: 0167-7055, 1467-8659. DOI: 10.1111/cgf.12681.
- [HJ09] Toshiya Hachisuka and Henrik Wann Jensen. "Stochastic progressive photon mapping". In: *ACM SIGGRAPH Asia 2009 papers on - SIGGRAPH Asia '09*. Yokohama, Japan: ACM Press, 2009, p. 1. ISBN: 978-1-60558-858-2. DOI: 10.1145/1661412.1618487.
- [HSS01] John Hubbard, Dierk Schleicher, and Scott Sutherland. "How to find all roots of complex polynomials by Newton's method". In: *Inventiones mathematicae* 146.1 (Oct. 2001), pp. 1–33. ISSN: 0020-9910, 1432-1297. DOI: 10.1007/s002220100149.
- [Jen96] Henrik Wann Jensen. "Global Illumination using Photon Maps". In: *Rendering Techniques '96*. Ed. by Xavier Pueyo and Peter Schröder. Series Title: Eurographics. Vienna: Springer Vienna, 1996, pp. 21–30. ISBN: 978-3-211-82883-0 978-3-7091-7484-5. DOI: 10.1007/978-3-7091-7484-5_3.
- [JM12] Wenzel Jakob and Steve Marschner. "Manifold Exploration: A Markov Chain Monte Carlo Technique for Rendering Scenes with Difficult Specular Transport". In: *ACM Trans. Graph.* 31.4 (July 2012). Place: New York, NY, USA

- Publisher: Association for Computing Machinery. ISSN: 0730-0301. DOI: 10.1145/2185520.2185554.
- [Jr+09] Joseph T. Kider Jr. et al. "Recreating Early Islamic Glass Lamp Lighting". In: *VAST: International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*. Ed. by Kurt Debattista et al. ISSN: 1811-864X. The Eurographics Association, 2009. ISBN: 978-3-905674-18-7. DOI: 10.2312/VAST/VAST09/033-040.
- [Kaj86] James T. Kajiya. "The rendering equation". In: *ACM SIGGRAPH Computer Graphics* 20.4 (Aug. 1986), pp. 143–150. ISSN: 0097-8930. DOI: 10.1145/15886.15902.
- [KHD14] Anton S. Kaplanyan, Johannes Hanika, and Carsten Dachsbacher. "The natural-constraint representation of the path space for efficient light transport simulation". In: *ACM Transactions on Graphics* 33.4 (July 2014), pp. 1–13. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/2601097.2601108.
- [LW93] Eric P. Lafortune and Yves D. Willems. "Bi-directional path tracing". In: *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*. Alvor, Portugal, Dec. 1993, pp. 145–153.
- [MH92] Don Mitchell and Pat Hanrahan. "Illumination from Curved Reflectors". In: *SIGGRAPH Comput. Graph.* 26.2 (July 1992). Place: New York, NY, USA Publisher: Association for Computing Machinery, pp. 283–291. ISSN: 0097-8930. DOI: 10.1145/142920.134082.
- [Nim+19] Merlin Nimier-David et al. "Mitsuba 2: a retargetable forward and inverse renderer". In: *ACM Transactions on Graphics* 38.6 (Dec. 2019), pp. 1–17. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3355089.3356498.
- [Shi90] Peter Shirley. "A ray tracing method for illumination calculation in diffuse-specular scenes". In: *Proceedings of Graphics Interface '90*. GI 1990. ISSN: 0713-5424 event-place: Halifax, Nova Scotia, Canada. Toronto, Ontario, Canada: Canadian Information Processing Society, 1990, pp. 205–212. DOI: 10.20380/GI1990.25.
- [STN87] Mikio Shinya, T. Takahashi, and Seiichiro Naito. "Principles and Applications of Pencil Tracing". In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '87*. New York, NY, USA: Association for Computing Machinery, 1987, pp. 45–54. ISBN: 0-89791-227-6. DOI: 10.1145/37401.37408.
- [SWZ96] Peter Shirley, Changyaw Wang, and Kurt Zimmerman. "Monte Carlo techniques for direct lighting calculations". In: *ACM Transactions on Graphics* 15.1 (Jan. 1996), pp. 1–36. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/226150.226151.
- [VG97] Eric Veach and Leonidas J. Guibas. "Metropolis light transport". In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*. Not Known: ACM Press, 1997, pp. 65–76. ISBN: 978-0-89791-896-1. DOI: 10.1145/258734.258775.
- [WHY20] Beibei Wang, Miloš Hašan, and Ling-Qi Yan. "Path cuts: efficient rendering of pure specular light transport". In: *ACM Transactions on Graphics* 39.6 (Dec. 2020), pp. 1–12. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3414685.3417792.
- [ZGJ20] Tizian Zeltner, Iliyan Georgiev, and Wenzel Jakob. "Specular manifold sampling for rendering high-frequency caustics and glints". In: *ACM Transactions on Graphics* 39.4 (Aug. 2020). ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3386569.3392408.
- [ZHD18] Tobias Zirr, Johannes Hanika, and Carsten Dachsbacher. "Re-Weighting Firefly Samples for Improved Finite-Sample Monte Carlo Estimates". In: *Computer Graphics Forum* 37.6 (2018), pp. 410–421. DOI: <https://doi.org/10.1111/cgf.13335>.

Versatile input view selection for efficient immersive video transmission

Dominika Klóska Adrian Dziembowski Jarosław Samelak
dominika.kloska@put.poznan.pl adrian.dziembowski@put.poznan.pl jaroslaw.samelak@gmail.com

Institute of Multimedia Telecommunications, Poznań University of Technology
Polanka 3, 61-131 Poznań, Poland

ABSTRACT

In this paper we deal with the problem of the optimal selection of input views, which are transmitted within an immersive video bitstream. Due to limited bitrate and pixel rate, only a subset of input views available on the encoder side can be fully transmitted to the decoder. Remaining views are – in the simplest approach – omitted or – in the newest immersive video encoding standard (MPEG immersive video, MIV) – pruned in order to remove less important information. Selecting proper views for transmission is crucial in terms of the quality of immersive video system user's experience. In the paper we have analyzed which input views have to be selected for providing the best possible quality of virtual views, independently on the viewport requested by the viewer. Moreover, we have proposed an algorithm, which takes into account a non-uniform probability of user's viewing direction, allowing for the increase of the subjective quality of virtual navigation for omnidirectional content.

Keywords

Immersive video, virtual view synthesis, MPEG immersive video (MIV)

1. INTRODUCTION

A natural consequence of rapidly growing interest in immersive video and virtual reality (VR) is the demand for efficient and versatile immersive media transmission. The virtual reality technology allows the user for immersing into the scene captured by a multicamera system and virtually navigating within it (Fig. 1). Such a navigation may be restricted to several degrees of freedom (DoF). For instance, 3DoF systems allow users to rotate their head around a single pivot point, and 3DoF+ systems additionally support restricted, translational movement of user's head [MPEG19], increasing the quality of experience (QoE) when using the head-mounted display (HMD) devices. The latest, most advanced systems – 6DoF – allow users for free, unrestricted navigation within a scene [MPEG17].

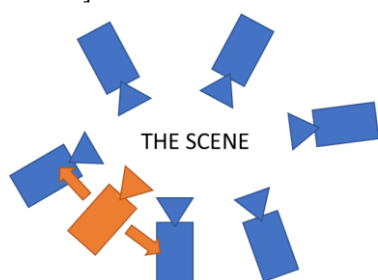


Figure 1. Idea of an immersive video system; the scene is captured by several cameras (blue), a viewer may virtually change their viewpoint (orange camera).

In order to obtain an immersive video sequence, it is required to use a multicamera system, containing even hundreds of cameras [Fuj06]. Practical systems

contain less cameras (e.g., 10 – 20 [Sta18]), but even in such a case a tremendous amount of data has to be processed and transmitted to the viewer. Moreover, the possibility of virtual immersion into the scene in the immersive video systems is provided by rendering [Fac18], [Sta22] of viewports demanded by the viewer. Such an operation requires information of the three-dimensional scene, which is typically represented in the MVD format (multiview video plus depth, Fig. 2) [Mul18]. Therefore, for each input view also a depth map should be transmitted.



Figure 2. Sequence in MVD format.

The easiest way to address the problem of transmission of a huge amount of multiview video data would be to encode each real view (e.g., using HEVC [Sul12]) and the corresponding depth map separately – such an approach is called multiview simulcast. However, this method is not effective due to the high bitrate and pixel rate [Boy21]. Moreover, the quality of the immersive content is not satisfactory because HEVC (or any typical 2D video encoder such as the newest, VVC [Bro21]) encoder was not developed for processing depth maps. It is possible to enhance the quality of the final immersive

vision with the use of MV-HEVC and 3D-HEVC [Tec16], which are HEVC extensions dedicated to encoding 3D content. However, these methods do not guarantee versatility, because they are not adapted for encoding sequences acquired by omnidirectional cameras, or by multicamera systems where the cameras are located arbitrarily. Therefore, none of the abovementioned methods can be used in practical immersive video systems.

The simplest practical solution allowing for a significant decrease of pixel rate is to transmit only a subset of the given real views. In such an approach, in order to obtain the best possible quality, these views have to be carefully chosen.

A more sophisticated, newest approach is based on the use of the MPEG immersive video (MIV) standard [Boy21], [ISO22] which defines the compression of immersive media in a form of multiview video pre- and post-processing combined with the typical video encoder, e.g., VVC [Bro21]. The MIV encoding process can be divided into three main steps, in which the input data (n views and corresponding depth maps) are processed into k video bitstreams called “atlases”, further encoded using the VVC encoder (Fig. 3).

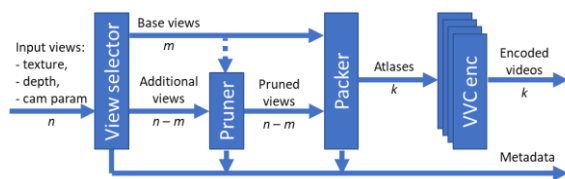


Figure 3. Simplified scheme of the MIV encoder.
Figure from [Dzi22a].

In the first step the MIV decides which views are the most important from the user’s point of view. These views are then being labeled as “base views” and are being placed in atlases in their entirety (Fig. 4A and C). Remaining views (“additional views”) contain a lot of redundant data are then pruned in order to remove the excess data. Finally, after the pruning operation additional views are packed into atlases as a form of patch mosaic (Fig. 4B and D) [Vad22].

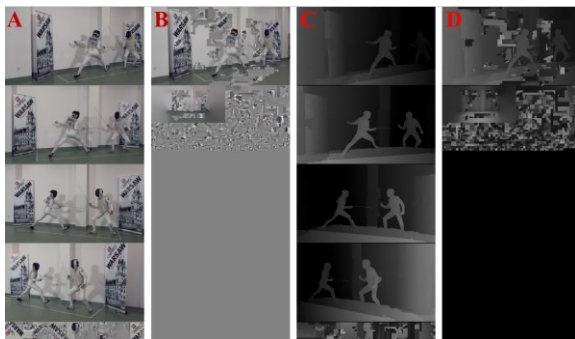


Figure 4. Four atlases produced by the MIV encoder using the MIV Main profile [Vad22]: 2 texture atlases (A, B) and 2 depth atlases (C, D).

On the decoder side, the atlases are firstly decoded using the typical video decoder (such as VVC). After the video decoding step, the views and depths stored in atlases are unpacked and then used for rendering of the views requested by user (Fig. 5).

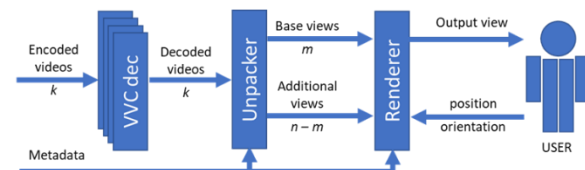


Figure 5. Simplified scheme of the MIV decoder.
Figure from [Dzi22a].

Irrespective of the immersive video coding approach, the effective input view selection is the crucial step in terms of providing the best quality and the highest coding efficiency. In this paper, we describe the view selection method which allows for efficient immersive video transmission in practical systems, where it is not possible to send all the real views to the decoder. Moreover, proposed algorithm performs efficiently both for content acquired by perspective and omnidirectional cameras, and can be used for 6DoF systems, where the user virtually immerses into the scene [Laf17].

2. INPUT VIEW SELECTION

2.1. View selection for virtual view synthesis

The proper input view selection method should provide the highest possible quality of synthesized views while preserving similar bitrate. Considerations on the influence of input view selection on the virtual view quality were described by the authors of this paper in [Dzi18], where we focused on optimizing the quality in simple free navigation systems [Sta18]. In [Dzi18], we assumed that the renderer has access to all of input views, but - in order to provide reasonable computational time - it can use only two of them for rendering purposes.

The input view choice requires addressing three problems: occlusions, finite resolution of video, and non-Lambertian surfaces; leading to the conclusion that the highest quality of rendered views can be obtained based on nearest left and nearest right input view. Obviously, in such a scenario it would be optimal to transmit these two views and skip all the others. However, a selection of these two views is possible only if the position of view requested by the viewer is known before the transmission.

2.2. View selection for immersive video transmission

In a practical immersive video system, where multiple viewers receive the same bitstream and are able to independently choose their point of view [Tan12], an assumption regarding viewer’s position known *a priori* before the transmission is invalid. Instead, it is required to choose input views in the

way, which guarantees the highest average quality of views watched by users, independently of their viewpoint.

In order to meet the requirements for immersive data transmission, where the position of a user cannot be predicted, the view selection method described in [Dzi18] has to be extended.

Taking into account the statement that the quality of synthesized view is highest when the rendering is performed on the basis of the nearest left and right real view, we have conducted a simulation. In the simulation we assumed a simple practical immersive video system with reasonable pixel rate [Boy21] and number of cameras [Sal18]:

- linear multicamera system with 13 evenly distributed cameras,
- 13 input views available at the encoder side,
- 4 input views transmitted to the decoder,
- 100 possible virtual positions of the viewer (evenly distributed too).

We assumed that the left-most and right-most input views are transmitted (in order to ensure, that for all virtual positions of the viewer there exists left and right input view). Therefore, the index of the first transmitted view was fixed to 1, and index of the fourth view was fixed to 13. Indices of remaining two input views to be transmitted were unknown, and they were iteratively changed in order to calculate their optimal position.

For each virtual position of the viewer, we calculated the total distance to the nearest left and nearest right view. The results are presented in Fig. 6, where the horizontal axis presents the index of the first real view used for virtual view synthesis, the vertical axis presents the index of the second real view.

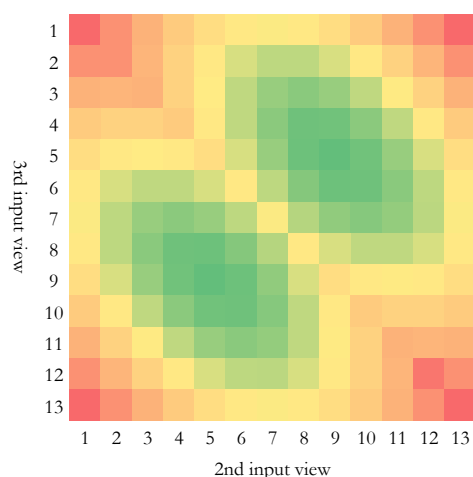


Figure 6. Total distance between nearest left and nearest right view calculated for all real views used for the experiment.

The green color in Fig. 6 indicates that the total distance between the virtual view and its transmitted

neighbors was low, red – that the distance was higher. These results show that the distance is minimized when the input views selected to be transmitted are distributed evenly. Of course, in Fig. 6 we presented only the distance measured for a simple simulation, not the quality of synthesized views. Therefore, in the next section we presented evidence for these considerations.

It should be noted that the presented model and experiment assumed a simple, linear camera arrangement. However, analogous conclusions can be taken also for more sophisticated multicamera systems.

3. EXPERIMENTAL RESULTS

3.1. Methodology

In order to prove authenticity of considerations presented in the previous section, we have performed an experiment. In the experiment, we assessed the quality of virtual views synthesized using various combinations of input views. The view synthesis was performed using the MPEG's reference software – VVS [Dzi19].

The test set comprised of two computer-generated sequences – BBB Butterfly and BBB Flowers [Kov15]. We have decided to use these sequences, as they contain multiple (79) input views, making possible quality assessment for several viewpoints. Both sequences were captured by 79 evenly-distributed cameras placed on an arc. For each sequence, views are numbered from v6 to v84. Seven views: v6, v19, v32, v45, v58, v71, and v84 were used as input ones, while all remaining views were treated as reference for objective quality evaluation.

The quality of synthesized views was calculated using two objective quality metrics, described in the MIV Common Test Conditions (MIV CTC) [MPEG22c] and commonly used in the experiments related to immersive video: WS-PSNR [Sun17] and IV-PSNR [Dzi22b]. Both quality metrics are full-reference ones, therefore the quality was assessed by comparing input views with virtual views synthesized in the same position (the same viewpoint).

In the experiment we assumed the transmission of four input views (of the seven available). Two input views were fixed: v6 as the first input view and v84 as the fourth one. The position of second and third input views was being changed in order to define the optimal arrangement of transmitted views.

3.2. Results

Mean IV-PSNR and WS-PSNR of synthesized virtual views are presented in Tables 1 and 2. The values were averaged over 75 synthesized views (v6 to v84, excluding four views used as input ones, for

which the quality is perfect, as no synthesis is needed).

3rd input view	v19		31.32	32.06	32.32	31.34
	v32	31.32		32.10	33.16	32.57
	v45	32.06	32.10		31.78	31.59
	v58	32.32	33.16	31.78		30.74
	v71	31.34	32.57	31.59	30.74	
2nd input view						
3rd input view	v19		39.39	41.08	41.03	39.44
	v32	39.39		40.78	41.65	40.36
	v45	41.08	40.78		40.04	39.79
	v58	41.03	41.65	40.04		38.05
	v71	39.44	40.36	39.79	38.05	
2nd input view						

Table 1. Mean IV-PSNR [dB] of virtual view (averaged over 75 views) calculated for different combinations of transmitted input views. 1st input view was set to v6, 4th input view: v84. Sequences: BBB Flowers (top) and BBB Butterfly (bottom).

3rd input view	v19		24.12	24.91	24.99	23.99
	v32	24.12		24.98	25.68	25.03
	v45	24.91	24.98		24.64	24.42
	v58	24.99	25.68	24.64		23.57
	v71	23.99	25.03	24.42	23.57	
2nd input view						
3rd input view	v19		31.99	33.05	32.91	31.80
	v32	31.99		32.90	33.52	32.68
	v45	33.05	32.90		32.40	32.26
	v58	32.91	33.52	32.40		30.91
	v71	31.80	32.68	32.26	30.91	
2nd input view						

Table 2. Mean WS-PSNR [dB] of virtual view (averaged over 75 views) calculated for different combinations of transmitted input views. 1st input view was set to v6, 4th input view: v84. Sequences: BBB Flowers (top) and BBB Butterfly (bottom).

As presented, in all considered scenarios (both sequences and both quality metrics), the best average quality of synthesized views can be achieved when using views v6, v32, v58, and v84, thus evenly distributed input views.

Such a view selection provides also highest quality in a worst-case scenario (the lowest quality among all synthesized views, Table 3).

Moreover, even distribution of transmitted input views minimizes Δ IV-PSNR (difference between lowest and highest quality among all synthesized views, Table 4), making the user's experience more stable, as the perceived quality change during virtual navigation among the scene is lower.

Tables 3 and 4 present only the results obtained for the IV-PSNR metric. The WS-PSNR results were omitted, as it behaves similarly, and the best results were achieved for evenly distributed input views.

3rd input view	v19		22.95	26.94	23.50	23.40
	v32	22.95		26.94	29.72	25.72
	v45	26.94	26.94		24.41	24.17
	v58	23.50	29.72	24.41		21.41
	v71	23.40	25.72	24.17	21.41	
2nd input view						
3rd input view	v19		31.54	36.28	35.08	32.28
	v32	31.54		36.15	38.36	34.95
	v45	36.28	36.15		33.75	33.38
	v58	35.08	38.36	33.75		30.38
	v71	32.28	34.95	33.38	30.38	
2nd input view						

Table 3. Lowest IV-PSNR [dB] of virtual view (among 75 views) calculated for different combinations of transmitted input views. 1st input view was set to v6, 4th input view: v84. Sequences: BBB Flowers (top) and BBB Butterfly (bottom).

3rd input view	v19		18.99	14.99	18.74	21.14
	v32	18.99		14.99	12.52	18.82
	v45	14.99	14.99		17.83	20.37
	v58	18.74	12.52	17.83		23.15
	v71	21.14	18.82	20.37	23.15	
2nd input view						
3rd input view	v19		14.06	9.91	10.12	12.67
	v32	14.06		10.16	7.83	9.94
	v45	9.91	10.16		13.35	12.49
	v58	10.12	7.83	13.35		16.20
	v71	12.67	9.94	12.49	16.20	
2nd input view						

Table 4. Δ IV-PSNR [dB] of virtual view (among 75 views) calculated for different combinations of transmitted input views. 1st input view was set to v6, 4th input view: v84. Sequences: BBB Flowers (top) and BBB Butterfly (bottom).

4. INPUT VIEW SELECTION FOR OMNIDIRECTIONAL CONTENT

4.1. Omnidirectional content problem

All the considerations presented in previous sections assumed that the viewer can watch the scene from any viewpoint, and the probability of choosing various viewpoints is the same. However, it is not true for 6DoF and 3DoF+ [Wie19] immersive video systems, where the user virtually immerses into the scene, e.g., using the HMD device. In such a case, a typical user tends to look around in the horizontal

plane, while not focusing on floor, ceiling or the sky above [Dzi22c].

Therefore, the use of the described view selection algorithm, which chooses input views most distant to each other may result in non-optimal selection. For instance, views captured by cameras facing down or up may be selected instead of views containing essential information about the scene (Fig. 7).



Figure 7. Three views of the Chess sequence [Ilo19].

The example presented in Fig. 7. A and B are two of 7 views selected as “base views” by the MIV encoder (working under the decoder-side depth estimation – DSDE – configuration [Mie22]). Fig. 7.C presents a view, which was selected as an “additional view” and skipped despite having more important information from the viewer’s perspective.

Such a selection increases the quality of the floor and the ceiling but decreases the quality of a chess knight – which is more crucial for the viewer.

4.2. Proposed solution

Taking into account the subjective non-uniform significance of different areas of the scene, we proposed a modification of the simple view selection algorithm, which penalizes the vertical distance between cameras.

In the basic approach (e.g., the one implemented in the 14th version of the Test Model for MPEG immersive video – TMIV 14 [MPEG22a]), basic views were selected by maximization of the total distance between them. The distance between two views i and j was calculated as:

$$r_{i,j} = \sqrt{(r_{i,j}^x)^2 + (r_{i,j}^y)^2 + (r_{i,j}^z)^2}, \quad (1)$$

where $r_{i,j}^x$, $r_{i,j}^y$, and $r_{i,j}^z$ are distances between views i and j along three axes of the global coordinate system.

We proposed to modify (1) by addressing the non-uniform probability of viewer’s watching direction and by penalizing the vertical distance. To achieve that the camera distances calculated in the view selection process are not homogenous meaning that the vertical direction is being treated differently from horizontal directions:

$$r_{i,j} = \sqrt{(r_{i,j}^x)^2 + (r_{i,j}^y)^2 + (w \cdot r_{i,j}^z)^2}, \quad (2)$$

Where r^x, r^y, r^z indicate a distance between two cameras among three axes, and where w is the inhomogeneity coefficient. In the experiments described in the further part of this section, the w value was set to 0.4.

The proposed change allows for selecting input views, which carry valuable information (Fig. 8.B) instead of sending views containing plain floor or ceiling of the scene, irrelevant for the viewer (Fig. 8).

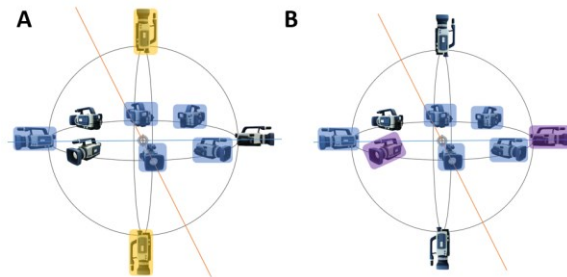


Figure 8. Sequence Chess. A – views selected for encoding by the anchor method (blue color highlights selected cameras that carry valuable information about the scene, yellow shows selected cameras that have less importance to the viewer and therefore can be omitted), B – views selected for encoding after the modification (purple color highlights cameras that were selected instead of the yellow cameras from Fig 8.A).

Cameras highlighted in blue were selected as base views for both basic and modified view selection algorithms. Besides them, the basic algorithm selected cameras facing up and down (yellow cameras in Fig. 8), while the modified algorithm – two cameras acquiring important parts of the scene. Considering the fact, that a typical viewer spends more time looking around on the horizontal plane rather than the vertical one (which contains the floor and the ceiling) [Dzi22c], we propose to send more views from the horizontal plane instead of the views facing upwards and downwards. It will have a positive influence on the final quality of the particular parts of the scene at which the user looks the majority of the time.

The proposed modification was appreciated by the experts of the ISO/IEC JTC1/SC29/WG 04 MPEG VC group and is included in the newest version of the Test Model for MIV – TMIV 15 [MPEG22b].

The influence of this view selection modification on the objective and subjective quality of the final immersive vision was described in the following subsections.

4.3. Methodology of the experiment

The experiment was conducted under the common test conditions for MPEG immersive video (MIV CTC) [MPEG22c], but the test set was limited to omnidirectional sequences only (fig 9).

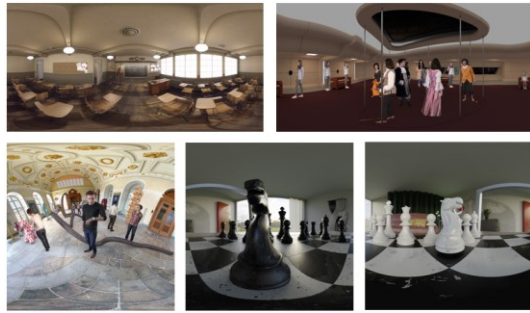


Figure 9. Ominidirectional sequences used for the experimental results, sequences: ClassroomVideo [Kro18], Hijack [Dor18], Museum [Dor18], Chess [Ilo19], and ChessPieces [Ilo20].

In the experiment, the TMIV 14 software [MPEG22a] was used. Each sequence was compressed with the use of five different rate points (RP) using the VVC encoder. The total bitrate for a sequence ranged from 2.5 Mbps (RP5) to over 20 Mbps (RP1). The quality of the synthesized virtual views was assessed using WS-PSNR and IV-PSNR objective quality metrics [Sun17], [Dzi22b].

In order to perform a thorough test, three configurations of TMIV were evaluated: MIV, MIV View and MIV DSDE. The detailed description of these configurations can be found in the publicly available MIV CTC document [MPEG22c].

Besides the objective quality measurement, also the subjective quality of rendered views was evaluated. The subjective quality assessment was performed based on pose traces [Boy21], according to the MIV CTC.

The subjective quality evaluation was done by 45 naïve viewers, watching two side-by-side videos (Pair Comparison method, Rec. ITU-T P.910 [ITU08] and ITU-R BT.500 [ITU98]). The viewers judged the quality of presented posetraces with the use of 7-number scale with values from -3 to 3.

To minimize the time duration of the subjective test, only three RP were shown to the viewers: RP1, RP3, and RP5. Moreover, subjects were assessing quality change only for sequences, for which the view selection result was different, than for unmodified TMIV14 (see Table 5).

Sequence Name \ MIV Configuration	MIV Main	MIV View	MIV DSDE
Classroom	x	x	x
Museum	✓	✓	✓
Hijack	x	x	✓
Chess	x	x	✓
ChessPieces	x	x	✓

Table 5. Overview of the sequences and different MIV configurations. „X” indicates the scenario in which view selection result was the same as with the unmodified TMIV 14 software.

a) Subjective quality evaluation

The results of performed subjective quality evaluation are presented in Figs. 10 and 11. In Fig. 10, an influence of the proposed method on efficiency of different MIV configurations are presented. Fig. 11 contains comparison of subjective quality change for different test sequences. The results are presented as an average quality change caused by the proposed modification and the 95% confidence interval, calculated according to ITU-R recommendations [ITU98] as:

$$CI = 1.96 \cdot \frac{SD}{\sqrt{N}}, \quad (3)$$

where CI is the confidence interval, SD – standard deviation, and N – number of viewers (in presented experiment N = 45).

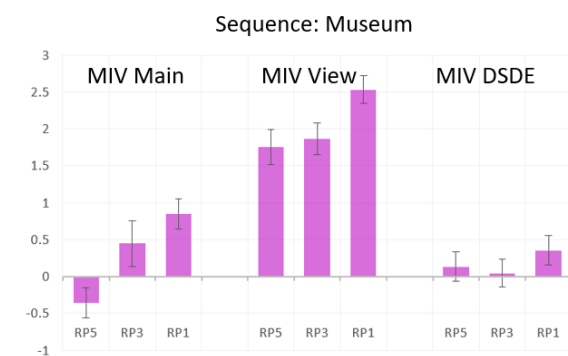


Figure 10. Subjective results for the Museum sequence in three different MIV configurations.

Subjective quality changes presented for the Museum sequence in Fig. 10 show that in almost every scenario there was a visible quality improvement. For 6 of 9 tests, the proposal allowed for achieving a statistically important quality improvement. For two tests (RP5 and RP3 in MIV DSDE configuration) the quality gain was also spotted, but it was not statistically important.

The proposal decreased the subjective quality in only one case – the heaviest compression in the MIV Main scenario. However, as presented in Fig. 12, for such a low bitrate the MIV Main cannot properly handle Museum sequence, and the quality of the content was unsatisfactory also before proposed modification.

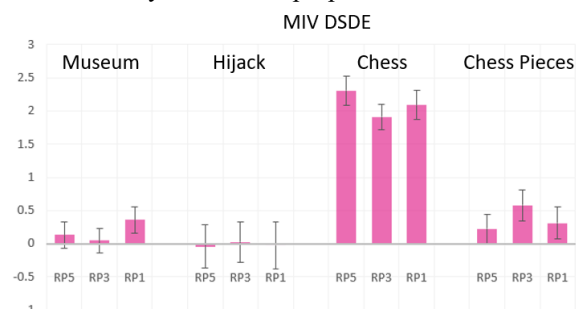


Figure 11. Subjective results for four sequences in MIV DSDE configuration.

Analysis of the results obtained for different sequences in the MIV DSDE configuration (Fig. 11) is similar to the results presented in Fig. 10. For 3 of 4 test sequences there is a quality gain induced by the proposed view selection modification. Moreover, for these sequences this gain is statistically important in 7 of 9 cases.

The only sequence for which the quality change did not occur is Hijack, but it should be noted that the proposal did not decrease the subjective quality.

b) Objective results

Objective quality results are gathered in Tables BDR and BDP, and presented as Bjøntegaard deltas [Bjo01]: BD-rate (Table 6) and BD-PSNR (Table 7).

Sequence		High-BR BD rate	Low-BR BD rate	High-BR BD rate	Low-BR BD rate
		WS-PSNR	WS-PSNR	IV-PSNR	IV-PSNR
Museum	MIV	-3.4%	0.2%	0.9%	2.3%
	MIV View	-4.6%	-5.5%	-7.7%	-3.2%
MIV DSDE		57.2%	32.5%	13.1%	10.2%
	Hijack	---	---	---	---
	Chess	---	---	---	---
	ChessPieces	---	---	---	---

Table 6. Objective metric (WS-PSNR and IV-PSNR) BD-rates obtained for 4 lowest rate points (Low-BR) and for 4 highest rate points (High-BR); “---” denotes, that the BD-rate calculation was not possible because of non-overlapping curves.

Sequence		High-BR BD rate	Low-BR BD rate	High-BR BD rate	Low-BR BD rate
		WS-PSNR	WS-PSNR	IV-PSNR	IV-PSNR
Museum	MIV	0.4%	0.1%	-0.0%	-0.2%
	MIV View	0.2%	0.3%	0.8%	0.3%
MIV DSDE		-1.6%	-1.3%	-0.8%	-0.6%
	Hijack	-13.2%	-13.2%	-11.1%	-11.1%
	Chess	-6.4%	-6.2%	-5.9%	-5.5%
	ChessPieces	-11.3%	-11.0%	-8.3%	-7.9%

Table 7. Objective metric (WS-PSNR and IV-PSNR) BD-PSNRs obtained for 4 lowest rate points (Low-BR) and for 4 highest rate points (High-BR).

Surprisingly, presented objective results show, that in general the proposed method performs worse than the basic view selection algorithm implemented in TMIV 14. However, it has to be highlighted that results presented in Tables 6 and 7 were obtained by averaging the IV-PSNR and WS-PSNR values of all synthesized views, including the basic views. An example is shown in Table 8, where exact IV-PSNR values for all 10 views of sequence Chess are presented.

As presented in Table 8, the average IV-PSNR for non-base (i.e., “additional”) views is similar to the quality obtained for unmodified TMIV14. The only views with significant quality degradation are v0 and

v9 (i.e., views captured by cameras facing up and down), which are less important to the viewer.

View	IV-PSNR [dB]		
	TMIV 14	Proposed	delta
v0	57.00	36.17	-20.84
v1	37.84	37.99	0.15
v2	56.78	56.76	-0.02
v3	33.20	56.28	23.09
v4	56.73	56.44	-0.29
v5	56.01	55.93	-0.08
v6	38.85	56.11	17.26
v7	56.02	56.04	0.03
v8	57.08	57.18	0.10
v9	56.34	30.08	-26.26
Average (all views)			-0.69
Average (only non-base views)			-0.02

Table 8. IV-PSNR of synthesized views, RP1, similar bitrate for both approaches (21.3 Mbps for TMIV 14 and 20.8 Mbps for proposed); Chess sequence, MIV DSDE configuration.

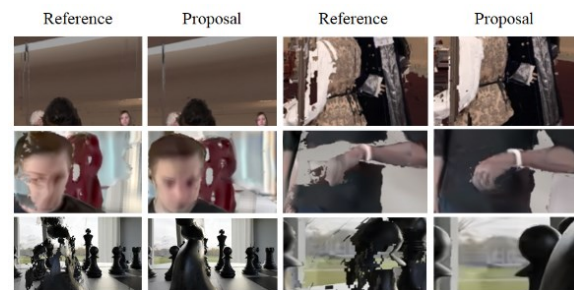


Figure 12. Visual comparison of posetraces generated with the use of original (reference) and proposed view selection method; sequences (from top): Hijack, Museum, and Chess.

5. CONCLUSIONS

This paper deals with problems on view selection for immersive video and its influence on the quality of final immersive vision on the decoder side.

Firstly, we have conducted an experiment to assess which views from multiview sequences should be selected into the virtual view synthesis process in order to obtain the best quality possible. Received results proved that the view selection algorithm should select views that are evenly distributed.

Moreover, in the paper we proposed an algorithm, which increases the subjective quality of virtual navigation by taking into account a non-uniform probability of choosing the viewing direction. We have noticed, that a typical user usually chooses to watch the scene in the horizontal plane, while the top and bottom parts of the omnidirectional scene are less important. This proposal was appreciated by the ISO/IEC MPEG VC experts, and is included in the reference software [MPEG22b] for the MPEG immersive video coding standard [ISO22].

The proposed approach is based on observations on the behavior of a typical user, without thorough

statistical analysis. Moreover, a correlation between optimal view selection and scene characteristics should be taken into consideration. Therefore, the topic of view selection for immersive video transmission will be studied further in our future research.

6. ACKNOWLEDGMENTS

This work was supported by the Ministry of Education and Science of Republic of Poland.

7. REFERENCES

- [Bjo01] Bjøntegaard, G. Calculation of average PSNR differences between RD986 curves. ISO/IEC JTC1/SC29/WG11 MPEG M15378, Austin, TX, 2001.
- [Bro21] Bross B. et al. Overview of the Versatile Video Coding (VVC) standard and its applications. *IEEE T. on Circuits and Systems for Video Technology* 31 (10), pp. 3736-3764, 2021.
- [Boy21] Boyce J. et al. MPEG Immersive Video coding standard. *Proceedings of the IEEE* 109 (9), pp. 1521-1536, 2021.
- [Dor18] Doré R. Technicolor 3DoF+ Test Materials. Doc. ISO/IEC JTC1/SC29/WG11 MPEG/M42349, 2018.
- [Dzi18] Dziembowski A. et al. View selection for virtual view synthesis in free navigation systems. *International Conference on Signals and Electronic Systems* 2018, Kraków, Poland, 2018.
- [Dzi19] Dziembowski A. et al. Virtual view synthesis for 3DoF+ video. *Picture Coding Symposium, PCS* 2019, 2019.
- [Dzi22a] Dziembowski A. et al. Spatiotemporal redundancy removal in immersive video coding. *Journal of WSCG*, vol. 30, no. 1-2, pp. 54-62, 2022.
- [Dzi22b] Dziembowski A. et al. IV-PSNR – the objective quality metric for immersive video applications. *IEEE T. on Circuits and Systems for Video Technology* 32 (11), pp. 7575-7591, 2022.
- [Dzi22c] Dziembowski, A., Klóska, D., Jeong, J.Y., and Lee, G. [MIV] Inhomogeneous view selection for omnidirectional content. ISO/IEC JTC1/SC29/WG4 MPEG 140, M60668, 10.2022.
- [Fac18] Fachada S. et al. Depth image based view synthesis with multiple reference views for virtual reality. *3DTV-Conf*, 2018.
- [Fuj06] Fujii T. et al. Multipoint measuring system for video and sound – 100-camera and microphone system, ICME conference, 2006.
- [Ilo19] Ilola L. et al. New test content for immersive video – Nokia Chess. Doc. ISO/IEC JTC1/SC29/WG11 MPEG, M50787, 2019.
- [Ilo20] Ilola L., Vadakital V.K.M. Improved NokiaChess sequence. Doc. ISO/IEC JTC1/SC29/WG11 MPEG, M57382, 2020.
- [ISO22] Standard ISO/IEC FDIS 23090-12. Information technology – Coded representation of immersive media – Part 12: MPEG Immersive video. 2022.
- [ITU98] Methodology for the Subjective Assessment of the Quality of Television Pictures, document Rec. ITU-R BT.500-9, ITU-R, 1998.
- [ITU08] Subjective Video Quality Assessment Methods for Multimedia Applications, document Rec. ITU-T P.910, ITU-T, 2008.
- [Kov15] Kovacs, P. BBB light-field test sequences. ISO/IEC JTC1/SC29/WG11, M35721, 2015.
- [Kro18] Kroon B. 3DoF+ test sequence ClassroomVideo, ISO/IEC JTC1/SC29/WG11 MPEG M42415, 2018.
- [Mie22] Mieloch D. et al. Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video. *IEEE T. on Circ. and Systems for Video Tech.* 32 (9), pp. 6360-6374, 2022.
- [MPEG17] Requirements on 6DoF (v1), ISO/IEC JTC1/SC29/WG11 MPEG N17073, 2017.
- [MPEG19] MPEG Call for Proposals on 3DoF+ Visual, ISO/IEC JTC1/SC29/WG11 MPEG/N18145 2019.
- [MPEG22a] Test Model 14 for MPEG immersive video, ISO/IEC JTC1/SC29/WG04 MPEG VC N0242, 2022.
- [MPEG22b] Test Model 15 for MPEG immersive video. Document ISO/IEC JTC1/SC29/WG04 MPEG VC, N0271, 2022.
- [MPEG22c] Common test conditions for MPEG immersive video. ISO/IEC JTC1/SC29/WG04 MPEG VC, N0232, 2022.
- [Mul18] Müller K. et al. 3-D Video Representation Using Depth Maps. *Proceedings of the IEEE* 99 (4), pp. 643-656, 2011.
- [Sal18] Salahieh, B. et al. Kermit test sequence for Windowed 6DoF activities. ISO/IEC JTC1/SC29/WG11 MPEG M43748, Ljubljana, Slovenia, 2018.
- [Sta18] Stankiewicz O. et al. A free-viewpoint television system for horizontal virtual navigation. *IEEE Transactions on Multimedia* 20 (8), pp. 2182-2195, 2018.
- [Sta22] Stankowski J., Dziembowski A. Real-time CPU-based view synthesis for omnidirectional video. 30. *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG* 2022, 2022.

- [Sul12] Sullivan G. et al. Overview of the High Efficiency Video Coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology* 22 (12), pp. 1649-1668, 2012.
- [Sun17] Sun Y. et al. Weighted-to-Spherically-Uniform Quality Evaluation for Omnidirectional Video. *IEEE Signal Processing Letters* 24 (9), pp. 1408-1412, 2017.
- [Tan12] Tanimoto M. et al. FTV for 3-D Spatial Communication. *Proceedings of the IEEE*, vol. 100, no. 4, pp. 905-917, 2012.
- [Tec16] Tech G. et al. Overview of the Multiview and 3D Extensions of High Efficiency Video Coding. *IEEE T. on Circuits and Systems for Video Technology* 26 (1), pp. 35-49, 2016.
- [Vad22] Vadakital V.K.M. et al. The MPEG immersive video standard – current status and future outlook. *IEEE MultiMedia* 29 (3), 2022.
- [Wie19] Wien M. et al., Standardization Status of Immersive Video Coding, *IEEE J. on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 5-17, 2019.

Massively Parallel CPU-based Virtual View Synthesis with Atomic Z-test

Jakub Stankowski

Institute of Multimedia Telecommunications
Poznań University of Technology
Polanka 3
61-131 Poznań, Poland
jakub.stankowski@put.poznan.pl

Adrian Dziembowski

Institute of Multimedia Telecommunications
Poznań University of Technology
Polanka 3
61-131 Poznań, Poland
adrian.dziembowski@put.poznan.pl

ABSTRACT

In this paper we deal with the problem of real-time virtual view synthesis, which is crucial in practical immersive video systems. The majority of existing real-time view synthesizers described in literature require using dedicated hardware. In the proposed approach, the view synthesis algorithm is implemented on a CPU increasing its usability for users equipped with consumer devices such as personal computers or laptops. The novelty of the proposed algorithm is based on the atomic z-test function, which allows for parallelization of the depth reprojection step, what was not possible in previous works. The proposal was evaluated on a test set containing miscellaneous perspective and omnidirectional sequences, both in terms of quality and computational time. The results were compared to the state-of-the-art view synthesis algorithm – RVS.

Keywords

Virtual view synthesis, immersive video systems, real-time video processing.

1. INTRODUCTION

The basic idea of an immersive video system is to allow a user for immersing into the scene by giving a possibility of free virtual navigation within a scene captured by a multicamera system [Goo12], [Sta18], equipped with perspective or omnidirectional cameras (Fig. 1). In a typical scenario, where the scene is represented using the multiview plus depth (MVD) representation [Mül11], such a possibility is provided by the synthesis (i.e., rendering) of virtual viewpoints.

There are multiple good-quality virtual view synthesis methods described in the literature, e.g., [Dzi19], [Fac18] or [Sen18]. However, these methods cannot be used in the real-time scenario, making them not suitable for practical immersive video systems, where the system's response to user's change of position should be immediate, and the virtual view should be synthesized with possibly smallest delay [Dzi18].

2. FAST VIRTUAL VIEW SYNTHESIS

In the literature, several real-time virtual view synthesis methods are described. However, the vast majority of them require dedicated hardware, such as powerful graphic cards (e.g., [Non18], [Zha17]),

FPGA devices (e.g., [Aki15], [Li19]) or even VLSI devices [Hua19].

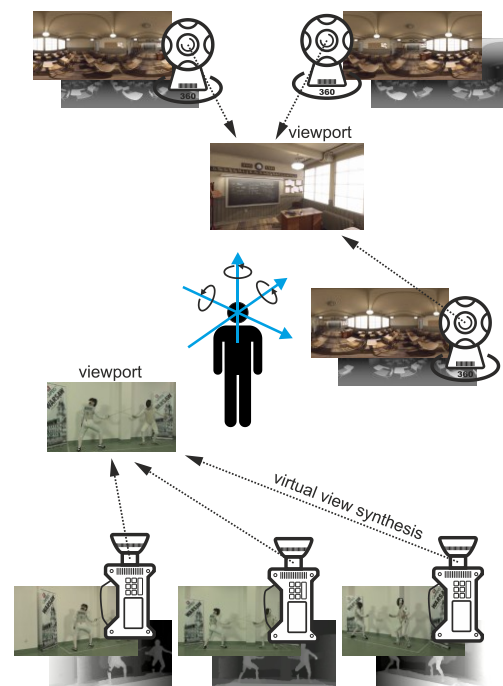


Figure 1. Idea of an immersive video system.

In this paper, we present a significant improvement of the methods we have described in [Sta20] and [Sta22], designed for real-time handling of perspective and omnidirectional content, respectively.

The general scheme of our fast CPU-based virtual view synthesis algorithm is presented in Fig. 2. It can be divided into three major steps: synthesis of the depth map corresponding to the virtual view (orange blocks in Fig. 2), synthesis of the virtual view itself (blue block), and postprocessing of synthesized view (grey blocks). The reprojection of depth and texture is performed differently for perspective views (using homography matrices [Sta20]) and omnidirectional ones (using equations described in [Sta22]). The postprocessing step is identical, independently on the video type and includes operations like filtering and inpainting.

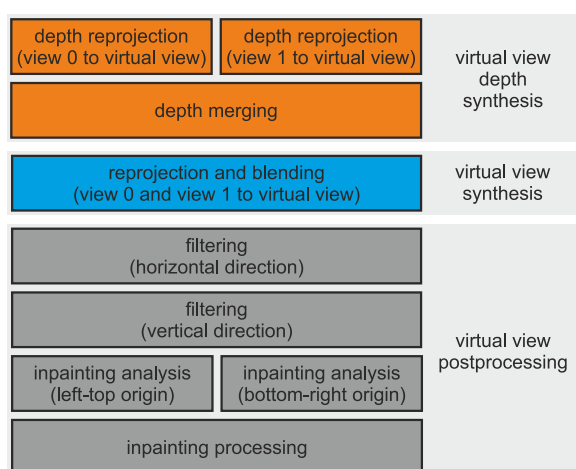


Figure 2. Overview of the fast virtual view synthesis algorithm. Figure from [Sta22].

As described in [Sta20] and [Sta22], the algorithm provides good-quality virtual views and allows to achieve real-time processing even for very high-resolution video, i.e., synthesizing of Full HD (or 2K×2K) virtual views based on two 4K input views.

The improvement presented in this paper allows to decrease this time even more, allowing for real-time synthesis of 4K sequences.

3. PARALLELIZATION LIMITATION

As described in [Sta20], the computational time of the view synthesis algorithm can be significantly decreased by using the multithread implementation and exploiting computing capabilities of modern multicore processors.

The stages related to texture reprojection and virtual view postprocessing (blue and grey blocks in Fig. 2) can be freely parallelized by dividing picture area into

rows, slices, tiles, etc. and processing each one using separate thread.

However, this simple divide-and-process approach cannot be applied to the most complex stage – depth reprojection. The location of reprojected depth is unpredictable and may induce a situation where two threads try to write data into the same memory location. This can lead to race condition and corruption of reprojected depth.

A solution for above mentioned issue was proposed in [Sta20] and further developed in [Sta22]. A technique presented in [Sta20] introduced using of the algorithm called Independent Projection Targets (IPT). When using IPT, different slices of depth (processed in separate threads) are reprojected into separate target buffers and then merged (Fig. 3). Unfortunately, the IPT algorithm has scalability limitations. Each processing thread requires a dedicated set of intermediate target buffers which leads to increased memory complexity. To make things worse, the intermediate buffers have to be merged causing the merge operation to become more time consuming when higher number of processing threads is used. The overhead related to added complexity of intermediate buffers merging offsets the gain from using higher number of depth reprojection threads.

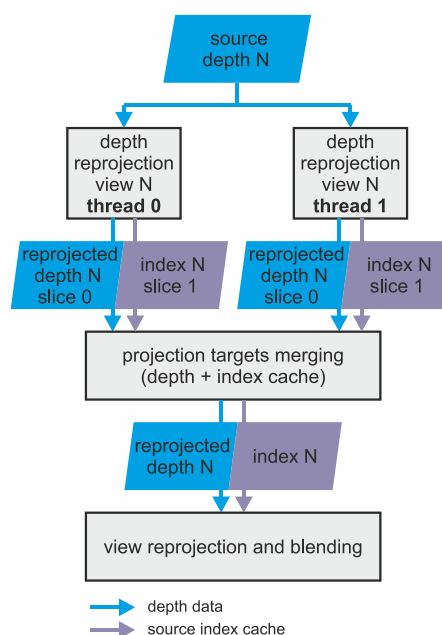


Figure 3. The idea of the independent projection targets (IPT) with data flow and intermediate data structures.

4. ATOMIC Z-TEST

In order to overcome the IPT scalability issues, we have developed a new algorithm called atomic z-test (AZT). The idea behind AZT is to use a special type

of memory access operation called “atomic” [Zhu84] to share single target buffer between processing threads in order to avoid excessive memory usage and intermediate buffers merging overhead. The atomic instruction allows to perform a single load-modify-write operation which cannot be interrupted by another core [Sch15].

In order to use the atomic z-test the data layout has to be changed. In previous implementations [Sta20], [Sta22] separate target depth and source index buffers were used. Since atomic instructions operate on single memory location, we had to combine target depth and source index buffers into single buffer. Therefore, the reprojected data buffer contains pairs of concatenated values – depth on more significant bits and index on least significant bits, both stored as 32-bit unsigned integer.

The idea behind z-test in depth reprojection is to select closest object (which corresponds to lowest depth / highest disparity value). Since one depth and index pair occupies a 64-bit memory location with depth placed on most significant bits, a 64-bit unsigned integer maximum operation can be used during z-test and depth merging. This allows us to perform z-test by updating a single 64-bit value as one atomic operation.

Different CPU (and GPU) architectures allow for a variety of atomic operations. The most common is compare-and-exchange also called compare-and-swap (CAS), however more sophisticated operations like addition, subtraction, etc., are sometimes available. In our case the desired instruction would be 64-bit atomic maximum. Unfortunately, no general-purpose CPU offers such an instruction. This leads us to necessity of simulating it by using compare-and-swap (CAS) operation. Implementation details are provided in section 5.

Fig. 4 illustrates simplified data flow in AZT algorithm (only two processing threads are drawn for clarity reasons). The presented diagram shows that AZT allows for using single reprojected depth and index buffer which allows for eliminating the time-consuming stage of depth merging.

According to [Zhu84], atomic operations are slightly slower than regular memory accesses, however the biggest performance penalty is related to a situation where two cores try to perform an atomic operation on the same location (to be precise – within the same cache line) and memory subsystem has to serialize request coming from different CPU cores. This leads us to conclusion that the performance of proposed algorithm can depend on two factors: the number of inter-thread collisions and the quality of the CPU memory subsystem implementation.

Nevertheless, the usage of AZT allows us to use all available CPU cores/threads and overcome IPT scalability issues.

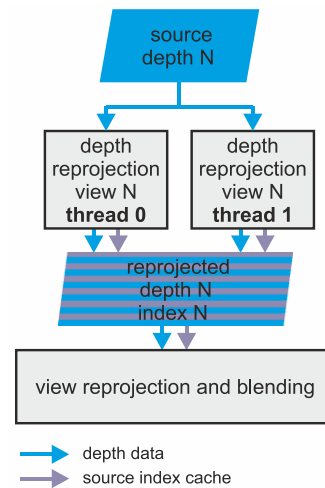


Figure 4. The data flow of the atomic z-test (AZT) with intermediate data structures.

5. IMPLEMENTATION

The proposed atomic z-test algorithm was implemented using the C++11 standard library [ISO11] and operations defined in `<atomic>` header. The `std::compare_exchange_weak<uint64_t>` function was chosen as a portable way to use compare-and-swap operation. The exemplary implementation of atomic z-test is provided below:

```
/**
 * @brief Performs atomic z test on DepthIndex buffer
 * @param PtrDI is pointer to location within buffer
 * @param NewD is reprojected depth value
 * @param NewI is index representing source depth location
 */
void AZT(uint64_t* PtrDI, uint32_t NewD, uint32_t NewI)
{
    uint64_t BuffDI = *PtrDI; //DI - DepthIndex
    uint32_t BuffD = (uint32_t)(BuffDI >> 32);
    if(BuffD <= NewD)
    {
        uint64_t NewDI = ((uint64_t)NewD<<32)|((uint64_t)NewI;
        while(!std::atomic_compare_exchange_weak(
            (std::atomic_uint64_t*)(PtrDI), &BuffDI, NewDI))
        {
            if((uint32_t)(BuffDI >> 32) >= NewD) { break; }
        }
    }
}
```

In addition to the usage of atomic CAS, we used already described vectorization techniques [Sta20], [Sta22] by using AVX2 and AVX512 extensions.

6. EXPERIMENTS

Test sequences

The test set contained 9 miscellaneous test sequences (Fig. 5), including:

- 3 omnidirectional sequences:
 - A01: ClassroomVideo [Kro18] (4K×2K),
 - C01: Hijack [Dor18] (4K×2K),
 - C02: Cyberpunk [Jeo21] (2K×2K),

- 3 perspective computer-generated sequences:
 - J01: Kitchen [Boi18] (FullHD),
 - J04: Fan [Dor20] (FullHD),
 - W02: Dancing [Boi18] (FullHD),
- 3 perspective natural sequences:
 - D01: Painter [Doy18] (2K×1K),
 - L01: Fencing [Dom16] (FullHD),
 - L03: MartialArts [Mie23] (FullHD).



Figure 5. Test set used in the experiments. 1st row (from left): ClassroomVideo, Hijack, Cyberpunk; 2nd row: Kitchen, Fan, Dancing; 3rd row: Painter, Fencing, MartialArts.

The sequences are commonly used in immersive video applications, e.g., within ISO/IEC JTC1/SC29/WG04 MPEG Video Coding group [MPEG23].

Experiment setup

Test was performed on two computers with modern x86-64 CPUs: AMD Ryzen 9 5950X (containing 16 uniform cores) and Intel Core i7-12700k (containing 8 regular and 4 weak cores). Both processors are able to execute instructions from AVX2 extension set so during experimental evaluation the AVX2 vectorized implementation was used. Unfortunately, during experiments we have no access to any AVX512 capable CPU, therefore no results for AVX512 implementation is provided.

Quality and time evaluation

In order to assess the quality of virtual views synthesized using proposed real-time view synthesizer, we compared it to the state-of-the-art ISO/IEC MPEG's reference software – RVS [Fac18], [MPEG18]. The comparison is reported in terms of two objective quality metrics commonly used in works on immersive video: IV-PSNR [Dzi22] and WS-PSNR [Sun17].

The computational complexity was evaluated by measuring the average processing time required for synthesizing of a single frame of the sequence including timing for individual processing stages (depth projection, depth combining, view projection, virtual view filtering and inpainting).

Evaluation results

The results of performed experiments are presented in Tables 1 – 3.

Tables 1 and 2 present a detailed comparison of computational time of the proposed algorithm and the previous implementation described in [Sta20]. As presented, the proposed atomic z-test (AZT) operation allows for a significant reduction of the computational time.

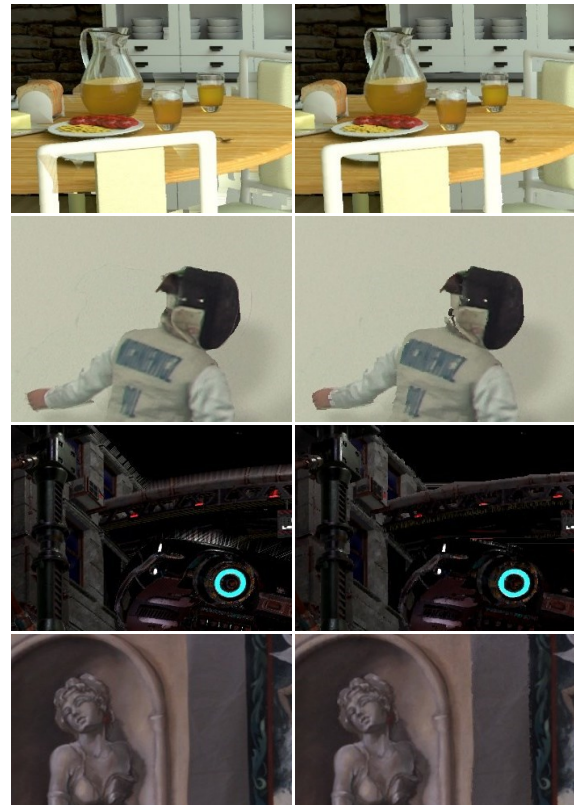


Figure 6. Fragments of virtual views synthesized using RVS (left) and the proposed method (right); sequences (from top): Kitchen, Fencing, Cyberpunk, and Painter.

For the 16-core AMD Ryzen CPU, AZT decreases the time required for synthesis of a single frame by more than 40%. For Intel i7-12700K CPU the decrease is smaller, but still significant. There are two reasons for lower gain caused by the use of AZT. The first one – a smaller number of cores, thus lower parallelization. The second reason is the heterogeneous structure of that CPU, which operates on 8 performant cores and 4 slower ones leading to unequal processing time for each core type.

As presented in Tables 1 and 2, the decrease of the computational time is caused by introducing the possibility of efficient parallelization of the depth projection step (denoted as “DP”). Such an implementation may lead to slightly longer step of combining depth candidates into the final virtual depth

Sequence	IPT algorithm processing time [ms]						Proposed algorithm processing time [ms]						Time reduction [%]	
	DP	DC	TP	F	I	Total	DP	DC	TP	F	I	Total	Depth	Total
A01	38.70	2.57	10.43	6.17	3.56	61.44	14.48	4.65	11.07	6.27	3.21	39.68	54%	35%
C01	38.52	2.52	10.05	5.27	2.73	59.10	13.72	4.66	10.94	4.90	2.71	36.93	55%	38%
C02	23.37	6.19	5.17	2.07	1.40	38.19	7.30	1.85	5.58	2.18	2.04	18.95	69%	50%
D01	5.60	3.31	2.87	2.03	1.01	14.81	2.98	0.84	2.79	1.55	0.67	8.84	57%	40%
J01	5.74	2.94	2.39	1.07	0.63	12.77	2.79	0.78	2.48	1.12	0.58	7.73	59%	39%
J04	7.54	2.93	2.47	1.37	0.76	15.07	2.82	0.77	2.50	1.15	0.60	7.84	66%	48%
L01	7.57	2.87	2.77	1.30	0.73	15.24	3.11	0.81	2.63	1.22	0.60	8.37	62%	45%
L03	7.71	2.94	2.42	1.24	0.80	15.11	2.90	0.84	2.55	1.14	0.65	8.07	65%	47%
W02	8.06	2.91	2.51	1.39	0.81	15.68	3.07	0.80	2.52	1.23	0.61	8.24	65%	47%
Average													61%	43%

Table 1. Performance evaluation – computation time comparison between IPT [Sta20] and proposed AZT algorithm for 16-core AMD Ryzen 9 5950X CPU. Processing stages: DP – depth projection, DC – depth combining, TP – texture projection, F – filtering, I – inpainting. “Depth time reduction” includes depth projection and depth combining.

Sequence	IPT algorithm processing time [ms]						Proposed algorithm processing time [ms]						Time reduction [%]	
	DP	DC	TP	F	I	Total	DP	DC	TP	F	I	Total	Depth	Total
A01	38.54	4.18	11.41	9.25	3.93	67.31	29.89	4.25	11.66	9.24	3.45	58.48	20%	13%
C01	37.34	4.13	11.10	6.46	3.45	62.48	30.47	4.24	11.29	6.76	3.16	55.92	16%	11%
C02	20.66	6.95	5.34	3.41	1.87	38.23	15.82	1.96	5.54	3.78	1.97	29.06	36%	24%
D01	5.81	3.36	3.00	2.91	1.19	16.29	6.67	0.81	2.73	2.95	1.10	14.27	18%	12%
J01	4.66	3.19	2.70	1.81	0.76	13.12	5.25	0.69	2.29	1.70	1.17	11.10	24%	15%
J04	5.17	3.19	2.70	1.88	0.75	13.69	5.43	0.71	2.35	1.95	0.78	11.23	27%	18%
L01	5.54	3.26	2.87	2.02	0.78	14.48	5.70	0.75	2.64	1.95	0.75	11.79	27%	19%
L03	5.36	3.18	2.98	1.54	1.68	14.74	5.62	0.71	2.91	1.61	0.85	11.69	26%	21%
W02	5.65	3.22	2.86	2.02	1.61	15.36	6.02	0.74	2.68	2.07	0.82	12.33	24%	20%
Average													24%	17%

Table 2. Performance evaluation – computation time comparison between IPT [Sta20] and proposed AZT algorithm for [8+4] core Intel i7-12700K CPU. Processing stages: DP – depth projection, DC – depth combining, TP – texture projection, F – filtering, I – inpainting. “Depth time reduction” includes depth projection and depth combining.

Sequence	Processing time [ms]			IV-PSNR [dB]			WS-PSNR [dB]		
	RVS	Proposed	Speedup	RVS	Proposed	Delta	RVS	Proposed	Delta
A01	15885	39.68	400	43.56	42.68	-0.89	32.21	31.74	-0.47
C01	15547	36.93	421	45.04	45.85	0.82	38.14	38.57	0.43
C02	7878	18.95	416	47.73	48.23	0.50	41.01	41.43	0.42
D01	3838	8.84	434	48.59	46.85	-1.74	38.46	36.94	-1.52
J01	3370	7.73	436	37.03	38.12	1.09	28.82	29.30	0.49
J04	3723	7.84	475	36.80	37.55	0.74	27.21	27.98	0.76
L01	3355	8.37	401	40.54	40.14	-0.40	29.55	29.21	-0.34
L03	3285	8.07	407	31.80	31.24	-0.55	26.81	26.14	-0.67
W02	3437	8.24	417	41.63	41.06	-0.57	29.43	28.91	-0.52
Average			423	41.41	41.30	-0.11	32.41	32.25	-0.16

Table 3. Performance and quality evaluation – comparison with the state-of-the-art view synthesis method RVS [Fac18].

map (depth map corresponding to the virtual view), but in total the entire depth processing step is significantly faster. All remaining steps (texture projection, filtering, and inpainting) are not impacted by the proposed AZT algorithm.

Table 3 shows the comparison of the quality of virtual views synthesized using the proposed method and the state-of-the-art synthesizer RVS. As presented, in terms of objective quality, both algorithms provide similar results, both for IV-PSNR and WS-PSNR. A slight quality decrease (0.11 for IV-PSNR and 0.16 for

WS-PSNR, on average) is the cost for a huge speedup – the proposed algorithm is more than 400 times faster than RVS.

Also the subjective quality of virtual views synthesized using RVS and proposed method is similar (Fig. 6). The characteristics of the synthesis artifacts is slightly different (e.g., caused by a different inpainting technique), but it can be certainly stated, that the proposed real-time algorithm allows to achieve at least similar quality to the state-of-the-art view synthesis technique.

7. CONCLUSIONS

In the paper we have presented a versatile CPU-based virtual view synthesis method which can be used in practical, real-time immersive video systems.

The novelty of the proposed method is based on introducing the atomic z-test approach, allowing for massive parallelization of the depth reprojection step, which is a crucial and most time-consuming part of the entire view synthesis pipeline.

The proposed virtual view synthesis method allows for achieving real-time processing for both perspective and omnidirectional sequences, even for very high resolutions. As presented in the paper, for 4K sequence it is possible to achieve real-time view synthesis at 25 frames per second. For lower resolutions (i.e., FullHD) it requires less than 10 ms per frame, making it possible to be used also for high frame rate immersive video systems.

8. ACKNOWLEDGMENTS

This work was supported by the Ministry of Education and Science of Republic of Poland.

9. REFERENCES

- [Aki15] Akin, A., Capoccia, R., Narinx, J., Masur, J., Schmid, A., and Leblebici, Y. Real-time free viewpoint synthesis using three-camera disparity estimation hardware. 2015 IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, pp. 2525-2528, 2015.
- [Boi18] Boissonade P., and Jung J. Proposition of new sequences for Windowed-6DoF experiments on compression, synthesis, and depth estimation. Document ISO/IEC JTC1/SC29/WG11 MPEG/M43318, Ljubljana, Slovenia, Jul. 2018.
- [Dom16] Domański M. et al. Multiview test video sequences for free navigation exploration obtained using pairs of cameras. Doc. ISO/IEC JTC1/SC29/WG11, MPEG M38247, 2016.
- [Dor18] Doré, R. Technicolor 3DoF+ test materials. ISO/IEC JTC1/SC29/WG11 MPEG, M42349, San Diego, CA, USA, 04.2018.
- [Dor20] Doré R. et al. InterdigitalFan0 content proposal for MIV. Doc. ISO/IEC JTC1/SC29/WG04 MPEG VC/ M54732, Online, Jul. 2020.
- [Doy18] Doyen D. et al. [MPEG-I Visual] New Version of the Pseudo-Rectified Technicolor painter Content. Doc. ISO/IEC JTC1/SC29/WG11 MPEG/M43366, Ljubljana, 2018.
- [Dzi18] Dziembowski, A., and Stankowski, J. Real-time CPU-based virtual view synthesis. 2018 International Conference on Signals and Electronic Systems, Kraków, Poland, 2018.
- [Dzi19] Dziembowski, A., Mieloch, D., Stankiewicz, O., Domański, M., Lee, G., and Seo, J. Virtual view synthesis for 3DoF+ video. 2019 Picture Coding Symposium (PCS), Ningbo, China, 2019.
- [Dzi22] Dziembowski A., Mieloch D., Stankowski J. and Grzelka A., IV-PSNR—The Objective Quality Metric for Immersive Video Applications, IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 11, pp. 7575-7591, Nov. 2022, doi: 10.1109/TCSVT.2022.3179575.
- [Fac18] Fachada, S., Bonatto, D., Schenkel, A., and Lafruit, G. Depth image based view synthesis with multiple reference views for virtual reality. 3DTV-Conference: The True Vision – Capture, Transmission and Display of 3D Video (3DTV-CON), Helsinki, Finland, 2018.
- [Goo12] Goorts, P., Dumont, M., Rogmans, S., and Bekaert, P. An end-to-end system for free viewpoint video for smooth camera transitions. 2012 International Conference on 3D Imaging (IC3D). Liege, Belgium, 2012.
- [Hua19] Huang, H., Wang, Y., Chen, W., Lin, P. and Huang, C. System and VLSI implementation of phase-based view synthesis. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, pp. 1428-1432, 2019.
- [ISO11] Information technology — Programming languages — C++, ISO/IEC 14882:2011, ISO/IEC JTC 1/SC 22 International Organization for Standardization.
- [Jeo21] Jeong, J.Y., Yun, K.J., Lee, G., Cheong, W.S., and Yoo, S. [MIV] ERP Content Proposal for MIV ver.1 Verification Test. ISO/IEC JTC1/SC29/WG04 MPEG VC, M58433, Online, 10.2021.
- [Kro18] Kroon, B. 3DoF+ test sequence ClassroomVideo. ISO/IEC JTC1/SC29/WG11 MPEG, M42415, San Diego, CA, USA, 04.2018.
- [Li19] Li, Y., Claesen, L., Huang, K., and Zhao, M. A real-time high-quality complete system for depth image-based rendering on FPGA. IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 4, pp. 1179-1193, 2019.
- [Mie23] Mieloch, D., Dziembowski, A., Szydełko, B., Klóska, D., Grzelka, A., Stankowski, J., Domański, M., Lee, G., and Jeong, J.Y. [MIV] New natural content – MartialArts. ISO/IEC JTC1/SC29/WG04 MPEG VC, M61949, Online, 01.2023.
- [MPEG18] “Reference View Synthesizer (RVS) manual,” Doc. ISO/IEC JTC1/SC29/WG11 MPEG, N18068, Macao, Oct. 2018.

- [MPEG23] Common test conditions for MPEG immersive video. ISO/IEC JTC1/SC29/WG04 MPEG VC, N0307, Online, Jan. 2023.
- [Mül11] Müller, K., Merkle, P., and Wiegand, T. 3-D Video Representation Using Depth Maps. *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643-656, Apr. 2011.
- [Non18] Nonaka, K., Watanabe, R., Chen, J., Sabirin, H., and Naito, S. Fast plane-based free-viewpoint synthesis for real-time live streaming. 2018 IEEE Visual Communications and Image Processing (VCIP), Taichung, Taiwan, pp. 1-4, 2018.
- [Sen18] Senoh, T., Tetsutani, N., and Yasuda, H. Depth estimation and view synthesis for immersive media. 2018 International Conference on 3D Immersion (IC3D), Brussels, Belgium, 2018.
- [Sch15] Schweizer H., Besta M. and Hoefler T., "Evaluating the Cost of Atomic Operations on Modern Architectures," 2015 International Conference on Parallel Architecture and Compilation (PACT), San Francisco, CA, USA, 2015, pp. 445-456, doi: 10.1109/PACT.2015.24.
- [Sta18] Stankiewicz, O., Domański, M., Dziembowski, A., Grzelka, A., Mieloch, D., Samelak, and J. A Free-viewpoint Television system for horizontal virtual navigation. *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 2182-2195, 2018.
- [Sta20] Stankowski, J., and Dziembowski, A. Fast view synthesis for immersive video systems. *Proceedings of the 28. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG'2020*, Plzen, Czech Republic, 05.2020.
- [Sta22] Stankowski J., and Dziembowski A., Real-time CPU-based view synthesis for omnidirectional video, 30th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2022, Pilsen, Czech Republic, 05.2022.
- [Sun17] Sun, Y., Lu, A., and Yu, L. Weighted-to-Spherically-Uniform Quality Evaluation for Omnidirectional Video. *IEEE Signal Processing Letters* 24.9(2017):1408-1412.
- [Zha17] Zhang, L., Li, Y., Zhu, Q., and Li, M. Generating virtual images for multi-view video. *Chinese Journal of Electronics*, vol. 26, no. 4, pp. 810-813, 2017.
- [Zhu84] Zhu C.-Q. and Yew P.-C., "A synchronization scheme and its applications for large multiprocessor systems", *Proc. 4th Int. Conf. Distrib. Computing Syst.*, pp. 486-493, 1984.

Evolutionary-Edge Bundling with Concatenation Process of Control Points

Ryosuke Saga^[0000-0003-1528-6534]

Osaka Metropolitan University
1-1 Gakuen-cho, Naka-ku
5998531, Sakai, Japan
r.saga@omu.ac.jp

Jaeyoung Baek

Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku
5998531, Sakai, Japan

ABSTRACT

Edge bundling is one of the information visualization techniques, which bundle the edges of a network diagram based on certain rules to increase the visibility of the network diagram and facilitate the analysis of key relationships among nodes. As one of evolutionary-based edge bundling, genetic algorithm-based edge bundling (called GABEB) is proposed which uses a genetic algorithm to optimize the placement of edges based on aesthetic criteria. However, it does not sufficiently consider the bundling between neighboring edges, and thus visual clutter issues still remain. Based on the above background, we propose an improved bundling method that considers the concatenating of control points at each edge using GABEB.

Keywords

Edge Bundling, Genetic Algorithm, Node-link Diagram, Information Visualization, Genetic Algorithm-based Edge Bundling (GABEB).

1. INTRODUCTION

Edge bundling is a method to reduce the visual clutter of a node-link diagram and facilitate intuitive understanding by adjusting the position of nodes and the arrangement of edges in a node-link diagram according to certain rules. Many studies have already proposed various edge-bundling methods, such as Force-directed Edge Bundling (FDEB) [Hol09] (Fig. 1), which is based on the dynamic rules and geometric rule-based methods such as Geometry-based Edge Bundling (GBEB) [Cui08].

On the other hand, Evolutionary-based edge bundling approaches like genetic algorithms (GA), which are evolutionary computations, have been implemented [Fer18][Mei22]. This is approached as an optimization problem to maximize the viewability defined by aesthetic rules, etc. These approaches are expected to provide visualization results that are not expected by humans. Among them, Genetic algorithm-based edge bundling (GABEB) is proposed [Sag20], which treats bundling as an optimization problem of a fitness function based on an evaluation value of aesthetic criteria [Sag16] and tries to optimize edge placement directly by moving control points. However, GABEB does not consider the bundling between edges located

in the neighborhood, wherein the neighboring edges does not overlap exactly, while visual clutters remain.

For example, as shown in Fig. 2, there are two parallel edges of equal length and distance 10 apart. In GABEB, the edge bundling is expressed by moving these control points, but in this case, it is desirable that at least the second control point is completely attached to each other. In this case, it is desirable that at least the second control points are completely attached to each other like dashed circles. In this case, it is desirable for v_1 and v_2 to move (5.0,0), (-5.0, 0), but calculating these values is difficult in GA because of the random number factor involved, and errors will inevitably appear. Therefore, some kind of post-processing is necessary. In other words, if the control points are considered to almost overlap, it is necessary to add a process to overlap (merge) them.

In this study, we aim to improve the visibility problem of GABEB by adding a process considering the bundling of multiple edges located in the neighborhood. We focused on bundling edges by

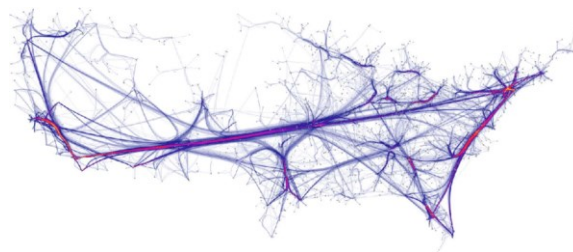


Figure 1. Edge Bundling Example [Hol09]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

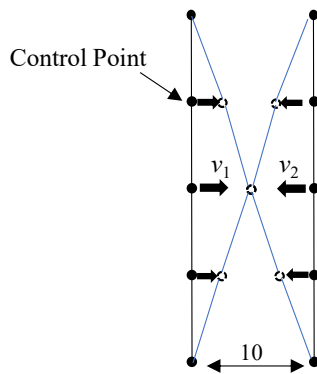


Figure 2. Problem description in GABEB

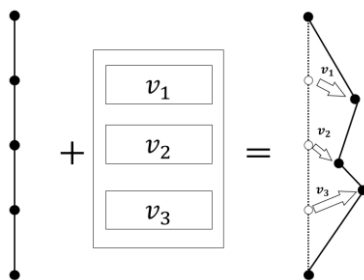


Figure 3. Genetic representation in GABEB

concatenating control points of neighboring edges and propose an improved bundling method that adds the process of concatenating control points of neighboring edges.

2. RELATED WORK

Edge Bundling and Evolutionary-based Edge Bundling

Edge bundling is a well researched research topic. Most works in this area define a model to express edge bundling with one of the best known methods being Holten's work where they proposed Hierarchical Edge Bundling for a graph based on a tree structure [Hol06].

Geometry-Based Edge Bundling (GBEB) proposed by Cui et al. [Cui08] realises edge bundling so as to bend edges based on meshes generated through a Delaunay triangulation, although this approach sometimes leads to some extreme bends. On the other hand, Holten et al. [Hol09] proposed FDEB which performs bundling based on Hooke's law. Also, Selassie et al. introduced Divided Edge Bundling by improving FDEB to apply to directed graph [Sel11], while Hurter et al. proposed Kernel Density Estimation Edge Bundling based on image-based visualisation [Hur12].

On the other hand, the approaches categorised into evolutionary-based edge bundling are proposed. Many graph layout algorithms using genetic algorithm [Bar00] [Bra96] [Elo96] [Net12] [Vra06] [Zha05] have been proposed since the last century. These methods aim to place nodes in a plausible way by

optimizing some evaluation value, and have been proposed for not only directed graph but also undirected graphs, orthogonal graphs and so on. The evolutionary-based approaches are based on the idea of graph layout algorithms, which view edge bundling as an optimization problem and attempt to implement edge bundling by solving the optimization problem. Ferreira et al. [Fer18] proposed a bundling method by solving the edge combination optimization problem. The method is useful but FDEB is necessary to bundling in real. Saga et al. [Sag20] proposed a method by solving the placement problem of each control point on edges. Although this method produces a bundling result as a result of the evolutionary computation, it has the problem of leaving visual clutter if the optimization is not successful. This paper proposes a method to solve the problem.

3. Genetic algorithm-based Edge Bundling

Genetic algorithm

Genetic algorithms (GAs), which belong to the family of evolutionary algorithms, simulate Darwin's theory of evolution [Gol89]. GAs are employed to solve difficult, often NP-hard, optimization problems. The genetic representation and fitness function depend on the problem and domain to solve. After these are defined, a GA proceeds iteratively through stages of selection, crossover, and mutation to improve a population of individuals that expresses candidate solutions to the problem.

GABEB is one of the algorithms based on the GA which treated bundling as a placement optimization problem of edge control points based on aesthetic criteria.

Genetic Representation

The genetic representation of GABEB is based on control-based approaches. The approach employed in FDEB divides an edge uniformly by c control points. By moving these control points the edges can be controlled for edge bundling. In our algorithm, edges in the input graph are also divided based on c uniformly spaced points as shown in Fig. 3. Then, for each control point, GABEB stores a distance-limited displacement vector v (as (x, y) coordinates) (where the limited distance is called maximum movement distance). Thus, for n edges and using c control points per edge, we encode $2 * n * c$ parameters.

Fitness Function

An appropriate fitness function is key to a successful GA. Here, there are also some general accepted aesthetic rules. The data-ink ratio [Tuf01] is one of the most widely used ones to evaluate visualization results quantitatively in all of visualization problems. It is based on the ink amount required for drawing a

visualized figure. The path quality, proposed by Cui in GBEB, is also useful to evaluate the degree of zig-zag in edge bundling. Furthermore, Saga proposed three quantitative criteria to evaluate edge bundling which are formulated from the difference of edge length, area illustrated by edges (which is similar to data-ink ratio), and density of edges [Sag16].

GABEB adopts these three criteria together with the path quality by Cui, and uses the four criteria separately and perform multi-objective optimization.

3.3.1 Mean Edge Length Difference

Mean Edge Length Difference (MELD) is a criterion to express the difference from the original edges after edge bundling. A smaller change of edge lengths indicates superior edge bundling because of over-bundling, whereas a large change often leads to a loss of the meaning of the original network. MELD is calculated as

$$MELD = \frac{1}{n} \sum_{e \in E} |L'(e) - L(e)| \quad (1)$$

where n is the number of edges, E is the edge set, and $L(e)$ and $L'(e)$ are the lengths of edge e before and after edge bundling, respectively. Employing this criterion, we can prevent edges from over-bending and over-bundling. MELD can be normalized to $[0;1]$ by

$$MELD = \frac{1}{n} \sum_{e \in E} |1 - L'(e)/L(e)|$$

GABEB aims to minimize the MELD.

3.3.2 Mean of Occupation Area

Mean of Occupation Area (MOA) indicates the degree among the compressed areas before and after edge bundling. Based on the idea that better bundling can compress the area occupied by the edges, MOA is calculated as

$$MOA = \frac{1}{N} \left| \bigcup_{e \in E} O(e) \right| \quad (2)$$

where N is the number of total areas, $O(e)$ is the set of areas occupied by edge e based on an occupation degree (we use 5% of unit area), and $||$ indicates the number of elements contained by a set. Minimizing the MOA is one of optimization goals of GABEB.

3.3.3 Edge Density Distribution

Edge Density Distribution (EDD) is rooted in the idea that a better edge bundling method can gather edges within a unit area and that the density per unit is high. EDD is calculated as

$$EDD = \frac{1}{|P|} \sum_{p \in P} (H(p) - H)^2 \quad (4)$$

where P is a set of pixels, $H(p)$ is the number of edges pathing pixel p , and H is the average of $H(p)$. GABEB aims to minimize the EDD.

3.3.4 Path Quality

Path Quality (PQ) expresses the degree of zig-zag. The higher the PQ, the better the edge bundling. PQ is calculated by the summation of angle differences between neighbors as

$$PQ = \sum_{e \in E} (-\sum_{i=3}^m \gamma_i |\Delta_i|) \quad (5)$$

with

$$\Delta_i = \begin{cases} A_i - A_{i-1} & \text{if } -\pi < |A_i - A_{i-1}| < \pi \\ |A_i - A_{i-1}| - 2\pi & \text{if } |A_i - A_{i-1}| > \pi \\ 2\pi + |A_i - A_{i-1}| & \text{if } |A_i - A_{i-1}| < -\pi \end{cases} \quad (6)$$

and

$$\gamma_i = \begin{cases} 0 & \text{if } \text{sign}(\Delta_i) = \text{sign}(\Delta_{i-1}) \\ 1 & \text{if } \text{sign}(\Delta_i) \neq \text{sign}(\Delta_{i-1}) \end{cases} \quad (7)$$

, where m is the number of segments divided by control points+1, and A_i is the angle between the original edge and the segment edge. GABEB tries to maximize PQ.

Genetic Operations

The main process of the proposed method follows NSGA-II [Deb et al., 2002] which is a method for multi-objective optimizations. Also, the genetic representation consists of real value for each gene, so the process uses BLX- α [Eshelman and Schaffer, 1993] for crossover. The overall of this process is as follows.

1. Initial population generation and evaluation
2. Selection, crossover by BLX- α and random mutation
3. Evaluation
4. Generation updating
5. Repeat 2. to 4. until the termination condition is satisfied.

Here, a generation is regarded as the process from step 2 to step 4. And BLX- α crossover operates to randomly generate a child from an extended area of the hyper-rectangle composed of the two parents, as shown in the following equation. From the parental genes p and q of dimension D , the child gene x is generated by the formula

$$x_i = r_i p_i + (1 - r_i) q_i \quad (8)$$

where i is an index of dimension. Also, the termination condition is configured by the number of generations. Using this process, the vector of each control point in the gene is changed in order to ensure that the edges are well bundled. However, it is quite difficult for control points to overlap and bundle with each other, etc., since GABEB are dealing with real values of vectors. In particular, when the amount of movement of v is large, control points of adjacent edges rarely overlap.

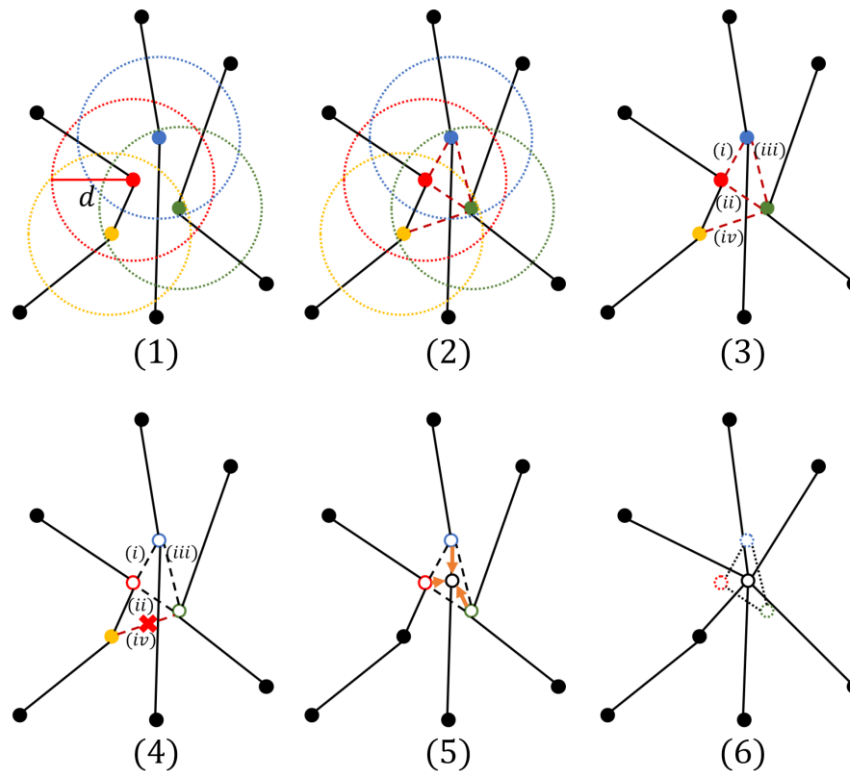


Figure 4. Concatenation Process

In order to improve the results of edge bundling, we add the steps related to concatenation process we propose before and after crossover and mutation steps. Hereafter, we describe the algorithm for concatenation and deconcatenation processes in detail.

4. GABEB WITH CONCATENATION PROCESS

In this paper, we propose a bundling method that considers concatenating of control points at neighboring edges to improve the visual clutter problem in GABEB. The overall of the improved process is as follows.

1. Initial population generation and evaluation
2. Deconcatenation of control points
3. Selection, crossover by BLX- α and random mutation
4. Concatenation of control points
5. Evaluation
6. Generation updating
7. Repeat 2. to 6. until the termination condition is satisfied.

Hereafter, we describe the algorithm for concatenation and deconcatenation processes in detail.

Control Point Concatenation and Deconcatenation Process

In this paper, we propose a bundling method that considers concatenating of control points at neighboring edges to improve the visual clutter problem in GABEB.

4.1.1 Concatenation Process

After the crossover and mutation process, the concatenating process of control points is performed. The control point merging process is performed as follows. An example figure of the concatenation process is shown in Fig. 4.

1. For all control points belonging to each edge, find the neighbouring control points where the distance is less than d (called maximum concatenating distance) and there is no control point belonging common edge in the combined set of control points (Fig. 4 (1), (2)).
2. Determine concatenating pairs in order of shorter distance between control points. If a control point included in a common edge is newly added to the set of control points that have already been joined by a control point pair that has already been decided to be concatenated, no concatenating is performed (Fig. 4 (3)). For example, when considering concatenation of the pair of control points shown in (iv), after the control point pairs (i), (ii), and (iii) have already been decided to be joined, the control point pair (iv) is judged that they are belonging to common edge due to the pair (iii), thus the control point pair (iv) is not joined.

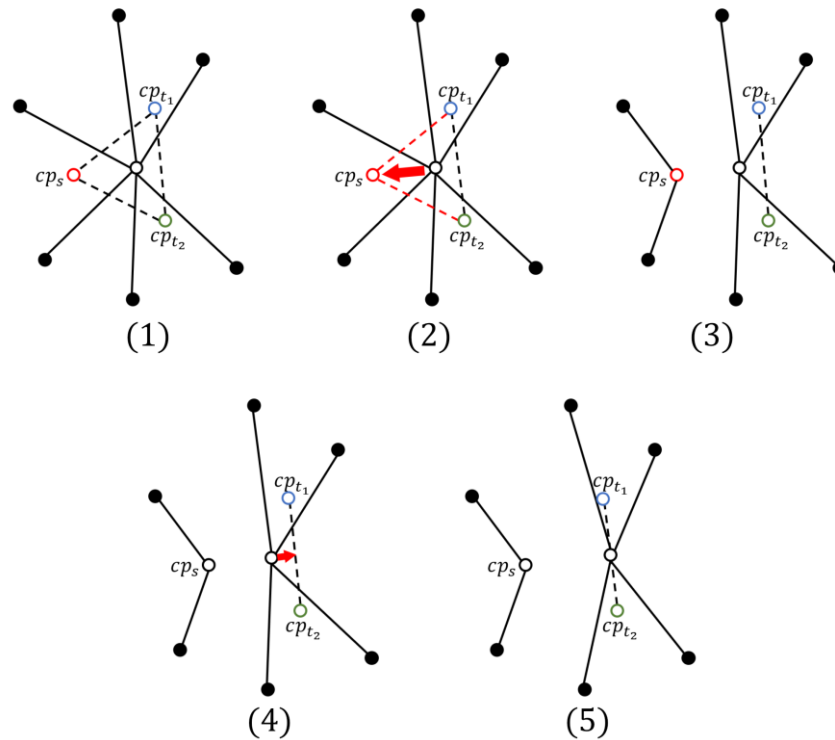


Figure 5. Deconcatenation Process

3. Stores the *pos* positions of each control point for which a join has been determined to be concatenating (Fig. 4 (4)).
4. Calculate the average position of each set of combined control points and assign them as the sharing position of the control points in the set shown as Fig. 4 (5), (6).

4.1.2 Deconcatenation Process

Before the crossover and mutation process, deconcatenation is performed when a bound control points within an individual becomes operation targets of crossover or mutation. The following procedure is used for deconcatenation (Fig. 5).

1. In the crossover and mutation process, check the presence of the control points that are bound to the target gene (Fig. 5 (1)).
2. If a bound control point cp_s is included in the control point set CP_s , remove the control point cp_s from CP_s and assign the position pos_s of the control point cp_s as the new position of the control point (Fig. 5 (2), (3)).
3. Perform the unbinding process for control points cp_t ($\forall cp_t \in CP_s$). In the case that cp_t is only bound to cp_s , assign the position pos_t of the control point cp_t as the position of the control point. If there are other control points bound to cp_s , calculate the average position of CP_s without cp_s and assign it to CP_s (Fig. 5 (4), (5)).

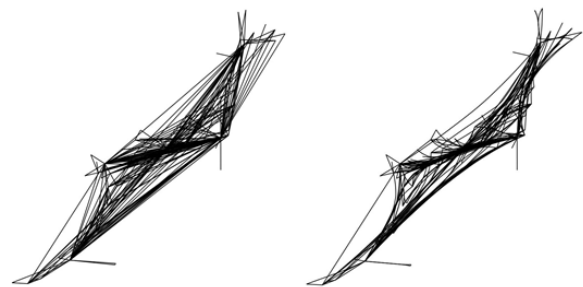


Figure 6. Original Japan Aerial Map(left) and FDEB result(right)

5. EXPERIMENTS

Goal, Dataset, Parameters and criteria

To check the effectiveness of the proposed method, we performed the experiments by applying proposed method to the node-link diagrams.

In this experiment, we used the node-link diagrams of the aerial map in Japan. The node-link diagram consists of 79 nodes (airports) and 233 edges (routes) in total. Bundled graph of the aerial map in Japan by FDEB is shown in Fig. 6 as example.

In the experiments, we check the four evaluation criteria used in fitness function, MELD, MOA, EDD, and PQ explained in Section 3.3. Also, we use Hypervolume which is widely used as an evaluation indicator for the non-dominated solutions in multi-objective optimization problems [Zit98][Li19]. The hypervolume is calculated from the area formed by the reference point and the solution set. And the larger this Hypervolume, the better the solution set is considered. We used reference point for Hypervolume to the worst value of each objective function.

Also, as parameters, the maximum movement distance and the maximum concatenating distance d are set to 10, 20, 30, 50. And the other parameters of the experiments are shown in Table 1.

Experiment Results

Bundled graph of the edge bundling results are shown in Fig. 6 and Fig. 7. Fig. 6 is the results of pareto solutions by GABEB and Fig. 7 is the results of proposed results. Also, bundled results with the change of the connection distance are shown in Fig. 8. We first compared the result figures of the bundling between Fig. 7 and Fig. 8. Because of the large d , the edges are basically hard to coalesce in GABEB. However, the results of the bundling in Fig. 8 are improved by the concatenation of the control points by the connection of the nearest neighbors. The comparison of the results of the bundling by the distance of the concatenation in Fig. 9 also shows that the more control points are aggregated as d increases. As a result, aggregation of the wide-area edges is well performed in bigger concatenating distances.

Next, we compared the evaluation value. The average evaluation value of the population is shown in Table 2, which shows proposed method archived better values in the two or three evaluation values by GABEB. Also, Hypervolume value of the non-dominated solutions in the whole population is shown in Table 3, and it indicates proposed method acquires more diverse solutions.

On the other hand, the computation time of proposed method shown in Table 5 is worse than GABEB. Moreover, the more longer movement distance

Initial Population Size	1000
Max Population Size	2000
Crossover Probability	0.9
Mutation Probability	0.05
α for BLX- α	0.5
Termination of generation	1000
MOA Unit Size	5
Control Point	3

Table 1. Parameters of Experiments

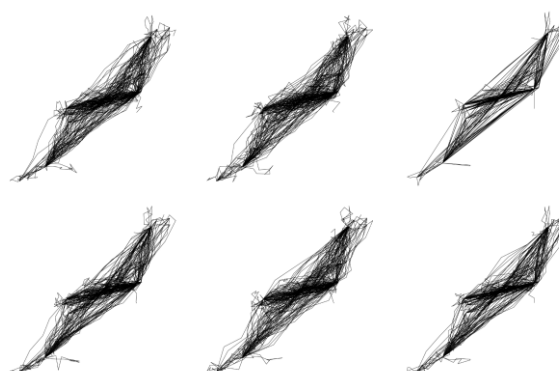


Figure 7. Examples of GABEB

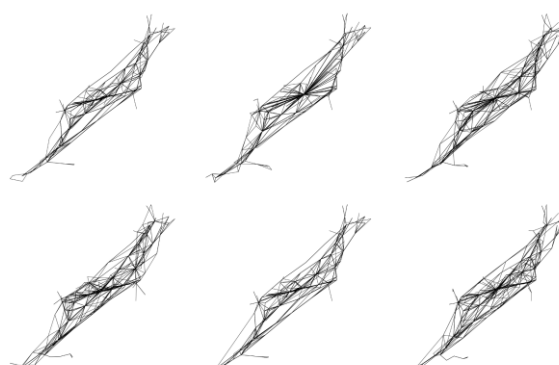


Figure 8. Sample Pareto Solutions of Proposed Method

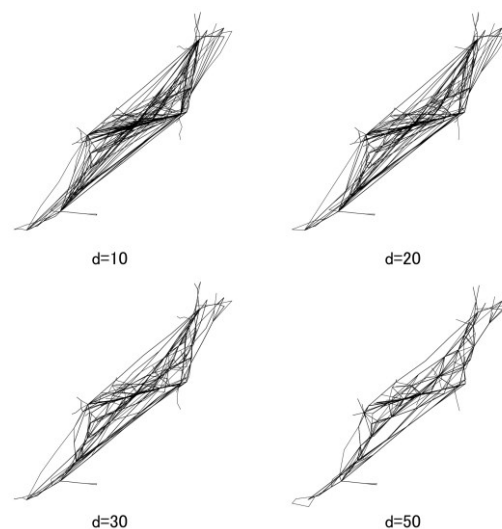


Figure 9. Example of Proposed Method Results with Different Concatenation Distance

increases computation time significantly compared to GABEB.

In the proposed method, the calculation of the distance between the control points in the concatenating process requires large amount of computation time. Thus, the alternative method of the calculation of the

d	Method	MELD	MOA	EDD	PQ
10	GABEB	0.9	0.133	2.135	19.208
	Proposed	98.57	0.023	0.248	23.626
20	GABEB	3.594	0.134	2.251	33.555
	Proposed	29.296	0.055	0.514	35.13
30	GABEB	7.816	0.135	2.328	46.623
	Proposed	13.255	0.104	0.827	68.274
50	GABEB	19.788	0.139	2.463	67.727
	Proposed	28.596	0.141	0.86	117.526

**Table 2. Evaluation of Edge Bundling Result
(Average of values)**

d	GABEB	Proposed
10	1679.129	4780.816
20	2626.932	7534.682
30	3523.665	13293.775
50	4755.462	22836.456

**Table 3. Hypervolume value of Non-dominated
Solutions**

d	GABEB	Proposed
10	134.322	199.701
20	144.838	242.154
30	145.207	273.804
50	147.366	336.035

**Table 4. Average computation time for an
generation(sec)**

distance of the control points such as approximation neighborhood search method needs to be considered.

6. CONCLUSION

GABEB is a method of bundling using a genetic algorithm as an optimization problem for edge placement based on aesthetic criteria, but GABEB does not sufficiently consider bundling process between neighboring edges, which causes the result of leaving visual clutter in the bundling results. We proposed an improved bundling method based on GABEB by considering the concatenation of control points of neighboring edges. By concatenating neighboring control points that satisfy certain conditions and proceeding with optimization with shared positions, which enabled to aggregating many control points and improving visual clutters.

In the experiment, proposed method performed bundling on Japan aerial map, and the results were compared with GABEB. Experiment results showed that the proposed method obtained a better evaluation values in some evaluation values and a more diverse solution set.

As future works, we believe it is necessary to solve the computational speed problem that makes application to large-scale node-link diagrams difficult, with faster concatenation processing. For this purpose, we plan to incorporate techniques such as Local Sensitive Hashing [Ind98] and SketchSort [Tab10] which are fast Nearest-Neighbor methods. Also, the results in this paper are shown using GA, but we would like to verify whether other optimization methods based on computational intelligence (such as meta heuristic algorithms like firefly algorithm [Yan08] can also be applied in Edge-Bundling, which aims for optimal placement of control points. Other possibilities include hardware acceleration (e.g., using GPGPU [Nak12]) rather than algorithms. Also, the proposed algorithm is implemented based on GABEB, which does not move nodes. Therefore, since graph drawing which is an algorithm to place the nodes properly needs to be considered separately, it is necessary to implement an algorithm that takes node placement into account as well.

7. ACKNOWLEDGMENTS

We appreciate KAKENHI 22K12116 that supports this research.

8. REFERENCES

- [Bar00] Barreto, A. and Barbosa, H. Graph layout using a genetic algorithm. In Proc. of Sixth Brazilian Symposium on Neural Networks, pp. 179–184. 2000.
- [Bra96] Branke, J., Bucher, F., and Schneck, H.. Using Genetic Algorithms for Drawing Undirected Graphs. In The Third Nordic Workshop on Genetic Algorithms and their Applications, pp. 193–206, 1996
- [Cui08] Cui, W., Zhou, H., Qu, H., Wong, P. C., and Li, X. Geometry-based edge clustering for graph visualization. In IEEE Transactions on Visualization and Computer Graphics, volume 14, pp. 1277–1284, 2008
- [Deb02] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 182–197, 2002.
- [Elo96] Eloranta, T., Eloranta, T., and Mäkinen, E. Timga - a genetic algorithm for drawing undirected graphs. Technical report, Divulgaciones Matematicas, 1996
- [Esh93] Eshelman, L. J. and Schaffer, J. D. Real-Coded Genetic Algorithms and Interval-Schemata, Foundations of Genetic Algorithms, Vol. 2, pp. 187–202, 1993.

- [Fer18] Ferreira, J. d. M., do Nascimento, H. A., and Foulds, L. R. An evolutionary algorithm for an optimization model of edge bundling. *Information (Switzerland)*, Vol. 9, No. 7, pp. 1–27, 2018.
- [Gol89] Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., 1989.
- [Hol06] Holten, D. Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 12, No. 5, pp. 741–748, 2006.
- [Hol09] Holten, D. and Van Wijk, J. J. Force-Directed edge bundling for graph visualization. *Computer Graphics Forum*, Vol. 28, No. 3, pp. 983–990, 2009.
- [Hur12] Hurter, C., Ersoy, O., Telea, A. Graph bundling by kernel density estimation. *Computer Graphics Forum*, Vol. 31, No. 3, pp. 865–874, 2012.
- [Ind98] Indyk, P. and Motwani, R. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. of the Thirtieth Annual ACM Symposium on Theory of Computing, STOC '98*, pp. 604–613, 1998.
- [Li19] Li, M. and Yao, X. Quality evaluation of solution sets in multi-objective optimisation: A survey. *ACM Computing Surveys*, Vol. 52, No. 2, 2019.
- [Mei22] Meikari, J. and Saga, R. Evolutionary node layout and edge bundling. In *Proc. of 2022 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–6, 2022.
- [Nak12] Nakashima, T., Tanaka, K., Fujimoto, N. and Saga, R. GPGPU Implementation of fuzzy rule-based classifiers, *Smart Innovation, Systems and Technologies* Vol. 16, pp. 323–332, 2012.
- [Net12] Neta, B., Araújo, G., Guimarães, F., Mesquita, R., and Ekel, P. A fuzzy genetic algorithm for automatic orthogonal graph drawing. *Applied Soft Computing*, Vol. 12, pp. 1379–1389, 2012.
- [Sag16] Saga, R. Quantitative Evaluation for Edge Bundling Based on Structural Aesthetics. *EuroVis'16: In Proc. of the Eurographics /IEEE VGTC Conf. on Visualization*, pp. 1–3, 2016.
- [Sag20] Saga, R., Yoshikawa, T., Wakita, K., Sakamoto, K., Schaefer, G., and Nakashima, T. A genetic algorithm optimising control point placement for edge bundling. In *VISIGRAPP 2020 – Proc. of the 15th International Joint Conf. on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Vol. 3, pp. 217–222, 2020.
- [Sel11] Selassie, D., Heller, B., Heer, J. Divided edge bundling for directional network data. *IEEE Transaction Visualization & Computer Graphics*, Vol. 17, No. 12, pp. 2354–2363, 2011.
- [Tab10] Tabei, Y., Uno, T., Sugiyama, M. and Tsuda, K. Single versus multiple sorting in all pairs similarity search. In *Proc. of ACM L2010*, pp. 145–160, 2010.
- [Tuf01] Tufte, E. *The Visual Display of Quantitative Information*, Graphics Press USA, 2001.
- [Vra06] Vrajitoru, D. and El-Gamil, B. R. Genetic algorithms for graph layouts with geometric constraints. In *Proc. of International Conference on Climate Informatics*, 2006.
- [Yan08] Yang, X.-S. *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2008.
- [Zha05] Zhang, Q.-G., Liu, H.-Y., Zhang, W., and Guo, Y.-J. Drawing undirected graphs with genetic algorithms. In *International Conference on Natural Computation*, pp. 28–36. Springer.
- [Zit98] Zitzler, E. and Thiele, L. Multiobjective optimization using evolutionary algorithms - A comparative case study. In *Lecture Notes in Computer Science*, Vol. 1498, pp. 292–301, 1998.

Low-Rank Rational Approximation of Natural Trochoid Parameterizations

Csaba Bálint

Eötvös Loránd University
1/C Pázmány Péter stny.
1117 Budapest, Hungary
csabix@inf.elte.hu

Gábor Valasek

Eötvös Loránd University
1/C Pázmány Péter stny.
1117 Budapest, Hungary
valasek@inf.elte.hu

Lajos Gergő

Eötvös Loránd University
1/C Pázmány Péter stny.
1117 Budapest, Hungary
gergo@inf.elte.hu

ABSTRACT

Arc-length or natural parametrization of curves traverses the shape with unit speed, enabling uniform sampling and straightforward manipulation of functions defined on the geometry. However, Farouki and Sakkalis proved that it is impossible to parametrize a plane or space curve as a rational polynomial of its arc-length, except for the straight line. Nonetheless, it is possible to obtain approximate natural parameterizations that are exact up to any epsilon. If the given family of curves possesses a small number of scalar degrees of freedom, this results in simple approximation formulae applicable in high-performance scenarios. To demonstrate this, we consider the problem of finding the natural parametrization of ellipses and cycloids. This requires the inversion of elliptic integrals of the second kind. To this end, we formulate a two-dimensional approximation problem based on machine-epsilon exact Chebyshev proxies for the exact solutions. We also derive approximate low-rank and low-degree rational natural parametrizations via singular value decomposition. The resulting formulae have minimal memory and computational footprint, making them ideal for computer graphics applications.

Keywords

Curves, Approximation, Arc-length parametrization, Natural parametrization

1 INTRODUCTION AND PREVIOUS WORK

The parametrization of a curve is not unique since there are infinitely many variations that result in the same shape. However, the analytic derivatives of the curve do change in the process. Natural or arc-length parametrization stands out in the sense that the artifacts of parametrization are absent from the algebraic formulation of derivatives. In other words, the derivatives of an arc-length parametrized curve only consist of geometric invariants, such as curvature and torsion, and their derivatives. This is a direct consequence of the Frenet-Serret formulae [Car18].

Unfortunately, natural parametrization is not a feasible practical representation. This was first proven rigorously by Farouki and Sakkalis [FS07] when they showed that no plane or space curve may be parametrized as a rational polynomial function of its arc-length, except for the line. However, they have

identified a subset of polynomials that possess polynomial arc-length functions. This class of polynomials is referred to as Pythagorean hodographs. Although it is still not possible to parametrize these by arc-length, the ability to express the arc-length in closed form proved to be of merit in various applications [Far08].

Nonetheless, approximate arc-length parametrizations may offer practical alternatives. Farouki considered the Möbius transformation to reparametrize degree n Bézier curves such that the result is approximately unit speed [Far97]. He showed that there is a unique reparametrization of this kind that minimizes a functional that penalizes deviation from unit speed traversal. A more elementary derivation to this result was given by Jüttler in [Jüt97].

Sánchez-Reyes and Chacón proposed an approximate arc-length parametrization of plane curves in [SC15] that does not rely on optimization. They constructed second-order geometric Hermite interpolants [BHS87] to approximate an arbitrary input curve. Quintic polynomials are capable of reconstructing both geometric invariants and parameterization up to second order at endpoints [Sch98]. The latter were chosen such that the interpolant is unit-speed and possesses orthogonal first and second derivatives, while the former was used to reconstruct position, tangent, and curvature centers at the parametric endpoints. To achieve the desired accuracy, they employed multiple geometric Hermite segments.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

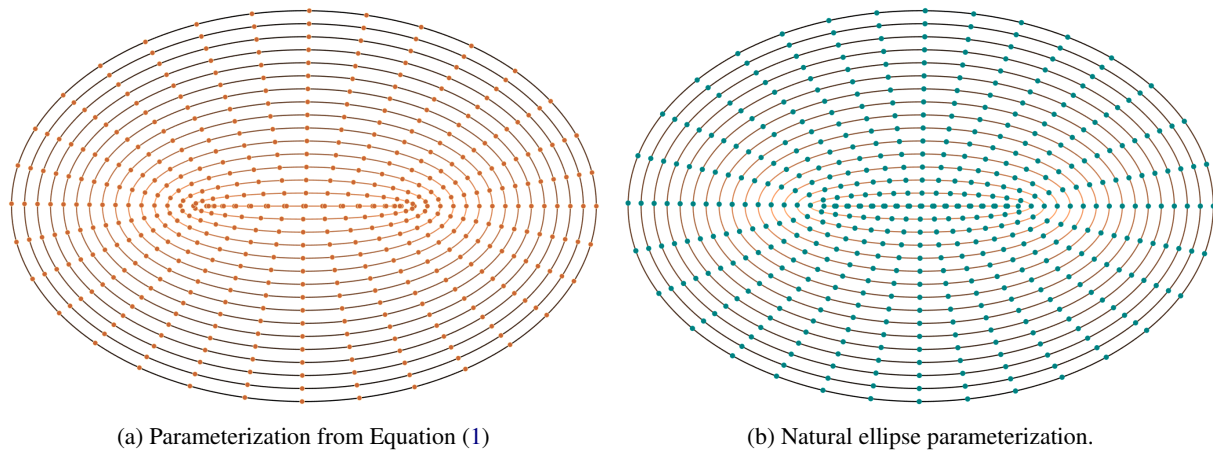


Figure 1: Ellipse and its natural parameterization

We also consider the problem of approximating arc-length parametrizations; however, we focus our attention on adjustable precision and restrict our discussion to trochoid curves only. To this end, we present a scalable framework that may be used to devise high-accuracy approximate natural parametrizations of trochoids by means of rational approximations.

In Section 2, we briefly review how an incomplete elliptic integral of the second kind is derived upon formulating the natural parametrization of ellipses and trochoids and what simplifications can be applied to it.

In Section 3, we show that a low-rank separable approximation may be computed from a properly sampled simplified formulation, and Section 4 shows that it can be realized by custom degree rational polynomials.

We show that even degree (2,1) rational polynomial separable approximations yield high accuracy results with negligible computational cost in Section 5.

2 APPROXIMATE NATURAL PARAMETERIZATIONS

There are various means to approximate the arc-length parametrization of a curve. Methods such as Runge-Kutta provide procedural solutions; however, our aim is to derive closed-form approximations.

Let us consider how the natural parameterization of an ellipse with semi-major axis a and semi-minor axis b is derived. A general parametrization is given as

$$\mathbf{p}(\phi) = \begin{bmatrix} a \cos(\phi) \\ b \sin(\phi) \end{bmatrix}, \quad \phi \in [0, 2\pi), \quad 0 < b < a \in \mathbb{R}. \quad (1)$$

From Equation (1), the arc-length function is

$$\begin{aligned} s_{\mathbf{p}}(\phi) &= \int_0^{\phi} \sqrt{a^2 \sin^2 \varphi + b^2 \cos^2 \varphi} \, d\varphi \\ &= b \cdot \int_0^{\phi} \sqrt{1 - \left(1 - \frac{a^2}{b^2}\right) \cdot \sin^2 \varphi} \, d\varphi. \end{aligned}$$

Thus, we can express the arc-length function with the incomplete elliptic integral of the second kind, denoted as $E(\phi|m)$. Inverting this and substituting back to Equation (1) leads to the natural parameterization of the ellipse:

$$\begin{aligned} s_{\mathbf{p}}(\phi) &= b \cdot E\left(\phi \mid 1 - \frac{a^2}{b^2}\right) \implies \\ \mathbf{p}(s_{\mathbf{p}}^{-1}(s)) &= \mathbf{p}\left(E^{-1}\left(\frac{s}{b} \mid 1 - \frac{a^2}{b^2}\right)\right). \end{aligned}$$

2.1 Trochoid parameterization

The following trochoid parameterization unifies hypotrochoids and epitrochoids while also describing circles, ellipses, hypocycloids, and epicycloids with only two parameters:

$$\mathbf{p}(\phi) = \begin{bmatrix} \cos \phi + a \cos(b\phi) \\ \sin \phi + a \sin(b\phi) \end{bmatrix}, \quad a, b \in \mathbb{R} \setminus \{0\}. \quad (2)$$

Note that if $a = 0$ or $b = 0$ or $b = 1$, the curve is just a circle. When $b = -1$, the above simplifies to an equation of an ellipse with $a + 1$ semi-major axis and $1 - a$ semi-minor axis. Observe that the curves

$$\mathbf{p}_{a,b}(\phi), \mathbf{p}_{-a,b}(\phi), \mathbf{p}_{a,\frac{1}{b}}(\phi), \mathbf{p}_{-a,\frac{1}{b}}(\phi)$$

only differ in scale and rotation from each other. Thus, without the loss of generality, we can assume that $a > 0$ and $0 \neq b \in [-1, 1]$.

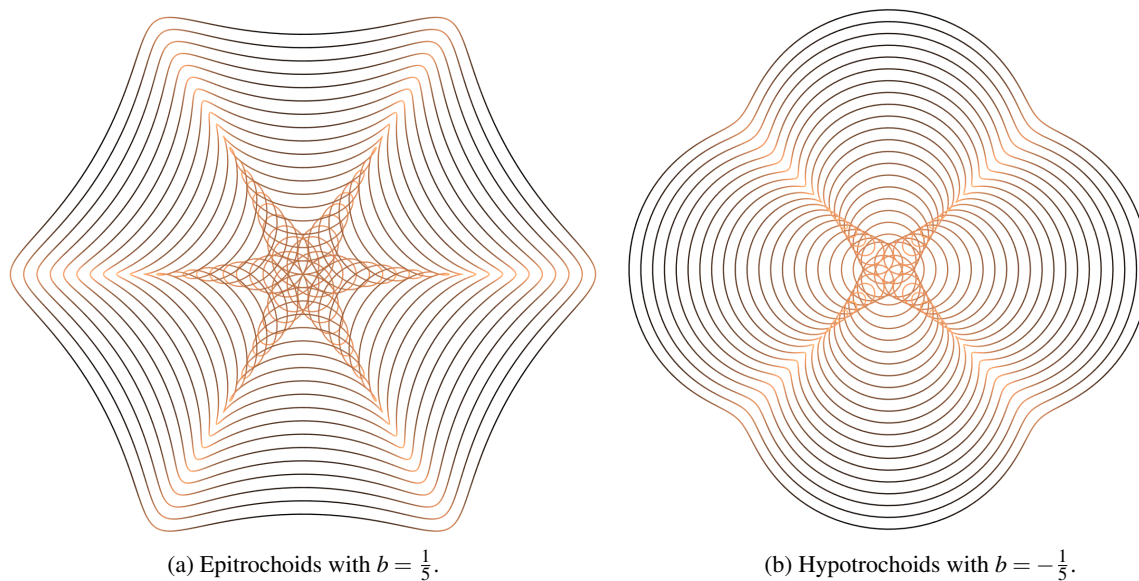


Figure 2: Trochoids with different values of a from Equation (2).

	Ellipses $b = -1$	Hypotrochoids $b < 0$			Epitrochoids $b > 0$			Circle $b = 1$
Curtate trochoids $ ab < 1$	$a = 0.25, b = -1$ 	$a = 0.5, b = -0.5$ 	$a = 0.75, b = -0.33$ 	$a = 1, b = -0.25$ 	$a = 1, b = 0.25$ 	$a = 0.75, b = 0.33$ 	$a = 0.5, b = 0.5$ 	$a = 0.25, b = 1$
	$a = 0.5, b = -1$ 	$a = 1, b = -0.5$ 	$a = 1.5, b = -0.33$ 	$a = 2, b = -0.25$ 	$a = 2, b = 0.25$ 	$a = 1.5, b = 0.33$ 	$a = 1, b = 0.5$ 	$a = 0.5, b = 1$
Cycloids $ ab = 1$	$a = 1, b = -1$ 	$a = 2, b = -0.5$ 	$a = 3, b = -0.33$ 	$a = 4, b = -0.25$ 	$a = 4, b = 0.25$ 	$a = 3, b = 0.33$ 	$a = 2, b = 0.5$ 	$a = 1, b = 1$
Prolate trochoids $ ab > 1$	$a = 2, b = -1$ 	$a = 4, b = -0.5$ 	$a = 6, b = -0.33$ 	$a = 8, b = -0.25$ 	$a = 8, b = 0.25$ 	$a = 6, b = 0.33$ 	$a = 4, b = 0.5$ 	$a = 2, b = 1$
	$a = 4, b = -1$ 	$a = 8, b = -0.5$ 	$a = 12, b = -0.33$ 	$a = 16, b = -0.25$ 	$a = 16, b = 0.25$ 	$a = 12, b = 0.33$ 	$a = 8, b = 0.5$ 	$a = 4, b = 1$

Figure 3: Classification of the trochoid curve family with Equation (2).

Table 1 and Figure 3 classify these trochoids, so we can relate Equation (2) to the well-known constructive derivation via rotating circles.

A wide range of plane curves fall into this categorization, such as the Tusi couple ($a = 1, b = -1$) that creates linear motion from rotational one. The cardioid ($a = 2, b = \frac{1}{2}$) that appears in our coffee cups or the deltoid curve $a = 2, b = -\frac{1}{2}$, limaçon curves $b = \frac{1}{2}$, the nephroid $a = 3, b = \frac{1}{3}$, the astroid $a = 3, b = -\frac{1}{3}$, and cycloids in general $|ab| = 1$ including the tautochrone and the brachistochrone curves.

We obtain the arc-length parameterization of trochoids similarly to ellipses:

$$\begin{aligned}
 s_p(\phi) &= \int_0^\phi \sqrt{a^2 b^2 + 2ab \cdot \cos((b-1)\varphi)} d\varphi \\
 &= \int_0^\phi \sqrt{(1+ab)^2 - 4ab \sin^2\left(\frac{b-1}{2}\varphi\right)} d\varphi \\
 &= |1+ab| \cdot \int_0^\phi \sqrt{1 - \frac{4ab}{(1+ab)^2} \sin^2\left(\varphi \frac{b-1}{2}\right)} d\varphi \\
 &= \frac{2|1+ab|}{b-1} \cdot E\left(\phi \frac{b-1}{2} \middle| \frac{4ab}{(1+ab)^2}\right) \quad (3)
 \end{aligned}$$

	$b = -1$	Hypotrochoids $b < 0$	Epitrochoids $b > 0$	$b = 1$
Curtate $ ab < 1$	ellipse	curtate hypotrochoid	curtate epitrochoid	circle
Cycloid $ ab = 1$	Tusi couple	hypocycloid	epicycloid	circle
Prolate $ ab > 1$	ellipse	prolate hypotrochoid	prolate epitrochoid	circle

Table 1: Classification of trochoids and their names.

2.2 Incomplete elliptic integral of the second kind

Before we approximate the inverse of the incomplete elliptic integral of the second kind, let us review its properties.

$$E(\phi|m) = \int_0^\phi \sqrt{1 - m \cdot \sin^2 \varphi} d\varphi, \quad \phi \in [0, 2\pi), m \in [0, 1]$$

The complete elliptic integral of the second kind is $E(m) = E(\frac{\pi}{2}|m)$. Special values: $E(0|m) = 0$, $E(\phi|0) = \phi$. The incomplete elliptic integral grows linearly and it is also 2π periodic, that is $E(\phi|m) = \frac{2}{\pi}E(m) \cdot \phi + E(\phi \bmod 2\pi | m)$. Moreover, we can tile the periodic part with the quarter period. For trochoids, we can calculate $E(\phi|m)$ for negative m values with $m \rightarrow \frac{m}{m-1} \in (0, 1)$ using the following formula. A pair of these corresponding values of m are highlighted in Figure 4a where the incomplete elliptic integral is visualized.

Lemma 1. For any $\phi \in \mathbb{R}$ and $0 \neq m \in \mathbb{R}$,

$$E(\phi|m) = \sqrt{1-m} \left(E\left(\frac{m}{m-1}\right) - E\left(\frac{\pi}{2} - \phi \middle| \frac{m}{m-1}\right) \right). \quad (4)$$

Proof. Substitute $\phi = \frac{\pi}{2} - \theta$ as if we rotated the trochoid by 90° degrees:

$$\begin{aligned} E(\phi|m) &= \int_0^\phi \sqrt{1 - m \sin^2 \varphi} d\varphi \\ &= \int_{\pi/2-\phi}^{\pi/2} \sqrt{1 - m \cos^2 \theta} d\theta \\ &= \sqrt{1-m} \int_{\pi/2-\phi}^{\pi/2} \sqrt{1 - \frac{m}{m-1} \sin^2 \theta} d\theta \\ &= \sqrt{1-m} \left(E\left(\frac{m}{m-1}\right) - E\left(\frac{\pi}{2} - \phi \middle| \frac{m}{m-1}\right) \right) \end{aligned}$$

□

Lemma 2. If $\xi \cdot E(m) = E(\phi|m)$ for any $\xi, \phi \in \mathbb{R}$ and $0 \neq m \in \mathbb{R}$, then

$$\phi = \frac{\pi}{2} - E^{-1} \left(E\left(\frac{m}{m-1}\right) (1 - \xi) \middle| \frac{m}{m-1} \right)$$

Proof. Substitute $\xi \cdot E(m)$ into the left side of Equation (4) and rearrange to obtain

$$E\left(\frac{\pi}{2} - \phi \middle| \frac{m}{m-1}\right) = E\left(\frac{m}{m-1}\right) - \frac{E(m)}{\sqrt{1-m}} \cdot \xi.$$

Using the same Lemma 1 for $\theta = \frac{\pi}{2}$, we get the formula for the complete elliptic integral $E(m) = \sqrt{1-m} \cdot E\left(\frac{m}{m-1}\right)$ which when applied for the above leads to

$$E\left(\frac{\pi}{2} - \phi \middle| \frac{m}{m-1}\right) = E\left(\frac{m}{m-1}\right) - E\left(\frac{m}{m-1}\right) \xi.$$

Rearranging the above completes the proof of Lemma 2. □

Figure 4 illustrates the geometry of the direct and inverse elliptic functions.

2.3 Simplifying $E^{-1}(\xi|m)$

Note that if $m < 0$, Lemma 2 shows that we can remap into the $(0, 1)$ range. Thus, let us now assume that $m \in [0, 1]$, $\xi \in [0, 1]$ so we can compute $\theta \in [0, \frac{\pi}{2}]$ from

$$\theta = E^{-1}(E(m) \cdot \xi | m),$$

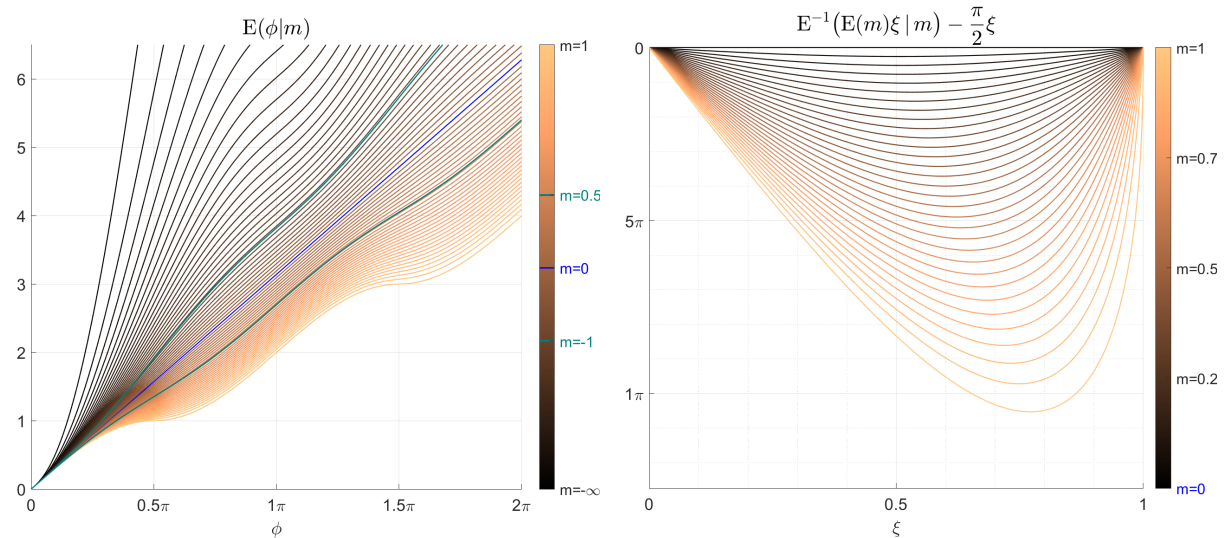
provided we have a good approximation of the right-hand side over the $(\xi, m) \in [0, 1]^2$ domain. To aid with the approximation, let us remove the linear term as in Figure 4b, square the function, and apply a $W(\xi, m)$ weight function:

$$\left(E^{-1}(E(m)\xi | m) - \frac{\pi}{2}\xi \right)^2 \approx W(\xi, m) \cdot G(\xi, m), \quad (5)$$

$$W(\xi, m) = \frac{m \cdot \xi \cdot (1 - \xi)}{\sqrt{2 - \xi - m}}, \quad (6)$$

where $G(\xi, m)$ is the function we approximate with. The weight function is used to smooth out the pole in the derivative at $(\xi, m) = (1, 1)$ and zero out the function at $m = 0$, $\xi = 0$, or $\xi = 1$ values where it should be zero. Figure 5a visualizes this approximation problem.

The computation of $G(\xi, m)$ can be carried out in multiple ways, but we found the above transformations helped the most to obtain a higher precision low-rank and low-degree approximation.

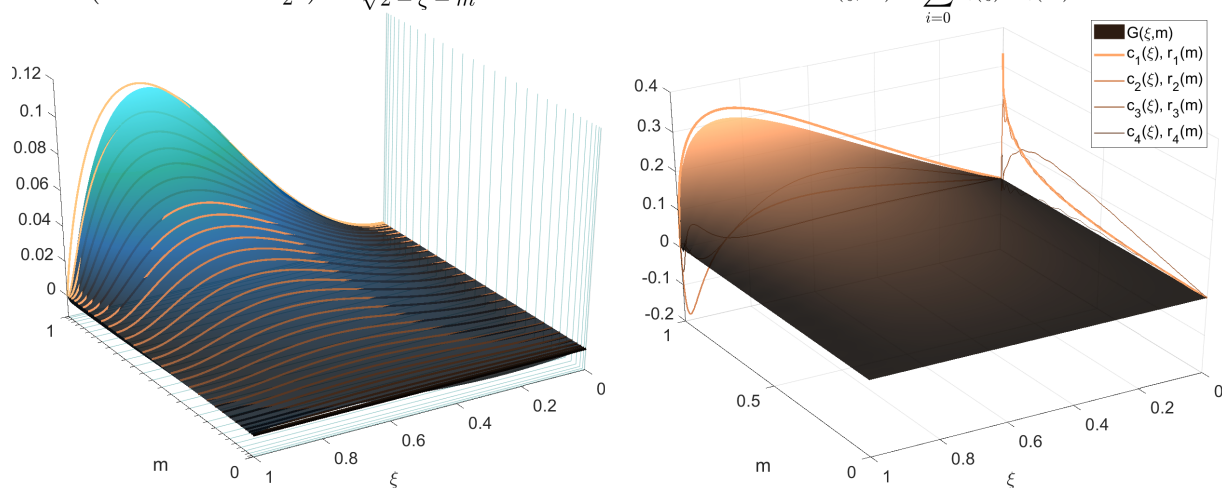


(a) $\theta \mapsto E(\theta|m)$ functions for $m \in (-\infty, 1]$. Lemma 1 relates the $m = 0.5$ and $m = -1$ teal curves. (b) $\xi \mapsto E^{-1}(E(m)\xi|m) - \frac{\pi}{2}\xi$ functions for various $m \in [0, 1]$ values show the deviation from the linear function.

Figure 4: The incomplete integral of the second kind and its modified inverse that we need to approximate.

$$\left(E^{-1}(E(m)\xi|m) - \frac{\pi}{2}\xi\right)^2 \approx \frac{\xi(1-\xi)m}{\sqrt{2-\xi-m}} \cdot G(\xi, m)$$

$$G(\xi, m) = \sum_{i=0}^k c_i(\xi) \cdot r_i(m)$$



(a) Transformed incomplete elliptic inverse orange curves approximated with the blue surface with weight function. (b) Low-rank component functions of the transformed inverse incomplete elliptic integral.

Figure 5: Transformation and low-rank approximation of the inverse elliptic integral of the second kind

3 LOW-RANK APPROXIMATION

We want to find a k -rank approximation first, that is, a pair of c_i and r_i functions such that

$$G(\xi, m) = \sum_{i=0}^k c_i(\xi) \cdot r_i(m) \quad c_i, r_i : [0, 1] \rightarrow \mathbb{R} \quad (7)$$

Figure 5b visualizes the $G(\xi, m)$ function with $c_i(\xi)$ and $r_i(m)$ component functions drawn on the back walls. If we evaluate the functions in Equation (5) at ξ_1, \dots, ξ_M and m_1, \dots, m_N values to get the matrix of function values we want G to reproduce, we can per-

form a k -rank Singular Value Decomposition (SVD), that is,

$$G = UDV^T, \quad U \in \mathbb{R}^{M \times k}, D \in \mathbb{R}^{k \times k}, V \in \mathbb{R}^{N \times k}$$

where we just have to find c_i and r_i functions that reproduce the i -th column of $U \cdot \sqrt{D}$, and $V \cdot \sqrt{D}$ matrices to extend G as a function with Equation (7). More specifically, for $i = 1, \dots, k$, $j = 1, \dots, M$, and $l = 1, \dots, N$:

$$c_i(\xi_j) \approx U_{ji} \sqrt{|D_{ii}|},$$

$$r_i(m_l) \approx V_{li} \operatorname{sgn} D_{ii} \sqrt{|D_{ii}|}.$$

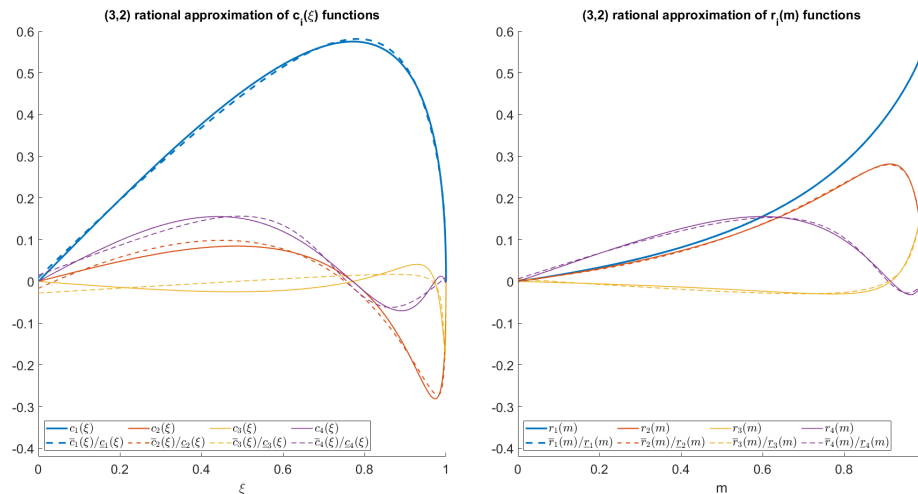


Figure 6: Component functions $c_i(\xi)$, $r_i(m)$ and their rational approximations $\frac{\bar{c}_i(\xi)}{\underline{c}_i(\xi)}$ and $\frac{\bar{r}_i(m)}{\underline{r}_i(m)}$ where the nominator polynomials are cubic, and the denominators are quadratic.

Figure 5 illustrates the transformation of the elliptic inverse integral and its decomposition into a low-rank approximation. Figure 6 shows the component functions of an approximation.

4 RATIONAL POLYNOMIAL APPROXIMATION

To approximate the above c_i and r_i functions at ξ_j and m_l values, we can apply low-degree rational polynomial approximations to find \bar{c}_i , \underline{c}_i , \bar{r}_i , and \underline{r}_i polynomials.

$$\begin{aligned} G(\xi, m) &= \sum_{i=0}^k c_i(\xi) \cdot r_i(m) \\ &= \sum_{i=0}^k \frac{\bar{c}_i(\xi)}{\underline{c}_i(\xi)} \cdot \frac{\bar{r}_i(m)}{\underline{r}_i(m)} \end{aligned} \quad (8)$$

For these last steps, we employed the Chebfun Matlab library [Dri14] for their state-of-the-art polynomial approximation algorithms. Figure 7 plots the error of a degree (3, 2) and rank 3 approximation.

5 RESULTS

Let us consider finding a separable (2, 1) rational approximation solution to the arc-length parametrization problem of trochoids using Matlab and Chebfun. Since Chebfun relies on the Chebyshev basis, our sample points are on the Chebyshev grid of

$$\begin{aligned} \xi_j &= -\cos\left(\pi \frac{j-1}{M-1}\right), \\ m_l &= -\cos\left(\pi \frac{l-1}{N-1}\right), \end{aligned}$$

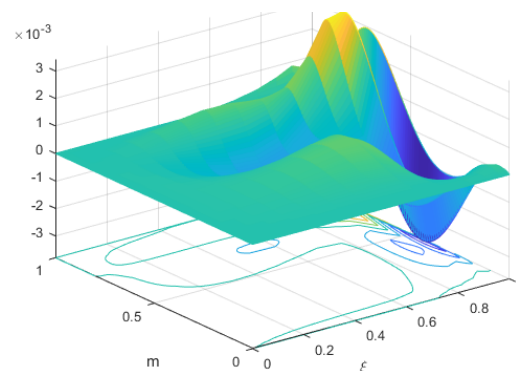


Figure 7: $W(\xi, m) \cdot G(\xi, m) - W(\xi, m) \cdot \hat{G}(\xi, m)$ approximation error for a 3-rank and (3,2)-degree decomposition.

for $j = 1, \dots, M$ and $l = 1, \dots, N$. The resulting approximation is

$$\begin{aligned} \hat{G}(\xi, m) &= \frac{\xi^2 - 0.9553 \cdot \xi - 0.04578}{1.539 \cdot \xi - 1.5626} \\ &\quad \cdot \frac{m^2 + 0.3193 \cdot m - 0.09471}{-23.993 \cdot m + 26.213} \\ \hat{E}(m) &= \frac{m^2 - 5.5788 \cdot m + 5.3188}{-2.6429 \cdot m + 3.3811} \end{aligned} \quad (9)$$

Figure 8 illustrates the error of this approximation compared to the exact arc-lengths for trochoids corresponding to $a = 1$ and $b = 1, 2, \dots, 16$. The error is within $3 \cdot 10^{-3}$ even at the arc-length of 60.

To evaluate our explicit approximate natural parameterization for any (a, b) trochoid at any $s \in \mathbb{R}$ parameter, we derive the inverse arc-length function from Eq. (3):

$$s_p^{-1}(s) = \frac{2}{b-1} \cdot E^{-1}\left(s \frac{b-1}{2|1+ab|} \middle| \frac{4ab}{(1+ab)^2}\right). \quad (10)$$

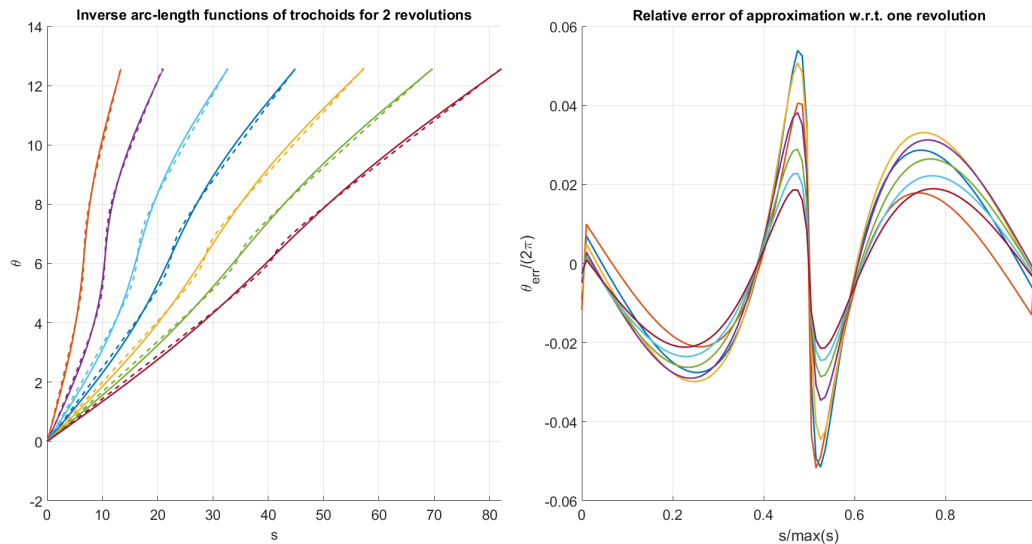


Figure 8: Precise s_p^{-1} compared to 1-rank separable (2,1)-degree rational approximation for trochoids with $a = 1, 3, \dots, 13$, and $b = 0.5$. Albeit the error is not small, the approximation here is a crude quadratic over linear rational separable approximation using the code seen in Fig. 9. Note that only one global approximation is needed to evaluate the arc-lengths of any trochoid.

Therefore, we can calculate

$$\xi := \frac{s}{\hat{E}(m)} \frac{b-1}{2|1+ab|}, \quad m := \frac{4ab}{(1+ab)^2}. \quad (11)$$

Since the approximant below has a period of 4 and a quarter period of 1 that we can tile with, that is

$$E^{-1}(E(m)(2-\xi)|m) = \pi - E^{-1}(E(m)\xi|m).$$

Assuming $\xi \in [0, 1]$, we compute \hat{G} with Eq. (8), for example with Eq. (9). Finally, evaluate $W(\xi, m)$ and $E^{-1}(s|m)$ which is obtained from Eq. (5) as

$$E^{-1}(E(m)\xi|m) = \frac{\pi}{2}\xi - \sqrt{W(\xi, m) \cdot G(\xi, m)}. \quad (12)$$

Figure 9 provides an example implementation for a purely separable approximation $g(\xi, m) = c(\xi) \cdot r(m)$ where the $c(\xi)$, $r(m)$, $E(m)$ functions are approximated with (2, 1) rational polynomials.

6 CONCLUSIONS

We derived an algorithmic framework to compute high-accuracy approximate trochoid natural parametrizations. This family of curves includes circles, ellipses, and a variety of well-known shapes.

We demonstrated that our formulation may be used to derive low-degree rational polynomial parametrizations that approximate unit-speed traversal. Nevertheless, for optimal results, we had to apply an appropriate simplification and weighting of the approximated function.

Even though we restricted the number of degrees of freedom to two by choice of trochoids, our construct may be generalized to higher dimensions, in other words, to curves of higher flexibility; this is a future research direction.

ACKNOWLEDGMENTS

Supported by the ÚNKP-22 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

REFERENCES

- [BHS87] Carl de Boor, Klaus Höllig, and Malcolm Sabin. “High accuracy geometric Hermite interpolation”. In: *Computer Aided Geometric Design* 4.4 (Dec. 1987), pp. 269–278. ISSN: 0167-8396. DOI: [10.1016/0167-8396\(87\)90002-1](https://doi.org/10.1016/0167-8396(87)90002-1).
- [Car18] Manfredo Perdigão do Carmo. *Differential geometry of curves & surfaces*. Revised & updated second edition. Mineola, New York: Dover Publications, INC, 2018. ISBN: 978-0-486-80699-0.
- [Dri14] Trefethen Driscoll Hale. *Chebfun Guide*. Oxford: Pafnuty Publications, 2014.
- [Far08] Rida T. Farouki. *Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable*. Ed. by Herbert Edelsbrunner, Leif Kobbelt, and Konrad Polthier. Vol. 1. Geometry and Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. ISBN: 978-3-540-73398-0. DOI: [10.1007/978-3-540-73398-0](https://doi.org/10.1007/978-3-540-73398-0).
- [Far97] Rida T. Farouki. “Optimal parameterizations”. In: *Computer Aided Geometric Design* 14.2 (Feb. 1997), pp. 153–168. ISSN: 01678396. DOI: [10.1016/S0167-8396\(96\)00026-X](https://doi.org/10.1016/S0167-8396(96)00026-X).

```

1 function [theta] = trochoidInverseArclength(s, a, b, Ep, Eq, Cp, Cq, Rp, Rq)
2     % Approximate inverse arc-length function of trochoids from Eq. (2)
3     arguments
4         s (1,:) double % distance taken on trochoid parameterized as:
5         a (1,1) double % p(t) = [ cos(t) + a*cos(b*t) ,
6         b (1,1) double % sin(t) + a*sin(b*t) ].
7         Ep (1,:) double = [ 1 -5.5788 5.3188 ] % Rational approx of the complete integral
8         Eq (1,:) double = [ -2.6429 3.3811 ] % E(m) ~ Ep(m)/Eq(m)
9         Cp (1,:) double = [ 1 -0.9544 -0.0481 ] % overlined c(xi) polynom in the paper
10        Cq (1,:) double = [ 1.2038 -1.2610 ] % underlined c(xi) polynom in the paper
11        Rp (1,:) double = [ 1 -3.5267 0.0720 ] % overlined r(m) polynom in the paper
12        Rq (1,:) double = [ 16.2103 -20.5069 ] % underlined r(m) polynom in the paper
13    end
14
15    m = 4*a*b/(1 + a*b + 1e-12).^2; % m from a,b in Eq. (3)
16    xi = 0.5*(b-1) / ((abs(1+a*b)+1e-12) * E(m)) * s; % xi from Eq. (11)
17    theta = 2/(b-1) * Einv(xi, m); % s^{-1}(s) from Eq. (10)
18
19    function e = E(m)
20        % Approximation of the complete elliptic integral of the second kind
21        e = 1;
22        if m < 0
23            e = sqrt(1-m); % Extending to m < 0 with Lemma 1 for the complete elliptic
24            m = m/(m-1); % We turn negative m values to be in [0,1] with Eq. (4)
25        end
26        e = e * polyval(Ep, m) ./ polyval(Eq, m); % Rational approximation
27    end
28
29    function einv = Einv(xi,m)
30        % Approximation of the inverse of the incomplete elliptic integral of the second kind
31        is_m_neg = m < 0;
32        if is_m_neg
33            xi = 1-xi; % We extend to m < 0 with Lemma 2 for the inverse elliptic
34            m = m/(m-1); % We turn negative m values to be in [0,1]
35        end
36
37        xi2 = mod(xi,2); % Remap xi from [-inf,inf] to [0,2].
38        xi1 = 1-abs(xi2-1); % Remap xi from [0,2] to [0,1]. Sign fixed in line 4
39
40        g = polyval(Cp,xi1) .* polyval(Rp,m) ... % Seperable rational approximation
41            ./ ( polyval(Cq,xi1) .* polyval(Rq,m) ); % of G(xi,m) from Eq. (8)
42        einv = 0.5*pi*xi - sign(1-xi2).*g; % Linear + period term in Eq. (12)
43
44        if is_m_neg
45            einv = 0.5*pi - einv; % If m was negative, we use Lemma 2.
46        end
47    end
48 end

```

Figure 9: This simplified Matlab code example evaluates approximate arc-length functions, omitting the weight function calculation.

- | | |
|--|--|
| <p>[FS07] Rida T. Farouki and Takis Sakkalis. “Rational space curves are not “unit speed””. In: <i>Computer Aided Geometric Design</i> 24.4 (May 2007), pp. 238–240. ISSN: 01678396. DOI: 10.1016/j.cagd.2007.01.004.</p> | <p>[SC15] J. Sánchez-Reyes and J. M. Chacón. “A polynomial Hermite interpolant for C2 quasi arc-length approximation”. en. In: <i>Computer-Aided Design</i> 62 (May 2015), pp. 218–226. ISSN: 0010-4485. DOI: 10.1016/j.cad.2014.12.001.</p> |
| <p>[Jüt97] Bert Jüttler. “A vegetarian approach to optimal parameterizations”. In: <i>Computer Aided Geometric Design</i> 14.9 (Dec. 1997), pp. 887–890. ISSN: 01678396. DOI: 10.1016/S0167-8396(97)00044-7.</p> | <p>[Sch98] Robert Schaback. “Optimal Geometric Hermite Interpolation of Curves”. In: (1998). Accepted: 2023-02-16T13:21:53Z Publisher: Vanderbilt U. Press.</p> |

Detection of Printed Circuit Board Defects with Photometric Stereo and Convolutional Neural Networks

Angelika Hable

Polymer Competence Center
Leoben GmbH
Roseggerstraße 12
A-8700 Leoben, Austria
angelika.hable@pccl.at

Marko Matore

Polymer Competence Center
Leoben GmbH
Roseggerstraße 12
A-8700 Leoben, Austria
marko.matore@pccl.at

Anton Scherr

AT&S Austria Technologie &
Systemtechnik Aktiengesellschaft
Industriepark 4
A-8350 Fehring, Austria
a.scherr@at.ats.net

Thomas Krivec

AT&S Austria Technologie &
Systemtechnik Aktiengesellschaft
Fabriksgasse 13
A-8700 Leoben, Austria
t.krivec@ats.net

Dieter Gruber

Polymer Competence Center
Leoben GmbH
Roseggerstraße 12
A-8700 Leoben, Austria
dieter.gruber@pccl.at

ABSTRACT

The quality inspection of printed circuit boards (PCBs) is no longer feasible by human inspectors due to accuracy requirements and the processing volume. Automated optical inspection systems must be specifically designed to meet the various inspection requirements. A photometric stereo setup is suitable for the inspection of highly reflective gold areas on PCBs. In this process, several image acquisitions are performed under different illumination directions. This can reveal defects that are not visible under other illumination systems. In this paper, we use a segmented ring light so that image acquisition is possible under four different illumination directions. Using these images, a normal map and a mean image are calculated. The defects on the gold areas of PCBs are detectable in either the normal map, the mean image, or both images. A convolutional neural network (CNN) for classification detects the defects. The input is a 6-dimensional image from normal map and mean image. An accuracy up to 95% can be achieved with the available dataset.

Keywords

Photometric stereo, printed circuit board defect detection, convolutional neural network, normal map, quality inspection

1. INTRODUCTION

It is impossible to imagine our modern world without printed circuit boards (PCBs). They are found in electronic devices and are therefore produced in very high quantities. At the same time, there are very high requirements for their quality inspection. These requirements can only be carried out by automatic inspection system, as human inspection is not possible due to the high production rates and the required accuracy of defect detection. Inspection systems for PCBs differ greatly in their scope of application. For example, there are systems for detecting missing electronic assemblies, for inspecting solder joints, and for inspecting PCB patterns. In this publication we deal with the detection of defects on gold areas (surface finish) of PCBs. This is an extension of [Hab+22]. Surface treatments have the purpose of protecting the copper from corrosion so that electronic

components can later be soldered onto the copper layer. Gold coatings belong to highly reflective surfaces, thus defects can partially become visible only under certain illumination directions. One illumination approach for the inspection of highly reflective components is photometric stereo [Woo80]. In this method, at least three image acquisitions of a component are taken under different illumination directions. This allows the orientation of the surface of the component to be determined for each point [Woo80] and to create a normal image. With the obtained normal map, a trained convolutional neural network can be used to detect defects such as cracks, dents and scratches on metallic parts [Cer+20]. Another possibility is to predict the normal map using convolutional neural networks (CNNs) [Cao+22, PMM22]. In addition to the orientation of the surface normals, height information of the component surface

3D depth map can be used to detect defects. Podrekar [Pod+17] et al. use a 3D depth map for quality inspection on tablets, which is calculated from the surface normal using path integration procedure. Defects can be detected by comparing the maps with a tablet model. Curvature images can also be used to visualize defects such as scratches and wrinkles on surfaces. Curvature images are derived from the surface gradient [Ren+20]. For the detection of defects that do not have a height characteristic, 3D depth map, normal map and curvature images are not suitable. To detect these defect classes in addition to defects with height characteristics on the component surfaces, additional image information must be taken into account. Saiz [Sai+22] et al. uses several photometric images and merges them to a RGB image which is fed as input to the segmentation CNN for the detection of defects on nickel-plated components. They propose a combination of curvature images, texture images and range images (image gradient magnitude). Another method to perform quality inspection on highly reflective components is to fuse the images with different illuminations to an RGB image [LOK19]. With this method Roughness, Slope and Reflectivity [LOK19] of the component surface can be obtained. In this publication we focus on the detection of defects on the gold areas of PCBs. The defect classes include defects that are only visible through the surface normal images, defects that are only visible through a texture image and defects that can be detected in both images. For this reason, a normal map and a texture image are necessary for quality inspection. Unlike [Sai+22] and [LOK19], we do not merge both images into one RGB image, but provide the defect detection CNNs with a 6-channel image as input.

2. EXPERIMENTAL

The PCB images were acquired using an inspection setup consisting of a high resolution area scan camera, telecentric lens, segmented ring light and XY stage. Figure 1 shows the setup. The setup of the inspection system and the inspection method is described in detail in [Hab+22]. However, instead of homogeneous illumination as mentioned in [Hab+22], a white light segmented ring light (CCS HPR2-150SW-DV04M12-5) from CCS Inc. was used. With this type of illumination, certain defect classes on the highly reflective gold areas could be visualized. The individual four quadrants of this illumination can be controlled one after the other with an LSS Light Sequence Switch from CCS Inc., This allows images to be acquired from different directions of illumination. On the basis of the acquired images a photometric stereo processing is possible. The surface normals were calculated with the least mean square function of the numpy library [Har+20] from the four image acquisitions and the positions of the illuminations (light vectors).

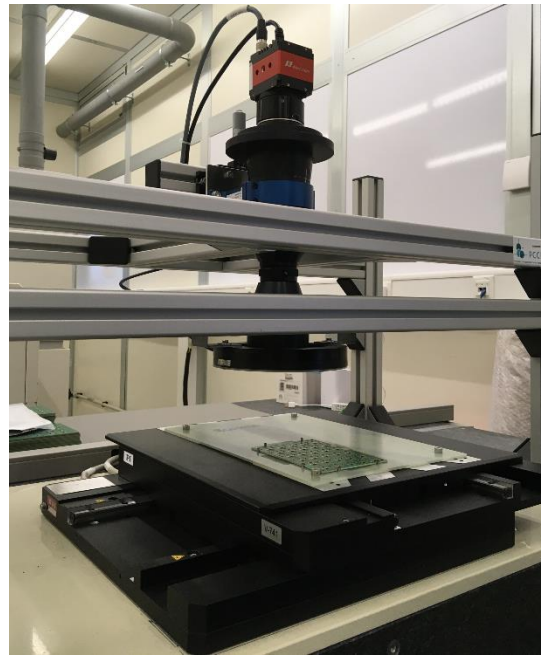


Figure 1: Inspection setup for the detection of PCB defects on gold areas. An XY stage moves the PCBs under the camera. Photometric imaging is performed with a segmented ring light and a high resolution camera so that four image acquisitions can be taken of each PCB from different illumination directions.

The light vectors are determined during the calibration of the photometric system. In this publication, we used a metal sphere and the formulas from [VW] to calculate the light vectors. In this calibration method, a metal sphere is placed under the camera and a total of 4 image captures are taken from different illumination directions using the segmented ring light. The position of the light rays directly reflected by the metal sphere can be determined by a thresholding procedure. In the case of specular reflection, the angle of reflection and the angle of observation are the same, so that the observed light intensity is equal to the incident intensity. By determining the center coordinates of the sphere, the radius and the position of the direct reflection on the sphere, the normal vector and subsequently the light vectors can be calculated using the equations in [VW]. Unfortunately, the results from the calculated light vectors showed a non-uniform normal map. This means that the color coding varies in different areas of the PCB. For a flat surface, the color coding should be the same. Despite non-uniform normal map, the defects remain visible. In addition to the normal map, a mean image is calculated from the four image acquisitions by fusing them to one RGB image using the numpy mean function.

The defects examined are copper residue, dent, contamination and discoloration. Each defect class is visible either in the normal map, in the mean image,

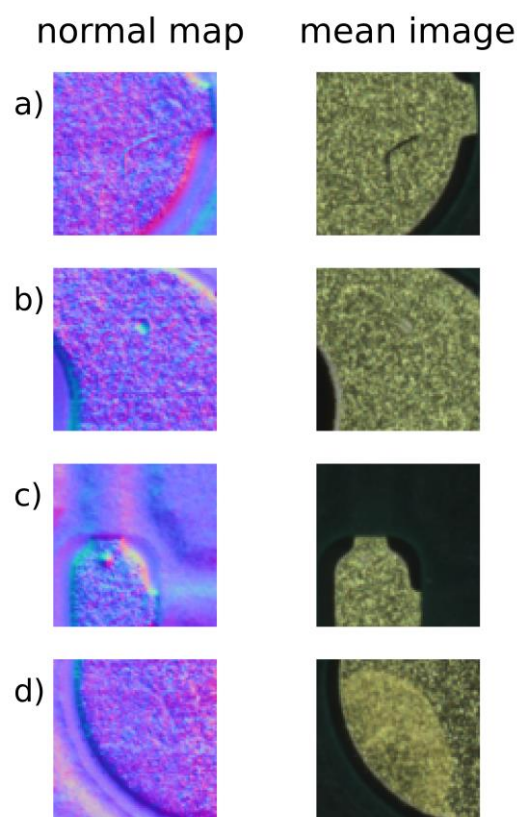


Figure 2: Defect classes displayed normal map and mean image. (a) contamination, (b) dent, (c) copper residue and (d) discoloration.

or in both images. In Figure 2 the examined defects are shown. The defect contamination is visible both in the mean image and partially in the normal map. The sliding window approach [Hab+22] is used to inspect the quality of the PCBs. Here, 100x100 patches are cut out from the normal map and mean images and are then resized to 124x124 patches and merged to a 124x124x6 dimensional image. This image is the input for the classification CNN. The output is a defect class (copper residue, dent, contamination, discoloration and OK).

3. RESULTS AND DISCUSSION

The classification CNN has a 6-dimensional image input and is composed of two consecutive convolutional layers, a batch normalization layer and a max pooling layer. At the end there are two fully connected layers with dropout layer in between. The architecture of the CNN is shown in Figure 3. Since the input consists of a 6-dimensional image (mean image and normal map), no existing network architecture such as ResNet with Transfer Learning was used, but an own network architecture was developed. Existing network architectures such as ResNet use 3-dimensional images as input. An adaptation of an existing network architecture to our needs would mean a complete training on a deep network and would be computationally more

	training	validation	test
copper residue	1083 (3249)	232	232
dent	2317 (4634)	496	495
OK	3417 (5142)	733	731
contamination	422 (5908)	90	85
discoloration	178 (3738)	38	37

Table 1: Composition of the entire dataset divided into training, validation and test dataset. The numbers indicate the amount of existing defects in the dataset. In parentheses are the number of existing data after augmentation by the augmentation algorithms.

expensive than the training of a small network from scratch. For this reason, a self-developed network architecture without transfer learning was chosen. When designing the architecture and selecting the hyperparameters such as batch size and learning rate, several training runs were performed and the network architecture and parameters were selected so that the best results in terms of accuracy and performance could be achieved. The PyTorch framework [Pas+19] was used to create the classification CNN and for training. Cross Entropy Loss, a batch size of 64, a learning rate of 0.0001 was chosen. The network was trained for 100 epochs. As optimizer the function OneCycleLR from PyTorch [Pas+19] was used. The maximum learning rate, a parameter of the optimizer, was set to 0.005.

The composition of the image data per defect is shown in Table 1. The dataset is very unbalanced. There are only few image data of the defect class discoloration while many image data of the OK class are present.

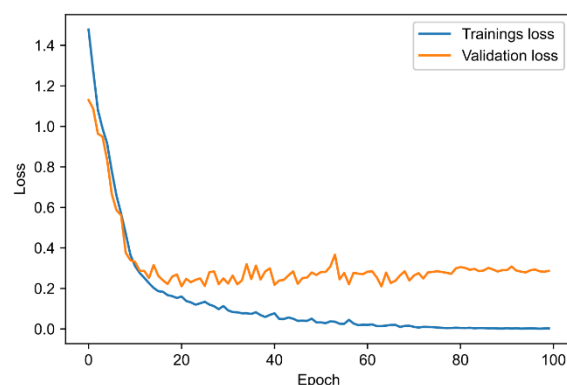


Figure 4: Training loss and validation loss when training the CNN on 100 epochs. The smallest validation loss (epoch 64) was chosen for the evaluation.

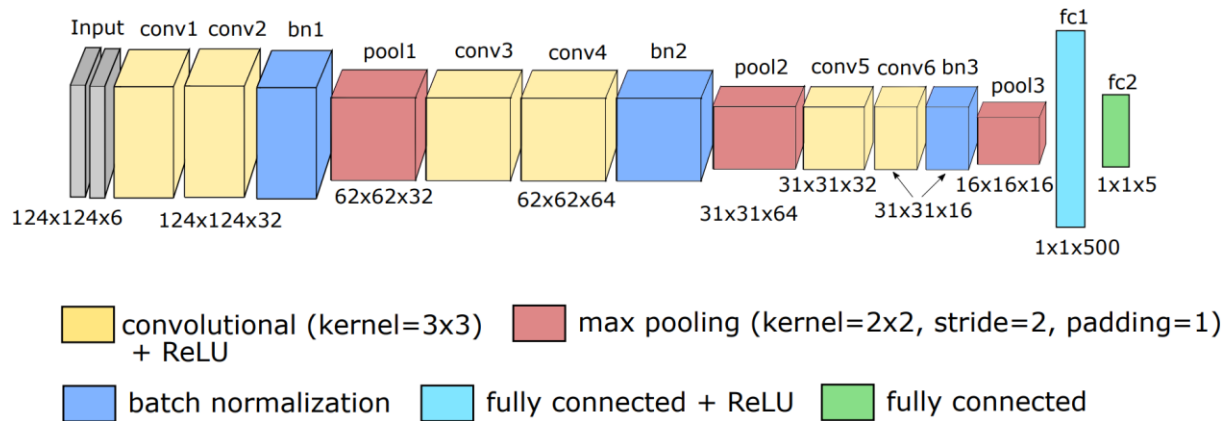


Figure 3: Network architecture of the CNN applied. The input (124x124x6) is composed of the fused mean image and normal map. The architecture structure consists of two consecutive convolutional layers, a batch normalization layer and max pooling layer. The classification is performed by two fully connected layers. The output of the last layer corresponds to the number of defect classes.

Augmentation algorithms were used to balance and augment the training dataset. All image data of the training dataset were enlarged by augmentation and for defect classes with limited data, the image data were augmented more often than, for example, image data of the OK class. For data augmentation, horizontal flips, rotations, blurring and affine transformations were performed. Furthermore, adjustments were made for brightness, contrast and intensities in the hue channel. Figure 4 shows the loss of training and validation. During training, the networks weights were saved at each epoch. For the evaluation, the network configuration of the CNN were used where the validation loss was the smallest. This was epoch 64. The results of the classification CNN are shown in Figure 5 in the form of a confusion matrix. It can be seen that the defect class discoloration provides a very good prediction accuracy, although this class has the smallest amount of image data available. Misclassification occurs especially with contamination but also with dents. These defects are mistaken for the OK class. This is possibly due to the fact that the dataset also contains difficult-to-detect contamination defects and very small expressions of the defect class dents. With this configuration, a defect detection accuracy of 95% could be achieved. The Matthews correlation coefficient (MCC) is 92%.

4. CONCLUSION

In this publication we present an inspection system for the detection of defects on the gold areas of PCBs. A photometric stereo setup is used to visualize the defects on the highly reflective gold surface and a normal map is calculated. All defects except for the defect discoloration are visible in the normal map. A mean image is calculated from the available image acquisitions, in which the defect classes discoloration and contamination are clearly visible. A classification

CNN receives normal map and mean image in the form of a 6-dimensional image as input. On the test dataset an accuracy of 95% and MCC value of 92% could be achieved. One of our goals is to improve the calibration method so that a more uniform normal map can be computed. Furthermore, we want to test different input images and compare the classification accuracies.

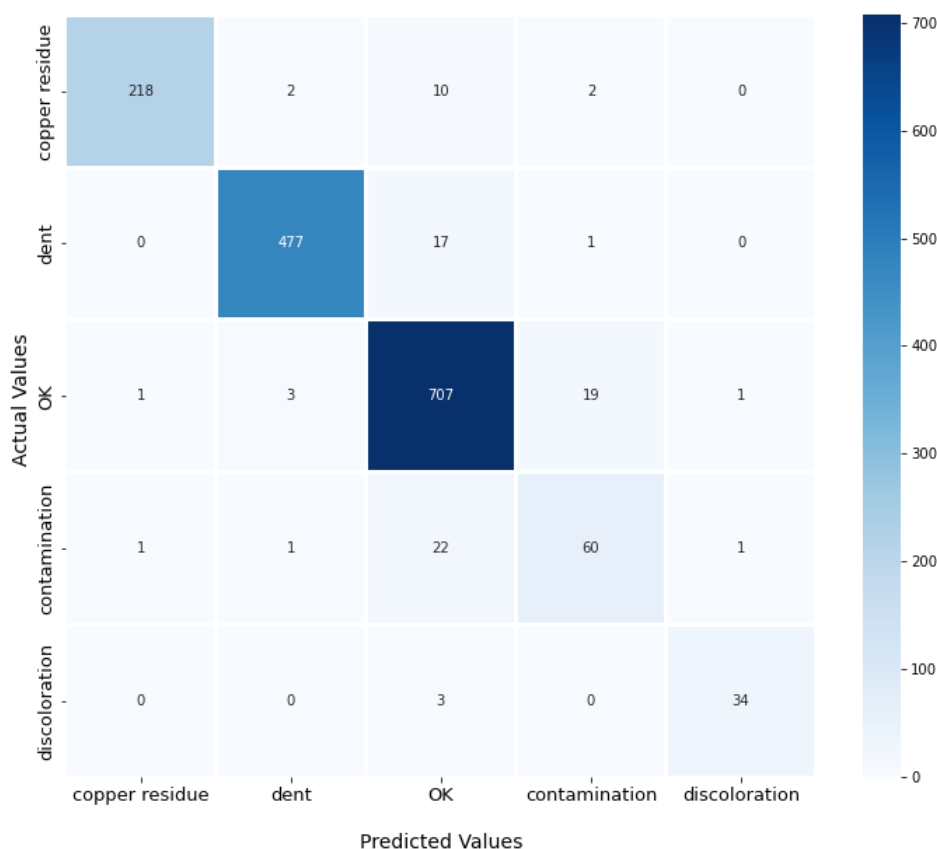


Figure 5: Confusion matrix of defect classes on the test dataset. The dataset is unbalanced. Many misclassifications occur with contamination. The best results are obtained with discoloration.

5. ACKNOWLEDGMENTS

The research work of this article was performed at the Polymer Competence Center Leoben GmbH (PCCL, Austria) within the framework of the COMET-program of the Federal Ministry for Transport, Innovation and Technology and the Federal Ministry of Digital and Economic Affairs with contributions by AT&S (Austria Technologie & Systemtechnik AG). The PCCL is funded by the Austrian Government and the State Governments of Styria, Lower Austria, and Upper Austria.

6. REFERENCES

- [Cao+22] Yanlong C., Binjie, D., Jingxi, C., Wenyan, L., Penging, G., Liuyi, H., Jiangxin, Y. Photometric-Stereo Based Defect Detection System for Metal Parts. *Sensors* 22.21 (2022), p. 8374. <https://doi.org/10.3390/s22218374>
- [Cer+20] Cerezci, F., Kazan, S., Oz, M. A., Oz, C., Tasci, T., Hizal, S., Altay, C. Online metallic surface defect detection using deep learning. *Emerging Materials Research* 9.4 (2020), pp. 1266–1273. <https://doi.org/10.1680/jemmr.20.00197>
- [Hab+22] Hable, A., Tabatabai, P., Lichtenegger, H. L., Scherr, A., Krivec, T., Gruber, D. P. Detection of Printed Circuit Board Defects on ENIG and ENIG Surface Finishes with Convolutional Neural Networks and Evaluation of Training Parameters. *Journal of Microelectronics and Electronic Packaging* 19.4 (2022), pp. 123–130. <https://doi.org/10.4071/imaps.1814291>
- [Har+20] Harris C.R. et al. Array programming with NumPy. 2020. <https://doi.org/10.1038/s41586-020-2649-2>.
- [LOK19] Lee, J. H., Oh, H. M., & Kim, M. Y. Deep learning based 3D defect detection system using photometric stereo illumination. 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC). IEEE. 2019, pp. 484–487. <https://doi.org/10.1109/ICAIIIC.2019.8669005>

- [Pas+19] Paszke, A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems* 32. Curran Associates, Inc., 2019, pp. 8024–8035. URL: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [PMM22] Pourmand, S., Merillou, N., Merillou, S. Depth Completion for Close-Range Specular Objects. *WSCG 2022: full papers proceedings: 30. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, p. 135–141. <https://www.doi.org/10.24132/CSRN.3201.17>
- [Pod+17] Podrekar, G., Tomaževič, D., Likar, B., Usenik, P. Model based visual inspection of pharmaceutical tablets with photometric stereo. In: *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*. IEEE, 2017, pp. 133–136. <https://doi.org/10.23919/MVA.2017.7986819>
- [Ren+20] Ren, X., Wang, W., Ren, J., Mao, X., Zhang, M. Research and application of label defect detection method based on machine vision. *Journal of Physics: Conference Series*. Vol. 1453. 1. IOP Publishing, 2020, p. 012084. <https://www.doi.org/10.1088/1742-6596/1453/1/012084>
- [Sai+22] Saiz, F. A., Barandiaran, I., Arbelaiz, A., Graña, M. Photometric stereo based defect detection system for steel components manufacturing using a deep segmentation network. *Sensors* 22.3 (2022), p. 882. <https://doi.org/10.3390/s22030882>
- [VW] Verma, C.S. and Wu, M.J. Photometric Stereo. URL: https://pages.cs.wisc.edu/~csverma/CS766_09/Stereo/stereo.html.
- [Woo80] Woodham, R.J. Photometric method for determining surface orientation from multiple images. *Optical engineering* 19.1 (1980), pp. 139–144

A new deterministic gasket fractal based on ball sets

Roberto Soto-Villalobos
Universidad Autónoma
de Nuevo León, Facultad
de Ciencias de la Tierra
Carretera a Cerro Prieto
km 8.0
67700, Linares, Mexico
roberto.sotovll@uanl.edu.mx
ORCID:0000-0002-3172-
8673

Francisco Gerardo
Benavides-Bravo
Tecnológico Nacional de
México, Instituto
Tecnológico de Nuevo
León
Av Eloy Cavazos 2001
67170, Guadalupe,
Mexico
francisco.bb@nuevoleon.tecnm.mx
ORCID:0000-0002-4596-
831X

Filiberto
Hueyotl-Zahuantitla
Cátedra
CONACyT-UNACH
Carretera Emiliano
Zapata km 8.0
29050, Tuxtla Gutiérrez,
Mexico
fhueyotl@conacyt.mx
ORCID:0000-0002-5527-
7141

Mario A. Aguirre-López*
Universidad Autónoma
de Chiapas, Facultad de
Ciencias en Física y
Matemáticas
Carretera Emiliano
Zapata km 8.0
29050, Tuxtla Gutiérrez,
Mexico
marioal1906@gmail.com
ORCID:0000-0002-5191-
3462
*Corresponding author

ABSTRACT

In this paper, we propose a new gasket fractal constructed in a deterministic iterated function system (IFS) way by means of interacting ball and square sets in \mathbb{R}^2 . The gasket consists of the ball sets generated by the IFS, possessing also exact self-similarity. All this leads to a direct deduction of other properties and a clear construction methodology, including a dynamic geometry procedure with an open-source construction protocol. We also develop an extended version of the fractal in \mathbb{R}^n . Some resulting configurations consisting of stacked 2D-fractals are plotted. We discuss about potential applications of them in some areas of science, focusing mainly on percolation models. Guidelines for future work are also provided.

Keywords

gasket fractal, n -sphere, iterated function systems, box-counting dimension, dynamical geometry, percolation

1 INTRODUCTION: BACKGROUND AND NOTATION

Fractals are geometrical shapes made up of smaller and smaller elements than add roughness to the entire shape. Talking about its length is not clear since there will always be something finer that will escape the sensitivity of the instrument used, increasing or decreasing this measurement. Then, they should be measured by borders, polygons, balls, boxes or new concepts that go beyond classical geometric concepts.

In other words, when we measure any shape by choosing different measurement scales, power law relationships written by $N = s^D$ are fulfilled, with N as the number of segmented figures, s the similarity dimension, and D the fractal dimension. If the shape satisfies the above relation with D as a

non-entire number, we are dealing with a fractal [B.B83]. Among those structures, gasket fractals are defined such as those ones that are constructed by joints of sets.

Gasket fractals have become a source of inspiration for the design of several digital and physical structures [Gua18, AHHN21, Rao21, FFSS22]. Some of the most popular fractals within this kind are generated by applying an iterated function system (IFS) to a deterministic formulation, such as the Koch snowflake [Koc04], the Sierpinski triangle [Sie15], the T-square [DJW16], the Cantor dust [Can70, Can71], and its two-dimensional and three-dimensional versions: the Sierpinski carpet [Sie16] and the Menger sponge [Edg04], respectively.

Our fractal design belongs to the the above sub-kind of construction. It is defined in an Euclidean space in \mathbb{R}^2 . Two types of sets are the basis of the generating IFS: on the one hand, $B(c, r)$ refers, as usual, to a closed ball with radius r that is centered at c , $r \in \mathbb{R}$, $c \in \mathbb{R}^2$; on the other hand, we refer by $S(c, r)$ to the set composed by all the points inside and on the borderline of a square, a “closed-square”, with side $2r$ that is centered at c . In this way, both kind of sets B and S , are defined by

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

their center position and radius of the maximum circumscribed circle inside the square. The paper is structured as follows: fractal's definition and its main properties are introduced in Section 2, for 2D and higher dimensions; the method of construction, both pseudo-code and procedure by dynamical geometry using open-source software, is detailed in Section 3; Section 4 exemplifies the potential applications of some resulting fractal arrangements and discuss about other potential uses; finally, concluding remarks and guidelines for future works are discussed in Section 5.

2 MODELING THE FRACTAL

Let the initial set $S(\kappa_0, \rho_0)$ and its largest circumscribed ball $B(\kappa_0, \rho_0) \equiv F_0$, $\kappa_0 \in \mathbb{R}^2$, $\rho_0 \in \mathbb{R}$. At the first iteration, $S(\kappa_0, \rho_0)$ is divided into the four largest squares $S(\kappa_i, \rho_1)$, $i = 1, 2, 3, 4$, each of them intersecting $B(\kappa_0, \rho_0)$ in only one point $x_i \in \mathbb{R}^2$, $\bigcap_{i=1}^4 x_i = \emptyset$. The couples of parameters defining each generated i -th square are obtained by the fulfillment of the following constraints:

$$\kappa_i \in S(\kappa_0, \rho_0), \quad \kappa_i \notin B(\kappa_0, \rho_0), \quad \bigcap_{i=1}^4 k_i = \emptyset \quad (1)$$

$$\rho_1 = \frac{1}{2} \sup \left\{ \delta : \left(S(\kappa_i, \delta) \subset S(\kappa_0, \rho_0), \right. \right. \\ \left. \left. S(\kappa_i, \delta) \cap B(\kappa_0, \rho_0) = x_i \right) \right\}, \quad \forall \delta > 0. \quad (2)$$

Due to symmetry, all the generated squares $S(\kappa_i, \rho_1)$, $i = 1, 2, 3, 4$, have the same radius length ρ_1 , and are equidistant to the center of their generator square $S(\kappa_0, \rho_0)$ in direction to its vertices, going through its respective x_i , see the representation in Fig. 1. So, the sub-index for the radius length goes with the iteration number. All this leads to the respective generated balls $B(\kappa_i, \rho_1)$, $i = 1, 2, 3, 4$, which make up F_1 .

Now, this function system is extended to the m -th iteration, $m \geq 1$, by substituting $S(\kappa_0, \rho_0)$ with each j -th square generated at the $(m-1)$ -th iteration, and applying Eqs. (1) and (2) to them. Since each new ball becomes a generator of four balls of lower size but possessing the property of exact self-similarity, we can easily calculate that generated number of balls $N_m = 4^m$ balls with radius $\rho_m = \rho_0 u^m$, with $u = (1 - \sqrt{2}/2)/2$, at the m -th iteration. Those power-law behaviors are shown in a logarithmic view in Fig. 2.

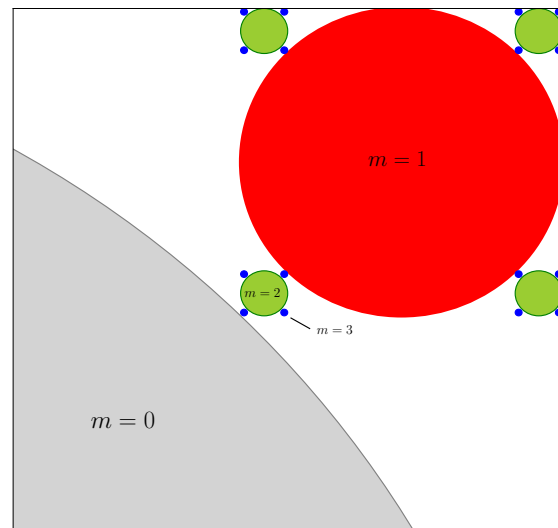


Figure 1: Top-right view of the fractal, the first three iterations are shown. The initial ($m = 0$) circumscribed gray ball $B(\kappa_0, \rho_0)$ generates four squares as described in the text, each of which leads a new level ($m = 1$) of circumscribed red balls $B(\kappa_{1:4}, \rho_1)$. Each red ball generates a new level ($m = 2$) of squares with their respective circumscribed green balls $B(\kappa_{5:20}, \rho_2)$. The process continues taking the green balls as generators, four blue balls ($m = 3$) generates for each green ball, and so on.

Other properties, such as the cumulative area A_m and the fractality D of the gasket can be directly calculated from ρ_m and N_m . Namely:

$$A_m = \sum_{i=0}^m N_i \pi (\rho_i)^2 = \pi (\rho_0)^2 \sum_{i=0}^m 4^i u^{2i}, \quad (3)$$

being the last expression in terms of the initial radius ρ_0 . Then, taking $m \rightarrow \infty$ and normalizing Eq. (3) with respect to $S(\kappa_0, \rho_0)$, it becomes

$$A = \left(\frac{1}{(2\rho_0)^2} \lim_{m \rightarrow \infty} A_m \right) \times 100\% \approx 86\%. \quad (4)$$

In turn, since our gasket is deterministic and exact self-similar, the fractal dimension can be computed by a simple formula related to the box-counting formulation [Fal90]:

$$D = \frac{\ln v}{\ln \frac{1}{u}} = \frac{\ln 4}{\ln \frac{2}{1-\sqrt{2}/2}} = 0.72, \quad (5)$$

where v takes its value from the number of balls generated per square. In this way, the fractality of our gasket is lower than other known 2D self-similar gaskets, such as the Sierpinski triangle ($D_{ST} = 1.59$), the Sierpinski carpet ($D_{SC} = 1.89$),

the T-square ($D_{T-square} = 1.58$), and the Koch curve ($D_{Koch} = 1.26$). Indeed, it has a fractal dimension about the 1D Cantor dust ($D_{Cantor} = 0.63$).

Definition. Let $F_{m,q}$, the q -th ball generated at the m -th iteration of the IFS described above, then

$$F_m = \bigcup_{q=1}^{4^m} F_{m,q} \quad (6)$$

and the ball-gasket fractal consists of

$$F = \lim_{m \rightarrow \infty} F_m. \quad (7)$$

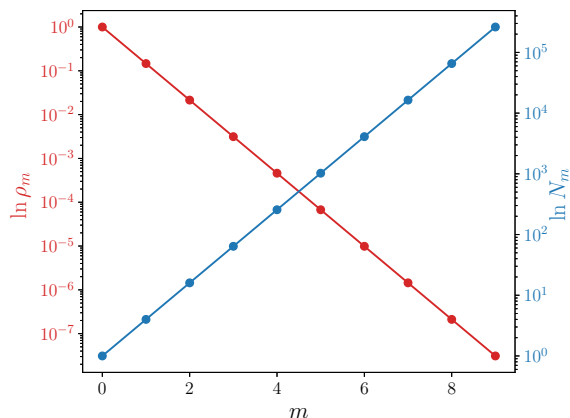


Figure 2: In red- logarithm on the radius ρ_m of the circumscribed balls as a function of the number of iterations m , assuming $\rho_0 = 1$. In blue- logarithm of the number of generated balls N_m in each iteration m .

Extension to \mathbb{R}^n

The construction method of F allows to formulate it for an euclidean space in \mathbb{R}^n . The first step is to extend our definitions for squares and balls to $S^n(c, r)$ and $B^n(c, r)$, in which they refer to the set of points inside an n -cube and an n -sphere, respectively, centered at $c \in \mathbb{R}^n$ and having radius $r \in \mathbb{R}$, i.e., the nomenclature for the original case in \mathbb{R}^2 would be $S^2(c, r)$ and $B^2(c, r)$.

Then, the number of the generated n -spheres depends on the number of vertices of the n -cubes: 2^n for the n -dimensional case [Cox74], so that a total of $N_m^n = 2^{nm}$ balls are generated at the m -th iteration, each one with different intersection point $x_i \in \mathbb{R}^n$, $\bigcap_{i=1}^{2^n} x_i = \emptyset$, and Eq. (1) becoming

$$\kappa_i \in S(\kappa_0, \rho_0), \quad \kappa_i \notin B(\kappa_0, \rho_0), \quad \bigcap_{i=1}^{2^n} \kappa_i = \emptyset. \quad (8)$$

Thus, our n -dimensional ball-gasket fractal is denoted by

$$F^n = \lim_{m \rightarrow \infty} F_m^n, \quad (9)$$

where

$$F_m^n = \bigcup_{q=1}^{2^{nm}} F_{m,q}^n, \quad (10)$$

and $F_{m,q}^n$ represents q -th ball generated at the m -th iteration.

Finally, since the number of generated balls per iteration changes with n but not the ball radius, the properties of the fractal must be also re-calculated for the n -dimensional case:

$$A_m^n = \sum_{i=0}^m N_i^n \frac{\pi^{n/2} (\rho_i)^n}{\Gamma(n/2 + 1)} = \frac{\pi^{n/2} (\rho_0)^n}{\Gamma(n/2 + 1)} \sum_{i=0}^m 2^{ni} u^{ni}, \quad (11)$$

where $\Gamma(\cdot)$ is the Euler's gamma function, and the second factor in the sum computes the area of an n -sphere [NIS]. In turn, the box-counting dimension for the n -dimensional case (D^n) changes with the number of generated n -sphere per each n -cube (v^n), so that

$$D^n = \frac{\ln v^n}{\ln \frac{1}{u}} = \frac{\ln 2^n}{\ln \frac{2}{1-\sqrt{2}/2}}. \quad (12)$$

3 METHODOLOGY FOR 2D CONSTRUCTION

The pseudo-code for the F^2 's construction is shown in Algorithm 1, according to the definition introduced in Section 2. The algorithm is designed in such a way that it not only draws the fractal (line 19) but saves the information of each generated circle, such as its radius (line 10), center (lines 15:18), and area (lines 6,21-22). Then, the outputs are the fractal F^2 , its parameters $\kappa_{0:index}$, $\rho_{0:m}$, and the normalized cumulative area A .

In turn, Algorithm 2 presents a set of steps to generate F_1^2 , up to the 1-st iteration, based on dynamic geometry via the open-source software Geogebra [HBA⁺13]. This way of construction could be reached with using only the eight buttons shown in Fig. 3, so that it is useful for educational purposes. Fig. 4 also illustrates the fractal up to $m = 1$, while the reader could refer to the *Supplementary material* for the construction protocol to generate it up to the 3-th iteration, taking $\rho_0 = 1$ units and $\kappa_0 = (0, 0)$.

Algorithm 1 Pseudo-code for F^2

```

1: procedure GENERATOR OF  $F^2(\kappa_0, \rho_0)$ 
2:   System Initialization ▷ Set  $m$ 
3:   Read the entry values
4:   newBalls=1 ▷ Balls at  $m = 0$ 
5:   index=newBalls ▷ cumulative count
6:    $A = 1$  ▷ Initialize the norm. cum. area
7:   for  $i$  in  $1:m$  do ▷ Start the iterations
8:     lastBalls=newBalls ▷ Generators
9:     newBalls= $4^i$  ▷ To generate
10:     $\rho_i = \rho_0 u^i$ 
11:    for  $j$  in  $1:lastBalls$  do ▷ Start to generate
12:       $G = index - lastBalls + j$ 
13:       $k = index + 4(j - 1)$ 
14:       $\delta = \rho_{i-1} - \rho_i$ 
15:       $\kappa_{k+1}^x = \kappa_G^x + \delta; \kappa_{k+1}^y = \kappa_G^y + \delta$ 
16:       $\kappa_{k+2}^x = \kappa_G^x - \delta; \kappa_{k+2}^y = \kappa_G^y + \delta$ 
17:       $\kappa_{k+3}^x = \kappa_G^x - \delta; \kappa_{k+3}^y = \kappa_G^y - \delta$ 
18:       $\kappa_{k+4}^x = \kappa_G^x + \delta; \kappa_{k+4}^y = \kappa_G^y - \delta$ 
19:      Draw  $B(\kappa_{k+l}, \rho_i)$  ▷ for  $l = 1 : 4$ 
20:    index=index+newBalls
21:     $A = A + 4^i u^{2i}$  ▷ Computing with Eq. (3)
22:     $A = \frac{\pi A}{4} \times 100\%$  ▷ Normalizing with Eq. (4)
23:  output:  $F^2, \kappa_0:index, \rho_0:m, A$ 

```

Algorithm 2 Construction of F_1^2 in GeoGebra.

```

1: procedure GENERATOR OF  $F_1^2(\kappa_0, \rho_0)$ 
2:   System Initialization ▷ Use
   the command SetAxesRatio(1,1) in order
   to fix a scale 1:1
3:   Draw a square centered at  $\kappa_0$  with side  $2\rho_0$ ,
   using button a) ▷
   Generating the points  $(-\rho_0, \rho_0) + \kappa_0, (\rho_0, \rho_0) +$ 
 $\kappa_0, (\rho_0, -\rho_0) + \kappa_0, (-\rho_0, -\rho_0) + \kappa_0$ 
4:   Draw a polygon using button b) ▷
   By joining all the previous points, starting and
   ending at the same point
5:   Find a midpoint of any side of the square
   with button c)
6:   Using button d), draw a circle with  $\kappa_0$  as its
   center, and the radius of the midpoint found
   previously
7:   Draw the diagonals of the square with but-
   ton e) ▷ At this step, the resulting picture is
   shown in Fig. 4 (a)
8:   Find the intersections of the circle with the
   diagonals with button f)
9:   Draw the perpendicular bisectors between
   the corners of the square and the intersections
   of the diagonal with the circle by means of but-
   ton g)
10:  Find the intersections of the sides of the
   square with the perpendicular bisectors using
   button f)
11:  Reflect the square corresponding to the 1st
   quadrant up to generate the rest of the new
   squares with button h). ▷ The resulting
   picture is shown in Fig. 4 (b).
12:  Note: To continue up to the  $i$ -th iteration,
   repeat steps 4 to 10 for the generated circles,
   substituting  $\kappa_0$  and  $\rho_0$  with the correspond-
   ing center and radius of each generated circle.
   Then continue to Step 11.

```









a) 	Point
b) 	Polygon
c) 	Midpoint or Center
d) 	Circle with Center through Point
e) 	Segment
f) 	Intersect
g) 	Perpendicular Bisector
h) 	Reflect about line

Figure 3: Table of buttons used for the geometrical construction in Geogebra.

4 RESULTING CONFIGURATIONS AND POTENTIAL APPLICATIONS

Although the objective of this work is not to delve into a direct application of the fractal, in this section we provide the reader some potential lines of research in which it could be used.

4.1 Stacked sets for percolation

Since its squared delimitation, diverse structured configurations or arrangements can be obtained from stacking F^2 sets. Fig. 5 (a) shows the most basic stack, which consists of an intersected structured in multiple scales. Indeed, each main generator ball $B(\kappa_0, \rho_0)$ touches its four-nearest similar

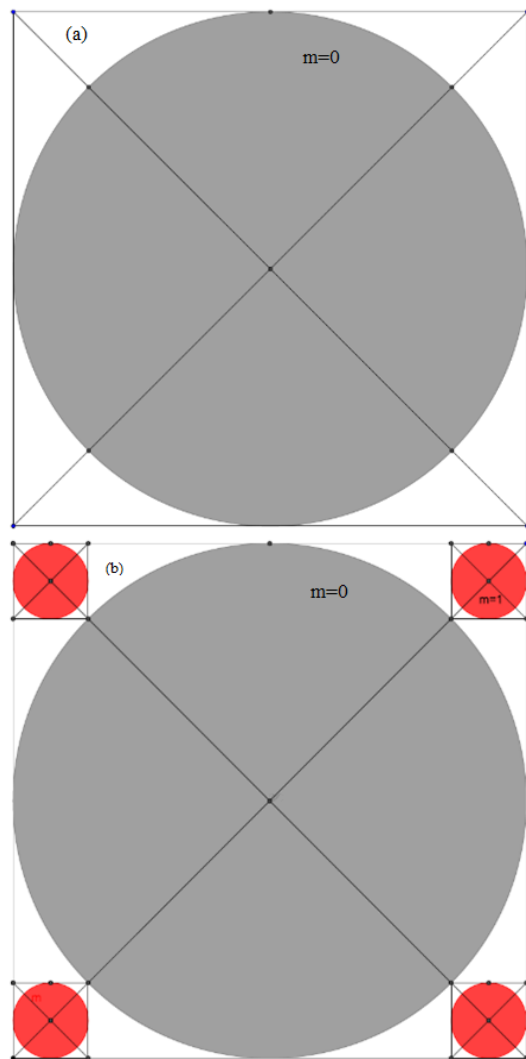


Figure 4: Geometrical construction in Geogebra. (a) Steps up to $m = 0$. (b) Steps up to $m = 1$.

neighbors; for the i -th iteration, $i > 0$, the generated balls in a corner of its main generator square ($S'(\kappa_0, \rho_0)$) touch their two-nearest similar neighbors from their corresponding main generators; in turn, the generated balls that are in contact with the border of its generator square, without being in a corner, touch only one similar neighbor.

In percolation terms, picture in Fig. 5 (a) defines a 2D slice of a porous but not-permeable structure (in the shown 2D slice) at different scales. In tact granite is a rock example that approximates to those characteristics [SBN05]. In the same line, Fig. 5 (b) shows an alternative configuration consisting of an overlapping of the fractals of Fig. 5 (a), by matching their nearest corners at their 1-st iteration. This also reduces porosity, because of the contraction of the stack, allowing variation in the modelling.

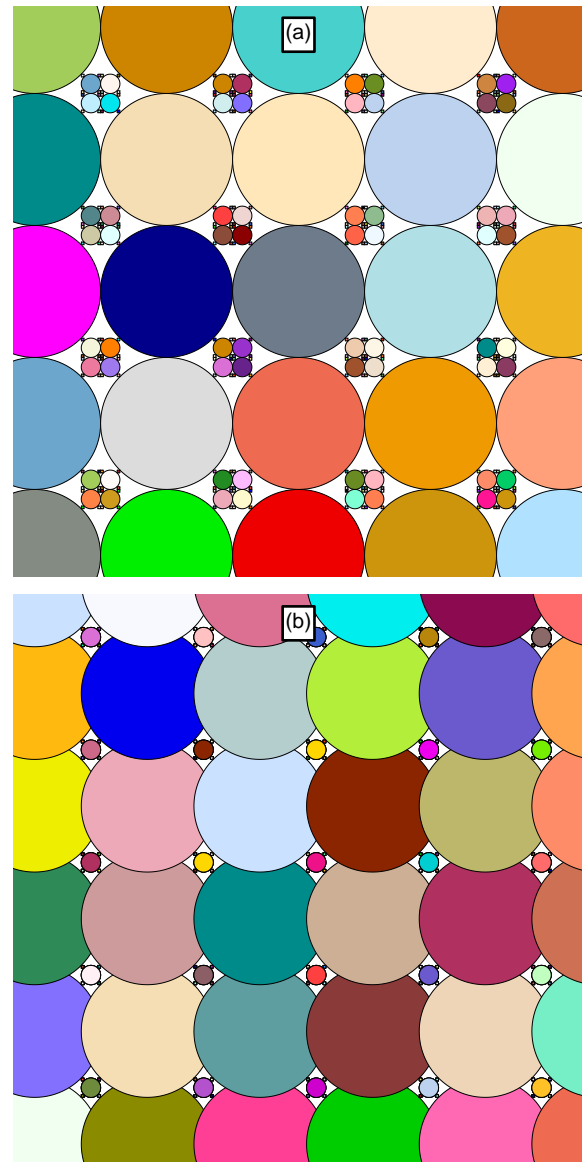


Figure 5: Porous and non-permeable stacks of fractals plotted with random colors up to the 3-rd iteration. (a) Most basic configuration. (b) Implementing an overlapping at the 1-st iteration.

Now, permeability could be introduced under the same reasoning by:

- adding a translation (ϕ) to interspersed lines of the stack, like that with $\phi = \rho_0/2$ in upward direction shown in Fig. 6 (a). Different from pictures in Fig. 5, this configuration only allows connected with two neighbors between main generators, while the rest of the balls do not touch each other, leading to a low permeability in mainly one direction but remaining the filled area in 86%. Similar structures are found in cracked crystalline rocks [NM14], and clay sediments making non-permeable beds [Car39].

- randomly selecting some balls not to draw. This leads to unknown paths of interconnected porous, reducing the filling area in dependence on the probability of drawing (p). Fig. 6 (b) shows an implementation of that kind of configuration with $p = 0.5$, which could be adaptable to make several shapes according to the geological structure to simulate or characterize. It could be useful for the case of sandstone, whose permeability depends of its relative composition [ZSDL15].
- a combination of the techniques mentioned previously.

In contrast to similar fractal models, like the space-filling packing from inversion of circles/spheres, that leave no porosity on the limit of infinitesimally small spheres [Ley05, SH18], our original fractal is a porous structure even in the limit, and can be easily modified to adjust the level of porosity.

Based on the above, our fractal would serve as a basis model for 2D or 3D rendering of rocks with a similar structure and/or fractal dimension, applying it as a direct modeling after the characterization of the rock [LWXY⁺22]. That methodology has been implemented previously to construct tessellations from iterative rules but not involving fractals [NM14, LMH10]. So, it could be a novel variation to improve that methodology, expecting some advantages such as the quick construction, and easy and deterministic control of the minimum-maximum size (by the iterations number). Nevertheless there are some limitations such as the fixed location and size of new circles, at the current version of our fractal, without modifying it or adding noise.

Figs. 5 and 6 were drawn by implementing the pseudo-code of Algorithm 1 in Rstatistics software [R C21].

4.2 Potential uses

In practice, the fractal dimension can also be determined by using the square side instead of the ball radius in Eqs. (5) and (12), leading to $D = 1.13$ and $D = 1.69$ for an extension to 3D. In this section we mention some similar dimensions and possible uses of our fractal. It is interesting that our fractal dimension is close to the value obtained ($D = 1.08 \pm 15$) by [Nos93], who indicates that the reason of the power-law form of the radio emission spectrum can be due to the fractal nature of the spatial distribution of radiating electrons; the value is also close to the fractal dimension of the perimeter ($D \sim 1.35$) of molecular clouds

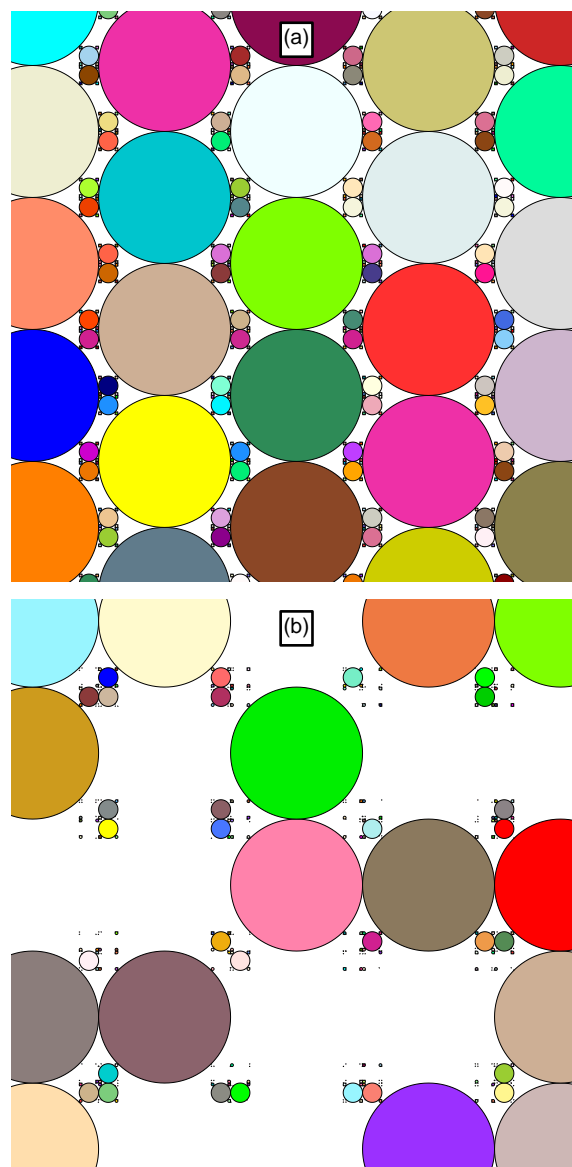


Figure 6: Porous and permeable stacks of fractals plotted with random colors up to the 3-rd iteration. (a) Structured modification applying a translation of $\phi = \rho_0/2$ in upward direction to interspersed lines. (b) Random modification by adding a probability of drawing $p = 0.5$ to each ball.

from two-dimension maps, see [SAP05] and references therein. These similarities on the fractal dimension suggest a similar level of roughness or sponginess, fact that could be useful to set conditions for modeling the phenomenon in the corresponding parameter space.

On the other hand, the relative low value of our dimension fractal is close to value, $D \sim 1.0 - 1.1$, for the corrosion-induced cracks in reinforced concrete (see, [JJZ⁺20]). In this case, variants of our fractal, as these in previous section, can have potential use to model different types of coarse ag-

gregate distribution to analyze (or even prevent) crack patterns. Last, but not least important is that our fractal could be taken as basis to generate garment geometrical patterns for clothing fabrics as in [Lam17] and [WZYW19]; furthermore, the 3D version could be used as an scene, like the sphere-flake used by [KML16], to test rendering methods. Here it is important to mention the existence of meshes based on fractals, some of them implemented in methodologies like the Delaunay triangulation using the Sierpiński triangle [BC92], and other ones exploring how phenomena occur with the developed fractal array, based on the Cantor set [SIK17]. In this sense, our fractal is more related to the latter one, due to the gaps in the filling space.

Finally, let the above mentioned potential uses, it is important to mention two particular characteristics of the fractal:

- It is difficult to appreciate the aesthetics of the fractal because the radius decreases almost one order in magnitude per iteration, see Fig. 2, so that the generated balls quickly disappear to the human-eye.
- the entire space is not fully filled, $A = 86\%$, this could be an advantage or disadvantage according to its use, and also contrast to others fractals involving a ball construction, such as the Apollonian fractal [Bou06].

5 CONCLUSIONS AND FUTURE WORK

We introduced a new gasket fractal consisting of 2D-balls embedded in squared sets, and constructed by means of a deterministic IFS. The fractal is exact-self similar, with a normalized cumulative area of 86%, and a box-counting dimension of $D = 0.72$, which differs from several well-known 2D fractals. Additionally, the gasket has the property of extend itself to \mathbb{R}^n while preserving a similar formulation to calculate its properties.

Beyond its limitations, our fractal possesses useful features that allow to apply it to diverse areas. In this way, we recommend to explore its fitting to represent percolation models and other topics inside numerical computational geometry.

Additionally to its applications, our gasket construction lays the basis for the definition of square-shaped fractals that are involved in the IFS of this work. A fractal composed of the generator squares, and two more based on the centers κ_i and in the intersection points x_i , are some examples of possible future lines of research.

ACKNOWLEDGMENTS

F.H.-Z. thanks FCFM-UNACH and the support from CONACyT through the program “Investigadoras e investigadores por México”, Cátedra 873. M.A.A.-L. thanks CONACyT for the postdoctoral grant 839412 and FCFM-UNACH for supporting his research stay.

6 REFERENCES

- [AHHN21] Ruaa Shallal Abbas Anooz, Ghufraan M Hatem, Iman Hafedh Yaseen Hasnawi, and Mohammed N Nemah. Design apollonian gasket antenna for millimeter-wave applications. *IOP Conference Series: Materials Science and Engineering*, 1094(1):012038, feb 2021.
- [B.B83] Mandelbrot B.B. *The Fractal Geometry of Nature*. W.H. Freeman, 1983.
- [BC92] S. W. Bova and G. F. Carey. Mesh generation/refinement using fractal concepts and iterated function systems. *International Journal for Numerical Methods in Engineering*, 33(2):287–305, 1992.
- [Bou06] Paul Bourke. An introduction to the apollonian fractal. *Computers & Graphics*, 30(1):134–136, 2006.
- [Can70] G. Cantor. Beweis, daß eine für jeden reellen werth von x durch eine trigonometrische reihe gegebene function $f(x)$ sich nur auf eine einzige weise in dieser form darstellen läßt, part 1. *Journal für die reine und angewandte Mathematik*, 72:139–142, 1870.
- [Can71] G. Cantor. Beweis, daß eine für jeden reellen werth von x durch eine trigonometrische reihe gegebene function $f(x)$ sich nur auf eine einzige weise in dieser form darstellen läßt, part 2. *Journal für die reine und angewandte Mathematik*, 73:294–296, 1871.
- [Car39] P. C. Carman. Permeability of saturated sands, soils and clays. *The Journal of Agricultural Science*, 29(2):262–273, 1939.
- [Cox74] H.S.M. Coxeter. *Regular complex polytopes*. Cambridge University Press, 1974.
- [DJW16] Nell Dale, Daniel T. Joyce, and Chip Weems. *Object-Oriented Data Structures Using Java*. Jones & Bartlett Learning, 2016.

- [Edg04] Gerald A. Edgar. *English translation of: "K. Menger (1926), Allgemeine Räume und Cartesische Räume. I., Communications to the Amsterdam Academy of Sciences"*. Westview Press, 2004.
- [Fal90] Kenneth Falconer. *Fractal geometry: mathematical foundations and applications*. John Wiley, 1990.
- [FFSS22] Z. Fair, M. Flanner, A. Schneider, and S. M. Skiles. Sensitivity of modeled snow grain size retrievals to solar geometry, snow particle asphericity, and snowpack impurities. *EGU-sphere*, 2022:1–22, 2022.
- [Gua18] Emanuel Guariglia. Harmonic sierpinski gasket and applications. *Entropy*, 20(9), 2018.
- [HBA⁺13] M. Hohenwarter, M. Borchers, G. Ancsin, B. Bencze, M. Blossier, A. Delobelle, C. Denizet, J. Éliás, Á Fekete, L. Gál, Z. Konečný, Z. Kovács, S. Lizelfelner, B. Parisse, and G. Sturr. GeoGebra 4.4, December 2013. <http://www.geogebra.org>.
- [JJZ⁺20] Haodong Ji, Haoyu Jiang, Ruoyi Zhao, Ye Tian, Xianyu Jin, Nanguo Jin, and Jing Tong. Fractal characteristics of corrosion-induced cracks in reinforced concrete. *Materials*, 13:3715, 08 2020.
- [KML16] Kevin Keul, Stefan Müller, and Paul Lemke. Accelerating spatial data structures in ray tracing through pre-computed line space visibility. In *WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016*, pages 17–25, 2016.
- [Koc04] H.V. Koch. Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire. *Arkiv for Matematik, Astronomi och Fysik*, 1:681–704, 1904.
- [Lam17] Artde Donald Kin-Tak Lam. A study on fractal patterns for the textile design of the fashion design. In *2017 International Conference on Applied System Innovation (ICASI)*, pages 676–678, 2017.
- [Ley05] Jos Leys. Sphere inversion fractals. *Computers & Graphics*, 29(3):463–466, 2005.
- [LMH10] Hengxing Lan, C. Derek Martin, and Bo Hu. Effect of heterogeneity of brittle rock on micromechanical extensile behavior during compression loading. *Journal of Geophysical Research: Solid Earth*, 115(B1), 2010.
- [LWXY⁺22] Wei Lin, Zhenkai Li Wu, Zhengming Xizhe Yang, Mingyi Hu, Denglin Han, Chenchen Wang, and Jizhen Zhang. Digital characterization and fractal quantification of the pore structures of tight sandstone at multiple scales. *Journal of Petroleum Exploration and Production Technology*, 12:2565–2575, 2022.
- [NIS] NIST Digital Library of Mathematical Functions. 5.19 mathematical applications. Accessed: December 27, 2022.
- [NM14] M. Nicksiar and C.D. Martin. Factors affecting crack initiation in low porosity crystalline rocks. *Rock Mechanics and Rock Engineering*, 47:1165–1181, 2014.
- [Nos93] M. D. Noskov. Influence of the fractal nature of the spatial distribution of radiating electrons on a cosmic radio emission spectrum. *Astronomy Reports*, 37(5):565–566, September 1993.
- [R C21] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.
- [Rao21] Nukala Srinivasa Rao. Design and analysis of koch snowflake geometry with enclosing ring multiband patch antenna covering s and l band applications. In Vijay Nath and J.K. Mandal, editors, *Nanoelectronics, Circuits and Communication Systems*, pages 167–176, Singapore, 2021. Springer Singapore.
- [SAP05] Néstor Sánchez, Emilio J. Alfaro, and Enrique Pérez. The fractal dimension of projected clouds. *The Astrophysical Journal*, 625(2):849, jun 2005.
- [SBN05] A. Selvadurai, M. Boulon, and T. Nguyen. The permeability of an intact granite. *Pure and Applied Geophysics*, 162:373–407, 2005.
- [SH18] D. V. STÄGER and H. J. HERRMANN. Self-similar space-filling sphere packings in three and four dimensions. *Fractals*, 26(03):1850022, 2018.
- [Sie15] Waclaw Sierpiński. Sur une courbe

- dont tout point est un point de ramification (in french). *Comptes rendus de l'Académie des Sciences*, 160:302–305, 1915.
- [Sie16] Waclaw Sierpiński. Sur une courbe cantorienne qui contient une image biunivoque et continue de toute courbe donnée (in french). *Comptes rendus de l'Académie des Sciences*, 162:629–632, 1916.
- [SIK17] Trifce Sandev, Alexander Iomin, and Holger Kantz. Anomalous diffusion on a fractal mesh. *Phys. Rev. E*, 95:052107, May 2017.
- [WZYW19] Weijie Wang, Gaopeng Zhang, Lum-ing Yang, and Wei Wang. Research on garment pattern design based on fractal graphics. *Eurasip journal on image and video processing*, 2019(1):1–15, 2019.
- [ZSDL15] Liwei Zhang, Yee Soong, Robert Dillmore, and Christina Lopano. Numerical simulation of porosity and permeability evolution of mount simon sandstone under geological carbon sequestration conditions. *Chemical Geology*, 403:1–12, 2015.

The Usage of the BP-Layers Stereo Matching Algorithm with the EBCA Camera Set

Adam L. Kaczmarek

Gdansk University of Technology, Faculty of
Electronics, Telecommunications and
Informatics, ul. G. Narutowicza 11/12, 80-233
Gdansk, Poland
adakaczm@pg.edu.pl

ABSTRACT

This paper is concerned with applying a stereo matching algorithm called BP-Layers to a set of many cameras. BP Layers is designed for obtaining disparity maps from stereo cameras. The algorithm takes advantage of convolutional neural networks. This paper presents using this algorithm with a set called Equal Baseline Camera Array. This set consists of up to five cameras with one central camera and other ones around it. Such a set has similar advantages as a stereo camera. In particular this equipment is suitable for providing 3D vision for autonomous robots operating outdoors. The research presented in this paper shows the extent to which results of using BP Layers are improving because of using the EBCA set instead of a stereo camera.

Keywords

camera array; stereo matching; stereo camera; disparity map; 3D scanning;

1 INTRODUCTION

This paper contributes to the development of 3D vision technologies for autonomous robots. This kind of robots are able to interact with their surrounding and to perform tasks without being directly controlled by human operators. A crucial module of such a robot is its vision system because it makes it possible for the robot to locate in 3D space objects with which it is interacting.

This paper presents research on a 3D vision system based on EBCA (Equal Baseline Camera Array) [Kac15, Kac19]. It is a camera array which consists of up to 5 cameras. Images from such an array are processed as if they were taken by a set of stereo cameras. The description of this array is presented in Sect. 3. The main application of the array is using it for robotic fruit harvesting in which autonomous robots can locate and pick up fruits without being directly controlled by human operators. However EBCA can be also applied to other kinds of robots, in particular those operating underwater [KB21].

The novelty of the research presented in this paper lies in using the EBCA set with the BP Layers (Belief Propagation Layers) algorithm [KSS⁺20]. BP Layers was proposed by authors from Graz University of Technology located in Austria and Czech Technical University in Prague. It is the algorithm which takes advantage of convolutional neural network in order to make it possible to determine locations of objects in 3D space on the basis of images from a stereo camera. The description of this algorithm is presented in Sect. 2.3.

The presented research describes the method of adapting the BP Layers algorithm to EBCA. BP Layers were originally developed for stereo cameras. EBCA is a set with a greater number of cameras. Therefore, it is necessary to apply a method which would make it possible for the BP Layers algorithm to take advantage of all images from the EBCA set instead of only two images taken by a single stereo camera. The method selected for this purpose is EEMM (Exception Excluding Merging Method) which is described in [Kac17c]. Experiments show that the application of this method reduces on average 22.18% of errors occurring in the results of BP Layers obtained from a single stereo camera.

2 RELATED WORK

Stereo vision is one of technologies which makes it possible to locate objects in the 3D space and to determine their shape. There are many other methods designed for this purpose, however stereoscopy has

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

some unique features which makes it particularly useful in some circumstances. Methods alternative to stereo cameras include using structured light 3D scanners, Light Detection and Ranging (LIDAR) and methods based on multi-view stereo vision (MVS) [GIV10, RCG21, SCD⁺06]

2.1 Equipment for 3D scanning

Devices which are the main alternative to stereo cameras are Structured light 3D scanners [GIV10]. Structured light 3D scanners provide data regarding location and shape of object with a very high quality. The main feature of light scanners is such that they emit light patterns in order to perform a measurement. As a consequence it is problematic to use 3D light scanners when they are exposed to intensive sunlight. Sunlight interferes with procedures performed by a scanner causing that a scanner needs to be equipped with a high intensity light source in order to make it possible to use it in sunlight. However, even with such a significant light source it is problematic to use these devices when objects are not in a close vicinity of a scanner. These problems do not occur when stereo cameras are used. They can be used in daylight and for large distances.

Another equipment which can be used for locating objects and determining their shape are laser based devices using LIDAR [RCG21]. LIDAR is based on emitting rays of lasers in a set of different directions and recording distance from each point included in the measurement. LIDAR devices can operate outdoors however their disadvantage is such that they have much lower resolution than stereo cameras. It is related to a problem with precisely aiming a laser beam in appropriate direction. The resolution of cameras mainly depends on their sensor size and lens. It is easier and cheaper to achieve high resolution data with cameras than with devices using LIDAR.

A 3D shape of objects can be also determined with the use of the multi-view stereo (MVS) technology [SCD⁺06]. In this method it is required to make a set of images of an object for which a 3D scan is obtained. These images need to be taken by placing camera in different points of view located around the object. On the basis of these images the MVS algorithm determines locations of cameras at the moment of taking images and then the algorithm calculates the shape of objects by matching in images locations of the same parts of a real object. These parts need to be visible in many images included in the processed set. The problem with the MVS technology is such that it requires to take images from around the scanned object. It is not possible to perform such a procedure in every situation. In particular it is problematic when an autonomous robot has to locate some distant objects

like in case of an autonomous car operating on a street. Stereo cameras retrieve data about the distance to objects and their parts without the necessity to move the device around these objects.

2.2 Stereo cameras

Images from a stereo camera need to be processed in order to estimate distances similarly as in case of MVS. Retrieving distances to objects visible in images consists of two steps. In the first step, a stereo matching algorithm identifies in two images locations of the same parts of real objects. In order to achieve this cameras are distinguished between a reference camera and a side camera. Locations of cameras are different therefore relative locations in two images of the same objects is also different. It does not apply only to objects which are located so far away from a stereo camera that their location on both of images is the same. This difference in locations of corresponding parts is called disparity. A set of all found disparities for a reference image forms a disparity map [HI15, Kac19].

Values of disparities indicate distance to objects. The closer the object is to a stereo camera the greater will be a disparity. Taking into account parameters of a stereo camera such as a distance between its cameras called baseline and focal length of lens disparities can be converted to distances forming a depth map.

A stereo matching algorithm searches in a side camera for areas with the lowest matching cost for every point of a reference image. In general stereo matching algorithms perform a local matching which means that the search for a corresponding area is performed only in a part of a side image in which it is expected that a matching area is present. After performing local matching most of stereo matching algorithms perform global matching in which disparities are optimized globally. In this step disparities are modified with respect to values of disparities in their vicinities. One of the methods of global optimization of disparities is based on Markov Random Fields (MRF) [SS02, Bes86]. MRF is a method from which the BP Layers algorithm is derived. It is the algorithm which is used in experiments presented in this paper.

The problem with stereo matching is such that algorithms do not correctly match all points of a reference image with points of a side image. This causes errors in values present in disparity maps obtained as a result of matching. A large number of stereo matching algorithms have been developed in order to find the best methods. There are also rankings of this kind of algorithms. The most popular rankings of stereo matching algorithms are Middlebury Stereo Vision (<https://vision.middlebury.edu/stereo/eval3/>) and KITTI Vision Benchmark Suite (<http://www.cvlibs.net/datasets/>)

kitti/ [SSZ01, SS02, SHK⁺14, GLU12]. The first ranking consists of over 190 algorithms the latter lists over 300 ones.

2.3 BP Layers algorithm

The BP Layers algorithm improves previously developed BP algorithm with the technology of convolutional natural networks. BP was proposed by Tappen and Freeman in [TF03]. It is a method for globally optimizing disparities using the concept of MRF and Conditional Random Fields (CRFs) [LMP01]. In order to develop the algorithm and to perform experiments authors of BP Layers took advantage of many important stereo matching algorithms including Belief Propagation (BP) [TF03], tree-structured dynamic programming [BG08] and semi-global matching [Hir08].

BP layers was implemented with PyTorch using the CUDA architecture. In order to train the network authors of BP layers used data released by authors of rankings of stereo matching algorithms. Rankings provide testbeds and benchmarks. The main part of such a testbed are sets of images taken by a stereo camera and ground truth containing correct values of disparities which stereo matching algorithms should obtain after processing image pairs. These input images and correct values are crucial data for training neural networks. BP Layers was executed using Middlebury Stereo Vision and KITTI test data.

3 EQUAL BASELINE CAMERA ARRAY

Equal Baseline Camera Array was designed to address the problem with errors occurring on disparity maps obtained on the basis of images from a stereo camera [Kac15, Kac17a, Kac17b]. EBCA preserves all the benefits of a stereo camera however simultaneously EBCA provides higher quality of data than a stereo camera. Park and Inoue were the first ones who proposed using this kind of a camera set [PI98]. Information regarding other researchers working with this set can be found in [KB21].

EBCA is a set of cameras which consists of a central camera and up to four side cameras located around the central one. The baseline is the same in every this kind of a camera pair. All cameras in the set are aimed in the same direction. Cameras in EBCA are regarded as if they create a set of up to four stereo cameras such that each one of them consists of the central camera and one of side cameras. Therefore all these stereo cameras share the same camera which is a central one. This camera has a function of a reference camera in all of these considered camera pairs. They will be marked with S_i , $i \in 1, 2, 3, 4$. The maximum value of i depends on the number of side cameras included in the set. The



Figure 1: The real EBCA set used in experiments

real EBCA used in experiments is presented in Fig. 1 [Kac19]. It consists of MS LifeCam Studio cameras.

Using EBCA resembles making many measurements of the same distance using cameras S_i . These measurements are partly independent as cameras S_i share a central camera, but they consist of different side cameras. Disparity maps can be obtained on the basis of images from each camera S_i . Data acquired from S_i is then processed in order to obtain a single disparity map which contains less errors than any disparity map acquired from a single stereo camera S_i . The method of merging data from cameras S_i is a scientific problem itself because it required developing a method which reduces the amount of errors to the highest possible extent.

4 TESTBED

The author of papers [Kac17c, Kac19] proposed a few algorithms for merging this data. Mainly two kinds of methods were proposed. In the first one disparity maps are calculated on the basis of images from every camera S_i , then data included in these maps is compared and merged in order to acquire a disparity map of a higher quality. The merging method proposed by the author of [Kac17c] was called EEMM (Exception Excluding Merging Method). Method EEMM was used in research presented in this paper because experiments described in [Kac17c] showed that it is the most suitable for adapting stereo matching algorithms to the EBCA set without the necessity to modify the source code of a stereo matching algorithm. In the second type of a merging method the internal structure of a stereo matching algorithm selected for processing data from EBCA is modified in order to adapt it so that the algorithm can simultaneously process all images from EBCA. This method of modifying stereo matching algorithms makes it possible to use parts of stereo matching algorithms as if only two images were processed however other parts of algorithms process



Figure 2: Test images used in the experiments with BP layers applied to EBCA

data from all images simultaneously. This method of adapting stereo matching algorithms to EBCA is described in [Kac19].

A testbed presented in [KB21] was used to evaluate results of applying the BP layers algorithm to EBCA with the use of the EEMM method. The testbed was designed for testing stereo matching algorithms designed for EBCA. The testbed contains six sets of images taken by EBCA presented in Fig. 1. Every set consists of images made from all cameras included in EBCA. These were images of plants with at least one fruit. Plants used for making the testbed were strawberries, cherries and redcurrant. Cameras in EBCA were calibrated using methods described in [KB21]. Apart from images the testbed contains ground truth which provides correct values of disparities which should be acquired as a result of using stereo matching algorithms.

Three sets from the testbed were used for evaluating the BP layers algorithm applied to EBCA. These sets were marked with ST_1 , CH_1 , RC_1 . Figure 2 shows images used for preparing the sets.

Results were evaluated with the use of three quality metrics. These were percentage of bad matching pixels (BMP), percentage of bad matching pixels in background (BMB) and the coverage level (COV) [Kac19]. BMP is the most common quality metric used for evaluating results of stereo matching. It identifies the share of points whose values are within acceptable error margin [SS02].

Another metric used for evaluating results is called BMB [Kac19]. This metric is calculated in the similar way as BMP however only points in the background are taken into account. The background of a reference

image is an area for which there are no matching areas in a side image. A reference image may contain views of some objects located behind objects placed in the foreground. Objects in the background are only partly visible. For example such a background is the ground visible between plants leaves. Such areas are however not visible on side images on which other parts of ground can be seen between leaves of a plant located in the foreground. Therefore a stereo matching algorithm is not able to match areas of the background visible on the reference image with areas on a side image. In such a case a stereo matching algorithm should mark on the resulting disparity map that the disparity in this areas is unknown.

The third metric used for evaluating results is called COV [Kac19]. This metric shows the extent to which a resulting disparity map contains disparities with regard to the size of an area in which matching was performed.

5 EXPERIMENTS

The author of this paper performed experiments in order to evaluate results of adapting the BP Layers algorithm to EBCA with the use of the EEMM merging method. Experiments were performed on data sets ST_1 , CH_1 and RC_1 presented in Sect. 4. The implementation of BP Layers provided by its authors as used. Experiments presented in this paper were based on the neural network trained with the KITTI 2015 data set. Experiments were performed on a computer with the NVidia GeForce 1060 6GB graphic card. The EEMM method was used to merge data from cameras C_i included in EBCA. EEMM was executed with both of its parameters, i.e. Q and B equal to 5 [Kac17c].

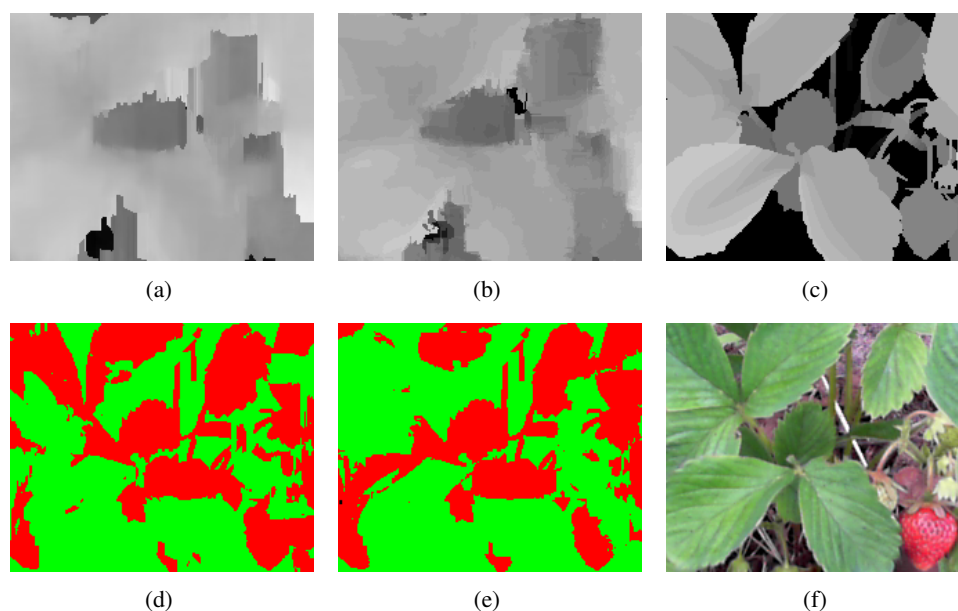


Figure 3: (a) Results of using BP layers with a single stereo camera, (b) Results of using BP layers with EBCA, (c) Ground truth, (d) error map for results presented in (a), (e) error map for results presented in (b), (f) 2D image

Figure 3 presents results obtained for the ST_1 set. Part (a) of this figure shows a disparity map obtained as a result of using original version of BP Layers with a single pair of cameras. Part (b) of Fig. 3 presents a disparity map acquired with the use of EEMM and EBCA with five cameras. These results can be compared with correct values of disparities available in ground truth visualized in part (c). Parts (d) and (e) are error maps generated for results presented in parts (a) and (b) of Fig. 3, respectively. Green color symbolizes areas in which matching was correct. Part (f) shows the image of the analyzed plant. It can be noticed that the disparity map presented in part (b) contains smaller areas with inappropriate values of disparities than the disparity map shown in (a).

These differences in the quality of results were also verified by calculating values of quality metrics described in Sect. 4. The purpose of experiments was to verify the influence of number of cameras included in EBCA on the quality of results. Figure 4 shows values of the BMP metric which was obtained for each data set used in experiments with regard to the number of cameras included in calculations. Figure 4 contains average values acquired as a result of using every possible subset of cameras from 5 camera EBCA show in Fig. 1. For example values for two cameras are average values of BMP calculated on the basis of four disparity maps acquired from stereo cameras S_1 , S_2 , S_3 and S_4 . Similarly, results for other number of cameras are average values based on all possible configurations

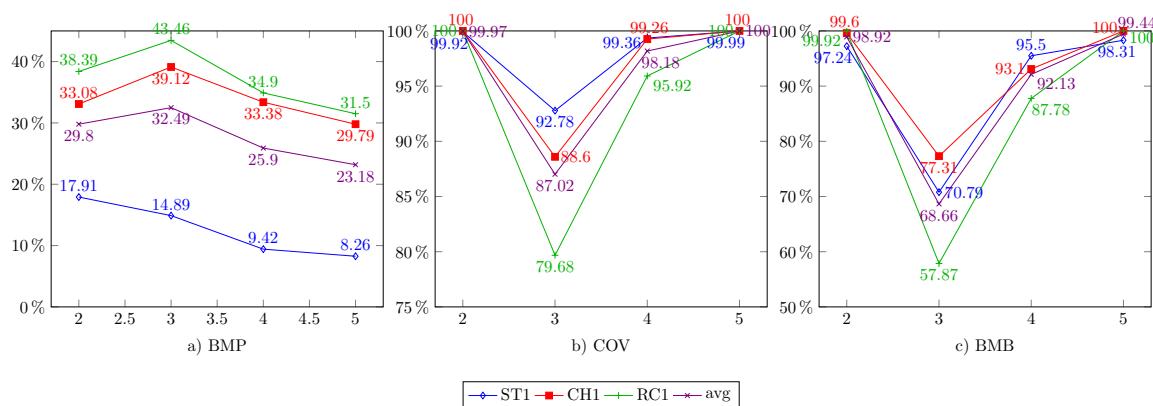


Figure 4: Values of quality metrics ((a) BMP, (b) COV, (c) BMB) obtained for different data sets (ST_1 , CH_1 and RC_1) and average values (AVG)

of using S_1 , S_2 , S_3 and S_4 within the limit of available cameras.

Results show that for every tested data set values of BMP was lower when 5 cameras were used instead of 2 ones. This is advantageous as lower BMP implies that fewer number of errors occurred in a disparity map. Results also showed that increasing the number of cameras does not always leads to lower values of BMP. In particular for data sets CH_1 and RC_1 adding a third camera to a two camera set caused a decrease of the disparity map quality estimated by BMP. It is mainly caused by the features of test data and the BP-Layers algorithm. The algorithm provides relatively large areas with same values of disparities even in case of areas for which it is impossible to obtain a disparity. This applies to all 4 considered stereo cameras from the EBCA set. In previous experiments presented in [Kac17c] it was already noticed that adding a third camera to a stereo camera does not lead to high improvement of results. However, if EBCA consists of at least four cameras the improvement is significant.

Experiments presented in this paper showed a similar relation. Increasing the number of cameras to 4 always causes that results are better than in case of using two cameras. Figure 4 also show average results calculated for data sets ST_1 , CH_1 and RC_1 . This chart show that on average EBCA with 5 cameras did not contain 22.18% of errors which were present in the results of using a single stereo camera.

Experiments also showed that the coverage level of disparity maps generated by BP Layers is over 99.97% as presented in 4(b). This means that the algorithm does not set values of disparity to unknown even if input images do not make it possible to appropriately acquire disparities. As a consequence the BMB metric presented in 4(c) is equal to over 98.92% for results obtained on a single stereo camera used with BP Layers as the algorithm provides incorrect values for all the background area. Number of errors in the background was reduced for all data sets when EEMM was used with three cameras. Instead of including in disparity maps incorrect values of disparities EEMM executed with three cameras caused that disparities in some areas were set to values indicating that disparities are unknown. Because of that the COV metric also became lower. Increasing number of cameras to four and five caused that values of BMB and the coverage level became almost as high as in case of using two cameras.

Experiments were also performed in order to verify whether there are stereo cameras in the EBCA set which produce better results than other cameras. Table 1 shows values of BMP obtained for different data sets with regard to the stereo camera which was used. Results show that for every data set using a different stereo camera led to obtaining the best results in terms

Table 1: Values of BMP for different stereo cameras included in EBCA

Camera	ST_1	CH_1	RC_1
S_1 (right)	16.15%	21.75%	28.34%
S_2 (up)	7.81%	37.22%	56.95%
S_3 (left)	11.11%	25.55%	20.81%
S_4 (down)	36.58%	47.81%	47.47%

of BMP. Camera S_2 which is the one consisting of an upper side camera generated both the best results in case of the ST_1 data set and the worst results for RC_1 . It can be noticed that the S_4 camera had the worst performance. A possible cause is such that lower side camera included in S_4 is differently illuminated than other side cameras. The source of light which is the sun in open field is always above EBCA. Therefore, it might influence the measurement. However, this observation requires more investigation. Moreover, as presented in [Kac19] removing lower side camera can also be advantageous in case of mounting EBCA on a robotic arm.

6 SUMMARY

The greatest benefit of using BP Layers with EBCA is such that on average 22.18% of errors in disparities measured with BMP are eliminated when five cameras were used instead of two ones. The BP Layers algorithm is not the one which is at the top of KITTI or Middlebury rankings. However, BP Layers has a very important features which makes this algorithm particularly important. First, it is an algorithm for which authors provided a source code. Thus, it is possible to verify its quality and results unlike most of other algorithms included in rankings of stereo matching algorithms. Second important feature is such that it is an algorithm based on CNN which has a very high speed as for such a kind of algorithm. Many stereo matching algorithms which takes advantage of CNN requires a large amount of computing and relatively a lot of time to generate results. In real application such as using the algorithm with an autonomous robot is it crucial to obtain results in real time. In KITTI Vision Benchmark presents execution times of algorithm. BP Layers runs KITTI test data in 0.39 s.

The disadvantage of using EBCA lies in the necessity to use a greater number of cameras than in case of a stereo camera. Another limitation of the research presented in this paper is such that BP Layers is an algorithm which requires using convolutional neural network. Therefore, in case of applying EBCA with this algorithm to a real autonomous robot it is necessary to equip the robot with a computer that have sufficient computational power and appropriate graphic card required by BP Layers. Using small single-board computers will not be enough. Another issue related to

using EBCA in real environment is such that it was not tested under harsh weather conditions such as heavy rain.

Applying BP Layers to EBCA with the use of the EEMM method causes the increase in the execution time of matching. When 5 cameras are used BP Layers needs to be executed four times because it needs to process four pairs of images from stereo cameras S_i included in EBCA. As far as computational complexity is concerned it does not cause any change in computational complexity even though it increases the execution time four times. The merging phase consists of comparing values for every point of four disparity maps when four of them are being merged. Therefore the computational complexity of this process is linear with regard to the number of points in the image. This complexity is not higher than a complexity of stereo matching algorithm because such an algorithm also needs to process every point of input images. Therefore, computational complexity of the EBCA approach and using a single stereo camera is the same. Nevertheless processing image pairs from different stereo cameras included in EBCA can be performed independently from each other. Each pair can be processed on a different processor or different core of a processor. Making calculations parallel will cause that results from EBCA will be obtained nearly in the same time as results based on only a pair of images. The increase in time will only be caused by the necessity to run the EEMM merging method which does not require significant computation power. Taking into account that the quality of this results will be higher it is an acceptable cost.

7 REFERENCES

- [Bes86] Julian Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986.
- [BG08] Michael Bleyer and Margrit Gelautz. Simple but effective tree structures for dynamic programming-based stereo matching. pages 415–422, 01 2008.
- [GIV10] Andreas Georgopoulos, Charalabos Ioannidis, and Artemis Valanis. Assessing the performance of a structured light scanner. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 5):251–255, 2010.
- [GLU12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, June 2012.
- [HI15] Rostam Hamzah and Haidi Ibrahim. Literature survey on stereo vision disparity map algorithms. *Journal of Sensors*, 2016:1–23, 12 2015.
- [Hir08] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341, Feb. 2008.
- [Kac15] Adam L. Kaczmarek. Improving depth maps of plants by using a set of five cameras. *Journal of Electronic Imaging*, 24(2):023018, 2015.
- [Kac17a] Adam L. Kaczmarek. Influence of aggregating window size on disparity maps obtained from equal baseline multiple camera set (ebmcs). In Ryszard S. Choraś, editor, *Image Processing and Communications Challenges 8, IP&C 2016, Advances in Intelligent Systems and Computing*, pages 187–194, Cham, 2017. Springer International Publishing.
- [Kac17b] Adam L. Kaczmarek. Stereo camera upgraded to equal baseline multiple camera set (ebmcs). In *2017 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4, June 2017.
- [Kac17c] Adam L. Kaczmarek. Stereo vision with equal baseline multiple camera set (ebmcs) for obtaining depth maps of plants. *Computers and Electronics in Agriculture*, 135:23 – 37, 2017.
- [Kac19] Adam L. Kaczmarek. 3d vision system for a robotic arm based on equal baseline camera array. *Journal of Intelligent & Robotic Systems*, Dec 2019.
- [KB21] Adam L. Kaczmarek and Bernhard Blaschitz. Equal baseline camera array-calibration, testbed and applications. *Applied Sciences*, 11(18), 2021.
- [KSS⁺20] Patrick Knöbelreiter, Christian Sormann, Alexander Shekhovtsov, Friedrich Fraundorfer, and Thomas Pock. Belief propagation reloaded: Learning bp-layers for labeling problems. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [LMP01] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*,

- ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [PI98] Jong-Il Park and Seiki Inoue. Acquisition of sharp depth map from multiple cameras. *Signal Processing: Image Communication*, 14(1-2):7 – 19, 1998.
- [RCG21] Ricardo Roriz, Jorge Cabral, and Tiago Gomes. Automotive lidar technology: A survey. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–16, 2021.
- [SCD⁺06] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528, June 2006.
- [SHK⁺14] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. *High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth*, pages 31–42. Springer International Publishing, Cham, 2014.
- [SS02] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002. Microsoft Research Technical Report MSR-TR-2001-81, November 2001.
- [SSZ01] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, pages 131–140, Dec. 2001.
- [TF03] M.F. Tappen and W.T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 900–906 vol.2, Oct. 2003.

Detail preserving non-rigid shape correspondences

Manika Bindal
Goa University
Taleigao, Goa
India 403206
manika.bindal@gmail.com

Venkatesh Kamat
IIT Goa
Farmagudi, Goa
India 403401
vvkamat@iitgoa.ac.in

ABSTRACT

Understanding shapes is an organic process for us (humans) as this is fundamental to our interaction with the surrounding world. However, it is daunting for the machines. Any shape analysis task, particularly non-rigid shape correspondence is challenging due to the ever-increasing resolution of datasets available. Shape Correspondence refers to finding a mapping among various shape elements. The functional map framework deals with this problem efficiently by not processing the shapes directly but rather specifying an additional structure on each shape and then performing analysis in the spectral domain of the shapes. To determine the domain, the Laplace-Beltrami operator has been utilized generally due to its capability of capturing the global geometry of the shape. However, it tends to smoothen out high-frequency features of shape, which results in failure to capture fine details and sharp features of shape for the analysis. To capture such high-frequency sharp features of the shape, this work proposes to utilize a Hamiltonian operator with gaussian curvature as an intrinsic potential function to identify the domain. Computationally it is defined at no additional cost, keeps global structural information of the shape intact and preserves sharp details of the shape in order to compute a better point-to-point correspondence map between shapes.

Keywords

shape matching, shape correspondence, functional maps

1 INTRODUCTION

Shapes in computational context refers to digital representation of any real world object such as humans, chairs, etc. These digital representations can be meshes, point clouds or voxel grids. With ever increasing technological advancements, the accuracy with which these digital representations are being captured has transformed the field of shape analysis. Particularly, shape matching is quite an interesting area enticing researchers across multiple domains from Computer Graphics, Image processing, Geometry Processing and Computer Vision. A sub-area focusses on the fundamental task of computing shape correspondences, where rather than just specifying if two shapes match, a mapping is also desired between various elements of given shapes. Major applications constitute object reconstruction, attribute transfer, Statistical modelling, Shape Interpolation and morphing, etc.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.



Figure 1: Reconstructed hand from human mesh via (left) 150 Laplace-Beltrami eigenfunctions (right); 150 Hamiltonian eigenfunctions

Based on how shapes can deform, varied approaches have been suggested [VKZHC011]. Rigid shapes undergo transformations that preserve extrinsic features i.e. euclidean distances remain intact while non-rigid shapes deform anyhow [BBK07]. Rigid deformation tends to transform the shape without changing its geometry or topology via rotation or translation. Non-rigid involves changing in geometry as well as topology via stretching or bending. Finding correspondences for rigid shapes has plethora of efficient solutions. However, due to the vast space of deformations for non-rigid shapes, it is an interesting area to work. Another con-

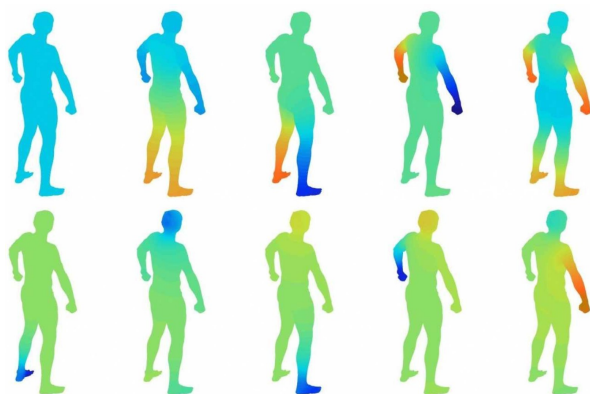


Figure 2: Laplace-Beltrami Eigenfunctions (top row); Hamiltonian Eigenfunctions (bottom row) arranged according to ascending order of eigenvalues respectively

sideration would be to establish a mapping either between all the elements of the shape or a partial subset of the elements of shape leading to determining full and partial correspondences respectively [RCB⁺17]. The field's vastness and complexity are evident from these aspects.

In this work, the analysis is restricted to Isometric deformations of triangulated meshes where distances along the surface are invariant, when deformed without any stretching or tearing of the surface. The proposed research aims to investigate whether utilizing the sharp features of two given triangle meshes can improve the point-to-point correspondence mapping between the respective shape by using a functional framework.

2 RELATED WORK

Shape matching can be broadly categorized into two approaches: geometric transformation-based and descriptor-based. Geometric transformation-based approaches involve finding the geometric transformation that aligns two shapes and then establishing correspondences based on proximity [GMGP05]. This approach typically involves minimizing the distance between corresponding points or features in the two shapes. Descriptor-based approaches, on the other hand, do not require the shapes to be aligned. Instead, correspondences are established based on similarities between shape descriptors or feature vectors [CCFM08]. This approach is often used when the shapes have different topologies or geometry. A hybrid approach can also be used, where the alignment and computation of correspondences are alternated and iteratively improved [Ale02]. This approach can be effective in cases where the shapes undergo non-rigid deformations, where the geometric transformation-based approach may not be suitable.

For rigid shapes, *Iterative Closest Point*, ICP algorithm [BM92] is the celebrated technique that iteratively finds

shape correspondence by aligning the shapes first by finding the optimal geometric transformation consisting of rotations and translations, then by utilizing Nearest Neighbour approach to find the closest points for the computed alignment [RL01]. Other methods for rigid shapes are surveyed in [BSBW14].

For non-rigid shapes, to allow any kind of mathematical analysis is to restrict the deformation to an isometry, wherein the distances between pair of points are preserved along the surface, while the shape is undergoing isometric deformation. Computing a map for high quality meshes with large number of vertices is computationally quite intensive. To cater different metric spaces, [EK03] introduced the idea of isometrically embedding the shapes into a canonical domain to allow any kind of shape analysis task. Shapes can also be embedded spectrally [ZVKD10] by utilizing eigenmodes of linear operators defined on the shape.

The current work is inspired by functional map framework [OBCS⁺12] developed to efficiently compute a mapping from the function space of one shape to another and subsequently determining point-to-point correspondences between them. Shape features are projected onto functional basis to reduce computation time during analysis and capture geometry along with other properties of shape effectively. Further, a map is computed between shapes by setting up an optimization problem, which then is refined to obtain point-to-point correspondences. Refer section 3 for more details on Functional Maps.

Identification of good basis is crucial for the functional map framework to output point-to-point correspondences. [OBCS⁺12] proposed to use Laplace-Beltrami eigenfunctions as functional basis. To capture high-frequency information on the shape, [NVT⁺14] proposed *compressed manifold modes* that are sparse basis with local support upto sign flip and ordering. By explicitly controlling the region of localization, [MRCB18] introduced *localized manifold harmonics* (LMH). These properties quickly become challenging to adopt when dealing with multiple meshes together as in case of shape correspondence, pose transfer, etc. Though [KBB⁺13] and [EKB⁺15] take into account multiple meshes and obtained basis incorporating the geometric information of all the meshes, it still fails to capture high-frequency information of the shapes.

3 BACKGROUND

Functional map is a promising framework for non-rigid shape matching that finds a map between the two function spaces defined on shapes rather than a map between shapes directly. Consider two shapes \mathcal{M} and \mathcal{N} which are represented as triangular meshes. A point-to-point map between \mathcal{M} and \mathcal{N} is given as $\mathcal{T}: \mathcal{M} \rightarrow \mathcal{N}$, where for any point $p \in \mathcal{M} \implies \mathcal{T}(p) \in \mathcal{N}$. If the

number of points are same on both the shapes, then a bijection is desired where \mathcal{T}^{-1} exists.

3.1 Functional Maps

Functional map framework works by defining spaces of scalar valued functions $\mathcal{F}(\mathcal{M}, \mathbb{R})$ and $\mathcal{F}(\mathcal{N}, \mathbb{R})$ on shapes \mathcal{M} and \mathcal{N} respectively. It aims at computing a linear mapping between these function spaces $\mathcal{T}_F: \mathcal{F}(\mathcal{M}, \mathbb{R}) \rightarrow \mathcal{F}(\mathcal{N}, \mathbb{R})$. Map \mathcal{T}_F which associate values of $f: \mathcal{M} \rightarrow \mathbb{R}$ and $g: \mathcal{N} \rightarrow \mathbb{R}$ can be represented as a matrix $\mathbb{C} \in \mathbb{R}^{k_1 \times k_2}$ with $\phi_{\mathcal{M}}$ and $\phi_{\mathcal{N}}$ are respective basis such that $|\phi_{\mathcal{M}}| = k_1$ and $|\phi_{\mathcal{N}}| = k_2$. The framework pipeline consists of following steps:

- 1 For each shape compute invariant feature descriptors say **F** and **G** with respect to isometric deformation
- 2 Choose basis $\phi_{\mathcal{M}}$ and $\phi_{\mathcal{N}}$ for both the shapes
- 3 Create function preservation constraints by projecting feature descriptors **F** and **G**, computed in step 1 onto respective basis as **A** and **B**
- 3 Set up other constraints like operator commutativity or regularization constraint
- 4 Compute optimal functional map **C** by minimizing the following energy:

$$E(\mathbf{C}) = \|\mathbf{CA} - \mathbf{B}\|^2 + \|\mathcal{S}_F^{\mathcal{N}} \mathbf{C} - \mathbf{C} \mathcal{S}_F^{\mathcal{M}}\|^2$$

- 5 Refine **C** further and compute point-to-point map by using ICP like algorithm

Note that $\mathcal{S}_F^{\mathcal{M}}$ and $\mathcal{S}_F^{\mathcal{N}}$ are operators mentioned in Step 3. \mathcal{T}_F acts linearly between function spaces and is sufficient to compute \mathcal{T} . Idea is to add a structure on the shape and work on that rather than directly on the shapes. Functional maps, due to its efficiency in dealing with high-resolution shapes by reducing the dimension where shape analysis is done, works well particularly for shape matching.

3.2 Laplace-Beltrami Operator (LBO)

The self-adjoint Laplace-Beltrami Operator on manifold \mathcal{M} is specified as $\Delta_{\mathcal{M}}: \mathcal{F}(\mathcal{M}, \mathbb{R}) \rightarrow \mathcal{F}(\mathcal{M}, \mathbb{R})$ which via spectral theorem, admits an eigendecomposition with non-negative eigenvalues λ and orthonormal eigenfunctions ϕ popularly known as *manifold harmonics* $\Delta_{\mathcal{M}} \phi = \lambda \phi$ [VL08]. For mesh \mathcal{M} with n vertices, a popular discrete [MDSB03] cotangent Laplace-Beltrami Operator matrix $\mathbf{L}_{\mathcal{M}} \in \mathbb{R}^{n \times n}$ is defined in terms of a sparse matrix $\mathbf{W}_{\mathcal{M}} \in \mathbb{R}^{n \times n}$ containing cotangent weights and a lumped mass matrix $\mathbf{A}_{\mathcal{M}} \in \mathbb{R}^{n \times n}$ containing vertex areas as

$\mathbf{L}_{\mathcal{M}} = \mathbf{A}_{\mathcal{M}}^{-1} \mathbf{W}_{\mathcal{M}}$. The eigendecomposition (refer Fig. 2) of such a Laplace-Beltrami operator can be posed as an optimization problem:

$$\min_{\Phi} \text{tr}(\Phi^T \mathbf{W}_{\mathcal{M}} \Phi) \quad \text{s.t.} \quad \Phi^T \mathbf{A}_{\mathcal{M}} \Phi = \mathbf{I} \quad (1)$$

where $\Phi \in \mathbb{R}^{n \times n}$ is the eigenvector matrix as $\Phi = (\phi_1, \phi_2, \dots, \phi_n)$ with eigenfunction ϕ_i arranged as columns according to increasing eigenvalues. Equation (1) is also equivalent to the generalized eigenvalue problem

$$\mathbf{W}_{\mathcal{M}} \Phi = \mathbf{A}_{\mathcal{M}} \Phi \Lambda$$

where $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ is the respective diagonal eigenvalue matrix. Refer [LZ10] for extensive details on spectral analysis via Laplace-Beltrami eigenfunctions.

3.3 Hamiltonian Operator

Hamiltonian operator H is a classical operator from quantum mechanics, that appears in Schrodinger's equation describing the wave motion of a particle. On a manifold mesh \mathcal{M} , Hamiltonian operator is described as the extension of Laplace-Beltrami operator $\mathbf{L}_{\mathcal{M}}$:

$$\mathcal{H}_{\mathcal{M}}(f) = \mathbf{L}_{\mathcal{M}}(f) + \mu \mathbf{V}_{\mathcal{M}}(f)$$

with parameter $\mu \in \mathbb{R}$ and $\mathbf{V}_{\mathcal{M}}: \mathcal{M} \rightarrow \mathbb{R}_+$ a potential function on \mathcal{M} [CSBK18]. Since Hamiltonian operator is the sum of two self-adjoint operators, it is also self-adjoint and hence, admits an eigendecomposition with real eigenvalues ζ and orthonormal eigenfunction ψ as $H\psi = \zeta\psi$ where ζ denotes particle energy at stationary eigenstate ψ . Refer Fig. 2 for first few Hamiltonian Eigenfunctions on the shape sorted according to increasing eigenvalues. Note that $\psi(x)$ i.e. eigenstate at point x on manifold represents the wave function of a particle where $|\psi(x)|^2$ specifies the probability of finding the particle at x . The generalized eigenvalue problem for Hamiltonian operator is specified as

$$(\mathbf{W}_{\mathcal{M}} + \mu \mathbf{A}_{\mathcal{M}} \text{diag}(v)) \Psi = \mathbf{A}_{\mathcal{M}} \Psi \Theta$$

, where $\Psi \in \mathbb{R}^{n \times n}$ is the orthonormal eigenvector matrix as $\Psi = (\psi_1, \psi_2, \dots, \psi_n)$ with eigenfunctions ψ_i arranged as columns, v is an n -dimensional potential vector and $\Theta = (\zeta_1, \zeta_2, \dots, \zeta_n)$ is the respective diagonal eigenvalue matrix. Parameter μ controls the trade-off between global and local support of eigenbasis. [CSBK18] introduced Hamiltonian operator to shape analysis domain.



Figure 3: Gaussian Curvature visualized on wolf shape

4 MOTIVATION

Since functional map is a flexible framework, there is an opportunity to make improvements at any step of the discussed pipeline (ref. sec.3.1). Though various attempts have been made to update functional maps by modifying it at various stages, basis selection remains a crucial step in the framework since it characterizes the domain in which the analysis is going to take place. Moreover, the basis should reduce representation complexity, be stable and compact. Laplace-Beltrami Eigenfunctions [Lev06] have mainly been utilized as basis to compute the desired mapping due to multi-scale property and invariance to isometric deformations of shapes. Though Laplacian eigenfunctions are compact and stable, it tends to smoothen out sharp features on the shape, which hampers the analysis. Also global nature of these eigenfunctions make it sensitive to topological changes. However, when dealing with challenging datasets, information regarding localized and detailed features of shape become significant, which Laplace-Beltrami eigenfunctions fail to capture. This work is motivated by proposing a basis that better captures the shape geometry by picking up sharp features of the shape and be computationally viable.

5 CONTRIBUTION

For Hamiltonian operator defined on shapes, the potential function is responsible for localizing the region so that Hamiltonian eigenfunctions capture high frequency information along with preserving the properties captured by Laplace-Beltrami eigenfunctions. From computing perspective, since discrete potential function is described as a diagonal matrix it amounts to no additional computation for basis over Laplace-Beltrami basis (ref sec. 3.3).

Since Gaussian curvature (K) fully characterizes the geometry of the shape [Ale02] and is given as the product

of the principal curvatures $K = k_1 * k_2$, it picks up the regions with negative and positive curvatures i.e. regions where sharp features of the shape appear. Refer Fig.3 for visualizing gaussian curvature on the shape where positive curvature regions are marked with yellow such as paws, while negative curvature regions are marked with red such as inside of the ear. Hence justifies the selection as a potential function.

In this work, the contribution is to suggest to use gaussian curvature as potential function to determine Hamiltonian basis, as it better captures the shape geometry which leads to better point-to-point correspondences, without any additional cost. Refer fig. 1 where shape signal was first projected onto each Laplace-Beltrami and Hamiltonian basis respectively and subsequently reconstructed via 150 of each set of bases. Note that Hamiltonian basis capture sharp features where fingers are also identified, while Laplace-Beltrami basis has smoothen out these details via utilizing same number of respective basis.

6 IMPLEMENTATION

6.1 Dataset

TOSCA dataset [BBK08] consisting of hi-resolution non-rigid shapes in a variety of poses have been utilized. The database contains a total of 80 objects, including 11 cats, 9 dogs, 3 wolves, 8 horses, 6 centaurs, 4 gorillas, 12 female figures, and two different male figures, containing 7 and 20 poses. Each object is a triangulated mesh with vertices, edges and triangular faces. Ground truth vertex-to-vertex correspondences are also provided, which are utilized to evaluate the performance of proposed technique.

6.2 Methodology

In this work, wolf meshes are considered to illustrate implementation details - wolf0 and wolf1. Functional framework to compute vertex-to-vertex correspondences between two triangular meshes is utilized, each mesh depicting a different pose of wolf shape from TOSCA dataset. Refer section 3.1 for the steps involved to determine correspondences via functional map.

First step is to compute feature descriptors for both the meshes, hence heat kernel signatures (HKS) [SOG09] and wave kernel signatures (WKS) [ASC11] are utilized which are defined via Laplace-Beltrami eigenvalues. Next step is to identify the bases for the shapes. For this Hamiltonian operator is selected as via potential function, bases can be enhanced to incorporate better shape geometry. Various potential functions with intrinsic features like gaussian curvature, gaussian curvature with absolute values and with extrinsic features like mean curvature and others were tried out with varied values of parameter μ (ref. Sect.3.3). Finally μ is

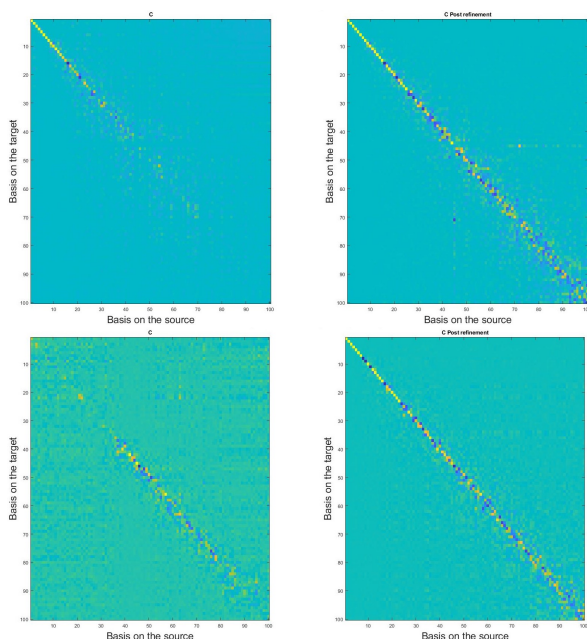


Figure 4: Functional map (pre and post refinement) via Laplace-Beltrami basis (top row); via Hamiltonian basis (bottom row) for wolf Mesh

takes as 5000 empirically and gaussian curvature was selected as the potential function to determine Hamiltonian operator. First 100 Hamiltonian eigenfunctions were considered as basis for the function spaces of shapes, arranged according to increasing eigenvalues for the next step.

Then function descriptors computed earlier were projected onto these chosen basis, so as to reduce the complexity of further shape processing. Along with Laplacian commutativity constraint the energy functional as discussed in section 3.1 is minimised via gradient-descent approach to get an optimal functional map. Procedure depicted in Section 6.2 of [OBCS⁺12] is utilized to refine the obtained functional map and also compute point-to-point correspondences simultaneously.

Refer Fig. 4 top row, that specifies functional map with 100 basis functions; previous to and post the refinement step via Laplace-Beltrami basis and Fig 4 bottom row for Hamiltonian basis. Note that Laplace-Beltrami basis picks up the low frequency features while Hamiltonian picks up high frequency features in the initial optimal map before refining the map.

To verify if the utilized approach performs better, accuracy of obtained point-to-point correspondence map needs to be established. For that, geodesic error is computed by summing up all geodesic distances from computed mapping of points to ground-truth mapping. For a vertex p in source mesh, let the obtained corresponding vertex in target mesh is q and the ground-truth establishes vertex r in target mesh as the corresponding

vertex for vertex p from source mesh, then the geodesic error at vertex p is the geodesic distance between vertices q and r on target mesh. Summing up geodesic error for each vertex on source mesh is represented as geodesic error for the obtained mapping.

Obtained results are compared with the existing approaches of Laplace-Beltrami basis [OBCS⁺12] and compressed manifold modes [NVT⁺14] as it claims to pick sharp details from the shapes.

With allowed normalized geodesic error threshold of 0.1, results are provided in Table 1 to compare results for wolf and Human meshes in different poses from TOSCA dataset. It shows that proposed basis performs better empirically over Laplace-Beltrami basis and wins over from compressed manifold modes by a very good margin. Refer Fig. 5 for obtained accuracy of point-to-point correspondences in terms of percentage of correct correspondences computed with respect to total number of vertices (or points) on the shapes, via Laplace-Beltrami basis, proposed Hamiltonian basis and via compressed manifold modes. Note that proposed basis give similar *exact error* i.e. percentage of point-to-point correspondence map with zero error to that of Laplace-Beltrami basis. However, with minimal error allowed proposed basis perform much better than other two. To enhance the empirical validity of our proposed work, accuracy of point-to-point correspondences for Human meshes from the same dataset are presented in the table, which justifies the use of proposed basis over existing ones.

Basis Type	Accuracy with less than 0.1 geodesic error	
	Wolf	Human
LBO Eigenfunctions	84.7%	64.09%
CMM	14.73%	12.6%
Proposed basis	98.6%	72.23%

Table 1: Different basis were utilized to determine functional space to compute point-to-point correspondences

For visualization purpose refer Fig.7 for wolf meshes, to see geodesic error plot as heat map on the shape itself in case of proposed and Laplace-Beltrami basis respectively. In case of Hamiltonian basis, at the very end of tail the error is high and rest of the shape has minimal error. However, for Laplace-Beltrami basis the error is scattered over the shape and particularly present at all regions with sharp features like paws, ears, etc. Compressed manifold modes performed quite poorly, hence excluded from this visualization. Similar to the wolf meshes, human meshes (refer Fig 8) also show similar result, wherein the geodesic error induced via Laplace-Beltrami bases is scattered while for Hamiltonian bases similarly show the localized error as depicted in wolf case.

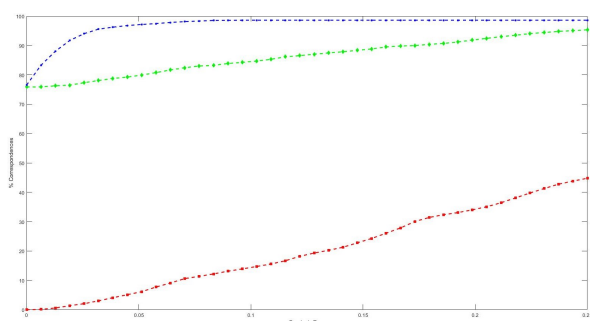


Figure 5: Accuracy of point-to-point correspondences; via Laplace-Beltrami basis (green line-diamond markers); via Hamiltonian basis (blue line-circular markers); via compressed manifold modes (red line-square markers) in the functional map framework for wolf mesh

Application

Texture mapping is considered as application for the computed correspondences via proposed basis. Refer Fig.6 for texture is transferred from wolf0 shape to wolf1, where particularly lower body of wolf is shown to highlight problematic areas.

7 DISCUSSION

In this work, the effect of selecting an intrinsic potential function, particularly Gaussian curvature, to describe Hamiltonian operator with respect to functional map framework has been studied. Due to modulations in manifold harmonics via an intrinsic potential function, Hamiltonian basis picked sharper features as compared to Laplace-Beltrami basis. These sharp features helped improve overall accuracy of point-to-point correspondences computed via functional framework considering Hamiltonian eigenfunctions as basis. Compressed manifold modes are localized basis which also aims at picking up the high frequency details, but fails to work for non-rigid shape matching because of the induced sparseness along with absence of global information determining the shape geometry. Proposed basis overcomes both these issues by being able to capture global information along with sharper features of the shape, hence fared well.

Based on empirical evidence gathered from testing multiple shape pairs in our study, it has been observed that introducing sharp feature information into the basis leads to an improved representation of shape geometry. This, in turn, enhances the ability of the basis to capture the geometric features of the shape. However, error localization is a phenomenon that is seen across multiple meshes, which might be due to Gaussian curvature picking up high curvature regions while not picking up no curvature zones or flat regions of the shape.

8 FUTURE SCOPE

The potential function proposed to compute Hamiltonian eigenfunctions is specified as an intrinsic feature of

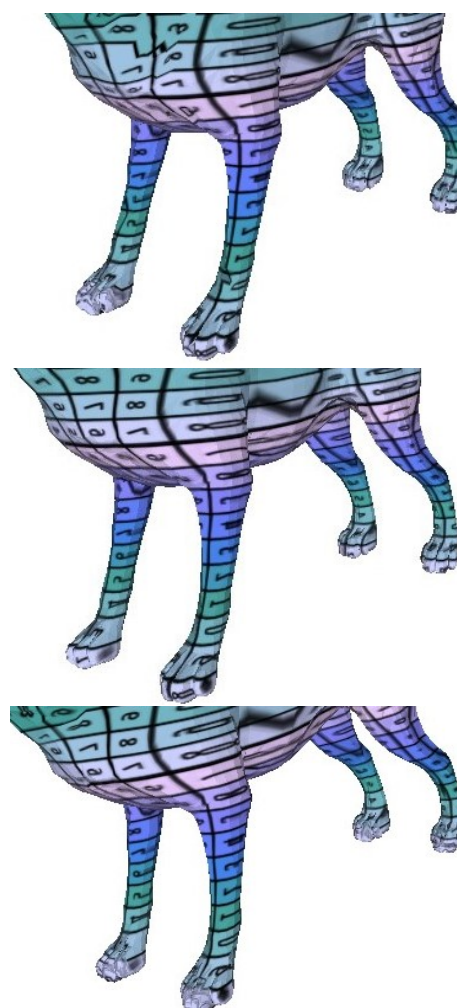


Figure 6: Texture mapping specified from wolf0 to wolf1 shape via computed correspondences; via Hamiltonian basis (top); via ground-truth (middle); via Laplace-Beltrami basis (bottom)

the shape which is fixed for each shape before the analysis initiates, be it some intrinsic invariant or any step function specifying a particular region on the shape. Along with its fixed nature, it shows error localization possibly due to Gaussian curvature not picking up no curvature zones. An interesting future prospect would be to compute a potential function that adapts itself such that it automatically picks those regions that are critical to both shapes considered together and in-turn enhances point-to-point correspondences too.

A potential avenue for future research is to investigate shape correspondences in non-rigid shapes that permit stretching and other deformations beyond isometric ones. This would present a greater challenge in the analysis, thereby providing an opportunity to explore new methods and approaches in this area.



Figure 7: Geodesic Error Plot visualized as heat map on the shape itself; via Hamiltonian basis (top); via Laplace-Beltrami basis (bottom) for wolf mesh

9 ACKNOWLEDGEMENTS

This work was financially supported by Visvesvaraya PhD Scheme, MeitY, Government of India under Grant MEITY-PHD-1090.

10 REFERENCES

- [Ale02] Marc Alexa. Recent advances in mesh morphing. In *Computer graphics forum*, volume 21, pages 173–198. Wiley Online Library, 2002.
- [ASC11] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1626–1633. IEEE, 2011.
- [BBK07] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Rock, paper, and scissors: extrinsic vs. intrinsic similarity of non-rigid shapes. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–6. IEEE, 2007.
- [BBK08] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.
- [BM92] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.
- [BSBW14] Ben Bellekens, Vincent Spruyt, Rafael Berkvens, and Maarten Weyn. A survey of rigid 3d pointcloud registration algorithms. In *AMBIENT 2014: the Fourth International Conference on Ambient Computing, Applications, Services and Technologies, August 24-28, 2014, Rome, Italy*, pages 8–13, 2014.
- [CCFM08] Umberto Castellani, Marco Cristani, Simone Fantoni, and Vittorio Murino. Sparse points matching by combining 3d mesh saliency with statistical descriptors. In *Computer Graphics Forum*, volume 27, pages 643–652. Wiley Online Library, 2008.
- [CSBK18] Yoni Choukroun, Alon Shtern, Alex M Bronstein, and Ron Kimmel. Hamiltonian operator for spectral shape analysis. *IEEE transactions on visualization and computer graphics*, 2018.
- [EK03] Asi Elad and Ron Kimmel. On bending invariant signatures for surfaces. *IEEE Transactions on pattern analysis and machine intelligence*, 25(10):1285–1295, 2003.
- [EKB⁺15] Davide Eynard, Artiom Kovnatsky, Michael M Bronstein, Klaus Glashoff, and Alexander M Bronstein. Multimodal manifold analysis by simultaneous diagonalization of laplacians. *IEEE transactions on pattern analysis and machine intelligence*, 37(12):2505–2517, 2015.
- [GMGP05] Natasha Gelfand, Niloy J Mitra, Leonidas J Guibas, and Helmut Pottmann. Robust global registration. In *Symposium on geometry processing*, volume 2, page 5. Vienna, Austria, 2005.
- [KBB⁺13] Artiom Kovnatsky, Michael M Bronstein, Alexander M Bronstein, Klaus Glashoff, and Ron Kimmel. Coupled quasi-harmonic bases. In *Computer Graphics Forum*, volume 32, pages 439–448. Wiley Online Library, 2013.
- [Lev06] Bruno Levy. Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 13–13. IEEE, 2006.
- [LZ10] Bruno Levy and Richard Hao Zhang. Spectral geometry processing. In *ACM SIGGRAPH Course Notes*, 2010.
- [MDSB03] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds.

- In *Visualization and mathematics III*, pages 35–57. Springer, 2003.
- [MRCB18] Simone Melzi, Emanuele Rodolà, Umberto Castellani, and Michael M Bronstein. Localized manifold harmonics for spectral shape analysis. In *Computer Graphics Forum*, volume 37, pages 20–34. Wiley Online Library, 2018.
- [NVT⁺14] Thomas Neumann, Kiran Varanasi, Christian Theobalt, Marcus Magnor, and Markus Wacker. Compressed manifold modes for mesh processing. In *Computer Graphics Forum*, volume 33, pages 35–44. Wiley Online Library, 2014.
- [OBCS⁺12] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012.
- [RCB⁺17] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer graphics forum*, volume 36, pages 222–236. Wiley Online Library, 2017.
- [RL01] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.
- [SOG09] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.
- [VKZHC011] Oliver Van Kaick, Hao Zhang, Ghasan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011.
- [VL08] Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pages 251–260. Wiley Online Library, 2008.
- [ZVKD10] Hao Zhang, Oliver Van Kaick, and Ramsay Dyer. Spectral mesh processing. In *Computer graphics forum*, volume 29, pages 1865–1894. Wiley Online Library, 2010.

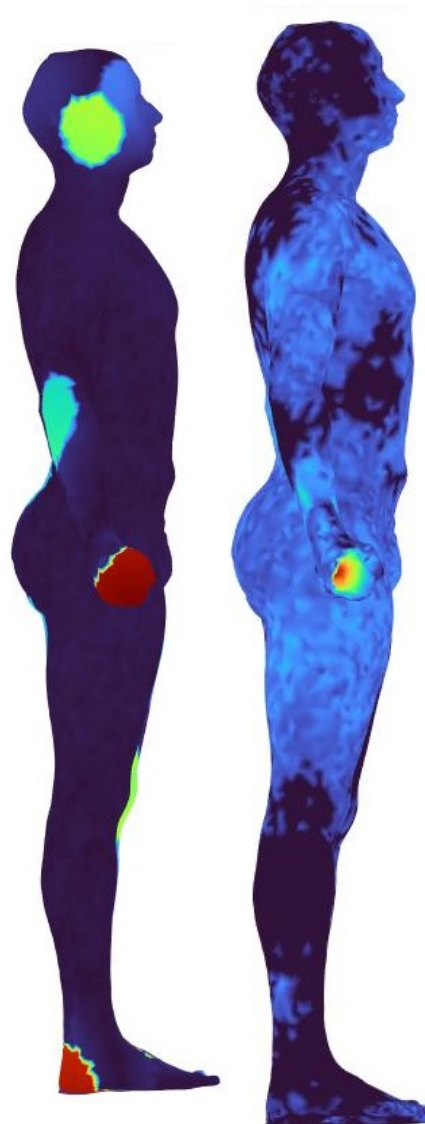


Figure 8: Geodesic Error Plot visualized as heat map on the shape itself; via Hamiltonian basis (left); via Laplace-Beltrami basis (right) for Human mesh

Using the Adaptive HistoPyramid to Enhance Performance of Surface Extraction in 3D Medical Image Visualisation

Antony Padinjarathala

School of Electronic Engineering
Dublin City University
Dublin 9, Ireland
antony.padinjarathala2@mail.dcu.ie

Robert Sadleir

School of Electronic Engineering
Dublin City University
Dublin 9, Ireland
robert.sadleir@dcu.ie

ABSTRACT

There are currently a range of different approaches for extracting iso-surfaces from volumetric medical image data. Of these, the HistoPyramid appears to be one of the more promising options. This is due to its use of stream compaction and expansion which facilitates extremely efficient traversal of the HistoPyramid structure. This paper introduces a novel extension to the HistoPyramid concept that entails incorporating a variable reduction between the HP layers in order to better fit volumes with arbitrary dimensions, thus saving memory and improving performance. As with the existing HistoPyramid techniques, the adaptive version lends itself to implementation on the GPU which in turn leads to further performance improvements. Ultimately, when compared against the best performing existing HistoPyramids, the adaptive approach yielded a performance improvement of up to 20% without any impact on the accuracy of the extracted mesh.

Keywords

Marching cubes, surface extraction, HistoPyramid, Parallel Processing, CUDA

1 INTRODUCTION

3D Medical imaging often involves extracting surfaces from volumetric datasets obtained using modalities like MRI & CT. These datasets are stored as 2D images of pixels which when combined build a 3D volume of voxels.

The surfaces in a medical image have constant density and so are called iso-surfaces. The Marching Cubes Algorithm (MCA) [Lor87], developed by Lorensen & Cline can be used to extract these surfaces. It is a robust algorithm that subdivides a volume into smaller $2 \times 2 \times 2$ overlapping neighbourhoods and processes each neighbourhood individually. This 'divide and conquer' approach is well suited to parallel processing. This method is consistent and accurate but can be quite slow.

The MCA performs computations on the whole volume, however, only a fraction of this volume will produce geometry. Alternative solutions to the MCA attempt to reduce the number of unnecessary computations that are performed. In this paper, one such solution that will be looked at is the HistoPyramid (HP) [Dyk08; Dyk10; Smi12]. The HP is a data structure that is used to transform the problem from input-centric to output-centric. This is a much more efficient solution. However, the HP approach can be

further improved to be more space-efficient and further optimised.

This paper introduces the Adaptive HistoPyramid (AHP), as a novel alternative solution to existing formations of the HP. The AHP further enhances and extends the HP such that it can operate on any arbitrary volume without the need for extra padding. It is a more flexible structure in comparison to the standard HP. Less padding should have the effect of reducing the memory required to store the AHP and also the amount of time to create the AHP.

Both the HP and AHP can be shown to be highly parallelizable and will be implemented using NVIDIA's Compute Unified Device Architecture (CUDA) which is a parallel programming platform [NVI20]. With this all of the steps involved can be executed on the GPU alone without requiring additional transfers between the GPU and CPU which adversely affects performance.

2 PRIOR WORK

2.1 Marching Cubes and Extensions

The MCA introduces key concepts that make it ideal for the task of surface extraction. First, the

3D grid of voxels is split into $2 \times 2 \times 2$ neighbourhoods formed by adjacent voxels. From this each individual neighbourhood is matched against one of the 15 Marching Cubes which approximate the geometry within the neighbourhood.

The MCA does have some limitations in its ability to handle 'sharp' shapes and those with ambiguity in certain sections. For this reason, more sophisticated methods based on the MCA such as Marching Cubes 33 [Che95] and Neural Marching Cubes [CZ21] were proposed. Marching Cubes 33 aims to improve the MCA by introducing more patterns so that there are more possible scenarios that can be modelled in each cube, thus reducing the potential for ambiguity.

Alternatively, the Neural Marching Cubes algorithm uses a cube with internal vertices and deep learning in order to create a model to recover more accurate geometry from the cube. While any of these quality-centric marching cubes options would be compatible with the performance-centric approach that is the focus of this paper, the standard MCA will be used as the starting point in order to provide the most accessible description of the technique that is being proposed.

2.2 Prefix Sum (Parallel Scan)

The Prefix Sum [Har07] is a common parallel algorithm. It can perform many parallel additions very quickly. It has many uses but the one that is most applicable for these purposes is stream compaction. Stream compaction involves modelling the problem as a series of streams and then removing unwanted or unnecessary streams. This is done by turning the MCA into streams that produce triangles with several streams for each neighbourhood and then culling the streams that don't produce triangles. The scan done with the Prefix Sum creates a cumulative sum of triangles over the entire volume and then a scatter operation is executed that selects only the streams that produce output triangles.

The MCA processes every neighbourhood which translates to outputting five streams per neighbourhood. In practice, the average number of triangles per neighbourhood will be less than one as typical medical scans feature a large amount of empty space. The Prefix Sum method uses stream compaction which ensures there are exactly as many streams as triangles. Until version 11.6 of CUDA, a sample was packaged with the CUDA

toolkit that extracted iso-surfaces using the Prefix Sum. This did not scale well with large volumes so a better solution is needed.

2.3 HistoPyramid

Another possible solution is using the HP method. The HP extends the prefix sum and creates a new data structure, a tree structure that maintains distribution information, instead of using a single compacted array of streams. Similar to the Prefix Sum it uses stream compaction to reduce the problem into a stream of triangles. The HP base layer has an entry for each neighbourhood which is the number of triangles that will be produced by the neighbourhood. It makes upper layers by summing entries at each layer until there is only one entry at the top layer which is the total number of triangles. These triangles are passed through each layer of the HP to find the position and dimensions of that triangle. Often a neighbourhood may produce multiple triangles, but each traversal of the HP produces exactly one triangle. These triangles make up the surface as before.

2.4 Parallel Processing

A great benefit associated with the MCA is that it is highly parallelizable. Even running the standard MCA on a GPU can result in a reduction in processing time. Parallel implementations of the MCA have been tested [Arc11] and can take between four and fifty times less time to process by utilising the GPU. However, memory issues may become a problem which limits the benefit that can be gained when processing larger volumes.

One reason is that parallel processing on a GPU tends to require that each parallel process has a fixed allocation of memory to deal with all possible input/output scenarios. i.e., each neighbourhood outputs the maximum number of triangles which is five even though in most cases no triangles will be produced. This will inevitably lead to inefficiencies in terms of memory requirements and processing speed. On some GPUs it may not be possible to process larger volumes in a single pass without having to do additional slow GPU to CPU transfers creating a bottleneck.

The HPs are highly parallelizable and the output is a compact set of triangles so it doesn't require as much memory despite having to store a full

HP data structure as well as storing the full set of underlying $2 \times 2 \times 2$ voxel neighbourhoods. The layers of the structure are composed of summing sections of previous layers, a process which can be carried out in a computationally efficient manner. The HP is a Pyramid of partial sums. With the top layer being the sum of all entries in the base layer. This is similar to a Prefix Sum operation that outputs intermediate layers. The HP layers split the work done by the Prefix Sum method and allow it to be traversed much faster while taking the exact same time to create. And, since the traversal is a simple algorithm that iterates over a number of triangles it is easy to implement in parallel.

3 DESIGN AND IMPLEMENTATION

3.1 Marching Cubes Algorithm

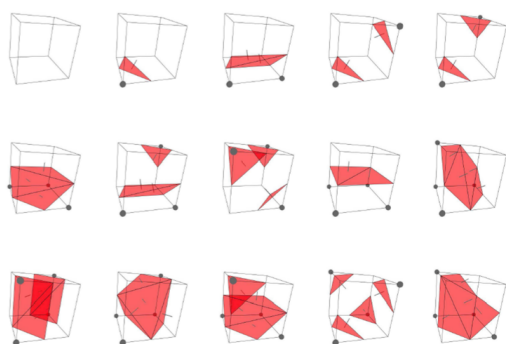


Figure 1: The 15 marching cubes cases.

To implement the MCA, the 3D grid of voxels is first split into $2 \times 2 \times 2$ neighbourhoods formed by adjacent voxels. Each neighbourhood is a $2 \times 2 \times 2$ overlapping region from the original dataset. Every voxel forms a corner of the neighbourhood and has a domain-specific value for its density.

3.1.1 Thresholding

For each neighbourhood, voxel densities are compared with a threshold. This threshold is the density of the surface to be extracted. Voxels are thus assigned as internal or external to this surface.

3.1.2 Identify Intersected Edges

If a neighbourhood contains voxels that are both external and internal to the surface, then it must be intersected by the surface. There are 256 ways that a neighbourhood can be intersected. However, using rotations and complementary cases,

this can be reduced to only 15 unique scenarios as shown in Fig. 1. Using a lookup table produced by Lorensen and Cline, a set of intersected edges was found in that neighbourhood that must contain vertices of the surface.

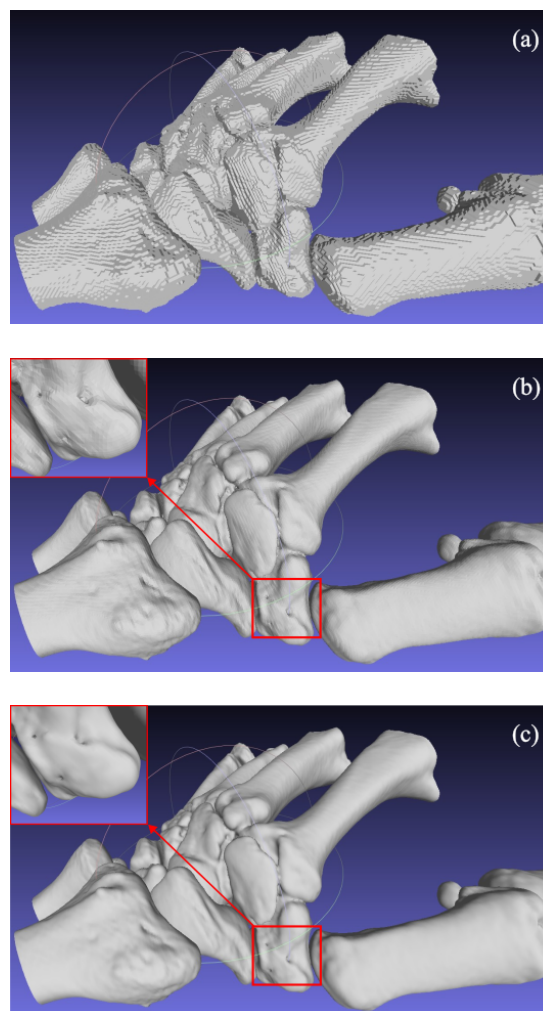


Figure 2: Evolution of the mesh extracted using the MCA. (a) A surface extracted before any interpolation occurs. (b) A surface extracted without calculating vertex normals. (c) A fully extracted surface with vertex normals for lighting calculations.

3.1.3 Interpolation of Points

Linear Interpolation is applied using the densities of the voxels on either end of an intersected edge and the threshold value to approximate the exact position of the vertex along this edge. Interpolation stops the rendered surface from looking 'blocky' as in Fig. 2(a) and instead look as they do in Fig. 2(b). After this the vertices are triangulated.

3.1.4 Calculating Vertex Normals

At this point the mesh is accurate but does not have the information required to facilitate per fragment lighting calculations. Consequently, normals must be calculated for each vertex. This is done for each voxel by finding the rate of change in voxel density along each axis local to that voxel using neighbouring voxels. Vertices lying on an edge containing those two voxels have their normals found using linear interpolation as before.

The results were examined to ensure that a smooth, high detail surface was produced as illustrated in Fig. 2(c). After the MCA was implemented on the CPU, the process was repeated on the GPU using CUDA and the resulting surface was verified to be identical to the surface produced by the CPU bound version of the algorithm.

3.2 HistoPyramid

The HistoPyramid is a data structure that will allow a faster extraction phase and is ideal for GPU implementations. It requires additional setup time; however this time is negligible in comparison to the gains made during the extraction phase.

3.2.1 First Pass

As before, the volume is split into overlapping neighbourhoods and thresholding is applied. However, the voxels from this are used with the lookup tables to quickly calculate the number of triangles that will be produced by each neighbourhood without performing any time-consuming calculations.

3.2.2 Constructing the HP

It has been suggested that accessing the data as a series of tiled 2D slices [Har07; Dyk08] rather than the original 3D sub-volumes has the potential to significantly reduce the computational overhead associated with indexing through the HP data. The HP base layer in this case will be a 2D array with each entry representing the number of triangles in a given neighbourhood. The layer above this is constructed by summing entries in the layer below. This reduces the size of each upper layer by a factor of four until one entry remains at the top layer that contains the total number of triangles produced by the surface.

This is a reduction factor r of four or 2×2 . It should be noted that for this reduction to be possible, the bottom layer must have sides of equal

length that are of size 2^k and will form $k + 1$ layers. The layers are padded to fit these constraints. The formation of the HP can be thought of as a Prefix Sum with Intermediate Partial Sum layers.

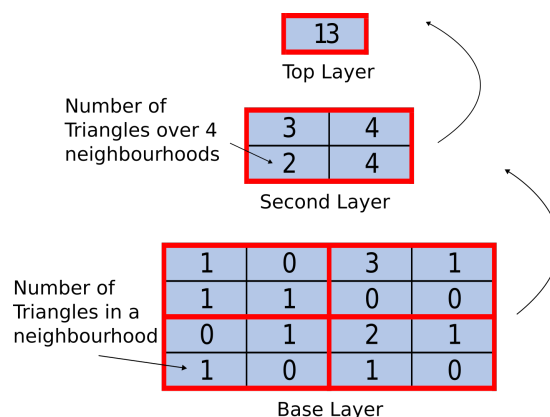


Figure 3: Construction of a 2D HP with 16 entries in its base layer.

For the above case, the reduction factor is four which equates to a 2×2 square. However, it is possible to sum over any area that is an $n \times m$ rectangle. This requires that the base layer is of area $n^k \times m^k$ and that there are $k + 1$ layers. Above, in Fig. 3, a 4×4 area is used to represent a possible $2 \times 2 \times 4$ volume.

3.2.3 Traversing the HP

Next the HP is traversed to find each triangle in the surface. The number of triangles equals the top entry since it is a sum of all base layer entries. The HP is traversed using indices from 0 to $N - 1$, where N is the top entry in the HP to find all N triangles. Instead of iterating on an entire volume like the MCA, the HP iterates on all the triangles in a volume and then the HP is traversed to find the exact position of each triangle.

This is done by finding which sub-area of a lower layer that a triangle belongs to. Once the base layer is reached, the neighbourhood containing the triangle along with its corresponding vertex positions are found. Each parallel process in this is a stream that produces exactly one triangle. This stream compaction transforms the basis of the problem such that it is based on the number of triangles instead of the size of the volume. This new basis means that the complexity is related to the length of the output which is why it is referred to as an output-centric method. MCA on the other

hand has complexity related to the input volume making it input-centric.

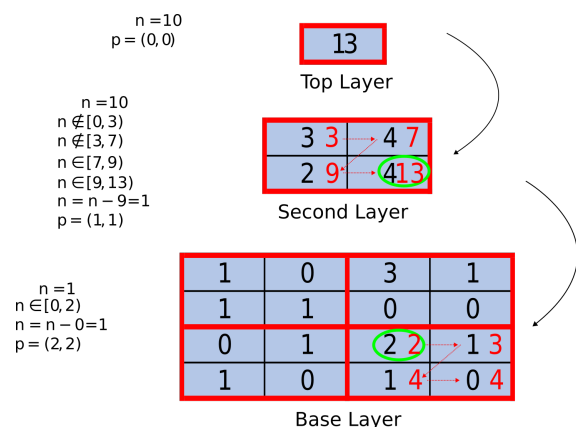


Figure 4: Traversal of a HP to find the location of the 10th of 13 triangles in a surface.

In Fig. 4, the n^{th} triangle is being extracted for $n = 10$. The entries in each 2×2 square are ordered starting at the top left and following the arrow in the figure. At each layer a cumulative sum of the entries is calculated as shown by the number in red to calculate the ranges contained within each entry. At the top layer, n must be within the area beneath the first entry. At the second layer the ranges contained by each entry are created. In the example, n belongs to the range created by the last entry. n is updated to be local to that entry and given a value of one. The position value p is updated also to point at this entry by multiplying the position vector by the reduction factor. In the base layer, n is within the first entry. The triangle is within this neighbourhood in the base layer. This neighbourhood produces multiple triangles and n decides which triangle to extract from the neighbourhood.

Unlike the construction phase, the number of instructions to retrieve the position of the triangle in a lower layer is not always constant since the process may need to check multiple entries. This can potentially result in slightly slower processing because conditional logic may lead to branch divergence and as a consequence the streams may vary in terms of the time required to execute. These processes run on threads, and in CUDA, threads run in collections of 32 called warps. These warps must be synchronised and if the execution time varies greatly the performance is affected negatively. Execution of a warp will continue until its longest running member thread

has completed its operation.

3.3 HP Modifications

Different formations of the HistoPyramid are possible by changing the way that reductions are applied.

3.3.1 3D HistoPyramid

The HistoPyramid can have 3D layers [Smi12]. In this scenario, the 3D base layer is not flattened into a tiled area. Instead, the 3D data is processed directly with constraints similar to those that apply to the 2D case. It is possible then to divide any volume of $n^k \times m^k \times l^k$ where n, m and l are the reduction factors in each dimension and $k + 1$ is the number of layers. Only $2 \times 2 \times 2$ reduction factors were considered in this case which is equivalent to an overall reduction factor of eight.

3.3.2 1D HistoPyramid

This HistoPyramid can also be flattened down to 1D layers [Dyk10]. This has the potential to reduce the indexing overhead even more than the 2D case. It also loosens the constraints on the size of the base layer to be any length that can be written in the form r^k where r is the reduction between each layer and $k + 1$ is the number of layers. A series of 1D HPs were evaluated and the best performing of these were considered further. As observed from experiments carried out, the best performing HPs had reduction factors of five or eight.

3.4 Adaptive HP

The HP makes the traversal phase and surface extraction phase of the process optimally efficient as they are output-centric. This is not the case for the first pass over the volume and the creation of the HP. As noted from repeated experimentation, most of the processing time is associated with the input-centric computations. These computations took on average 84% of the total computation time for surface extraction. The input-centric computations took more time for sparser datasets, especially where a large amount of padding of the HP structure is required to accommodate the HistoPyramid size constraints, while the opposite was the case in datasets that contained more extensive meshes. Reducing the effect of input-centric processes requires removing the padding created from the HP. To do this, the AHP

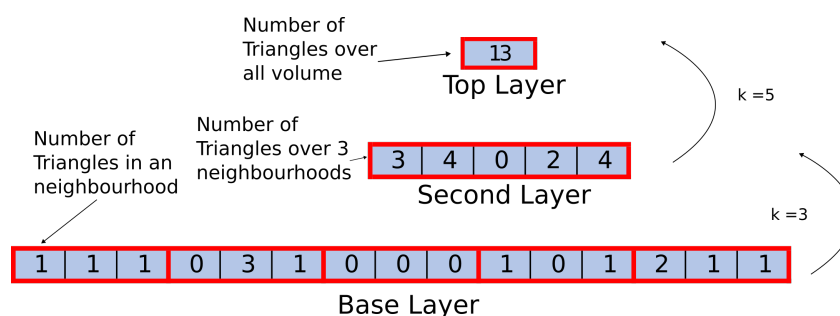


Figure 5: AHP removing the need for padding in a volume of fifteen neighbourhoods by using reduction factors of three and five.

can be used.

Starting with a 1D HP, the AHP takes advantage of the fact that a constant reduction between layers is not necessary. This is possible as long as each layer is treated the same at construction and traversal. Fig. 5 shows how the example used in Fig. 3 can be reshaped in the form of an AHP. With this the layers have a reduction of three and then five to make up the size of exactly fifteen which could not be expressed in the form r^k or by any of the HPs explored thus far.

The AHP must be traversed in the reverse order to construction to find the correct positions. As seen in Fig. 6, the traversal is identical to the HP and selects the same entry for $n = 10$ and for any other n , the main difference is how the layers are divided. The AHP aims to reduce the padding by a range of different reduction factors and finding an arrangement of these that will produce the least padding. The constraints on this structure are that the base layer is of size $\prod_{i=0}^K r_i$ where r_k is the reduction at layer k .

Any size base layer can theoretically be accommodated given that it is not a prime number. This significantly reduces the need for padding compared to other similar techniques. In fact,

it could be possible to take only a subset of an overall volume using this method and thereby reducing the memory requirements further. Larger reductions at each layer will produce fewer layers however it will also increase the amount of data at each layer which would slow down processing at that layer.

A method is required for determining the reduction factors. The maximum reduction factor found in existing works is eight [Smi12] but tended to be lower e.g. four or five [Dyk08; Dyk10]. With the AHP it will be possible to use larger reductions without possibly creating a large amount of padding. This is because, for a regular HP the HP can only take a limited set of sizes which are dependent on the reduction factor and this set of sizes becomes sparser as the reduction factor increases as the base layer size must be exactly r^k . Conversely, the AHP uses multiple reduction factors so does not have the same constraints on size as is the case with HP, so this is not an issue.

The reduction factors for a given volume were selected from a set of numbers ranging from 4-16. Any factor is a valid choice, even so, it is important to select factors that are not too large as it was determined that this created layers with a large possibility of branch divergence which

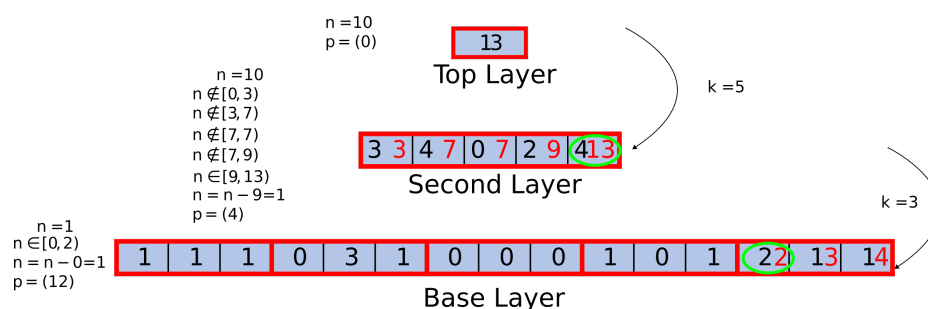


Figure 6: Traversal of a AHP to find the 10th of 13 triangles of a surface.

would negatively affect performance.

The reduction factors were determined empirically from the range of values from this set for this initial implementation. This was deemed adequate as it was generally possible to find an outcome that was close to ideal with very little computation overhead. The HP could be a possible configuration of the AHP so, in theory, an ideal AHP would at most take as much storage as most space-efficient regular HP.

4 RESULTS AND ANALYSIS

The algorithms were run on nine different datasets [Sta87; Ack98; Kik14; McC14] obtained from a range of sources. Table 1 presents a summary of each of the methods compared with the standard MCA as well as the additional memory requirements to store the base layer of the HP over all datasets. Additionally, detailed results are provided for three exemplar datasets that show the performance of the technique across a range of different modalities, dataset dimensions and surface morphologies.

All algorithms were evaluated on a PC with an Intel Core i7 2.60GHz CPU with six cores and 32 GB RAM, with a 4GB NVIDIA Quadro P2000 Graphics Card. The algorithms were implemented using version 4.8 of the .NET framework which incorporated CUDA 11.6 through the NuGet Package ILGPU [ILG22].

Table 1 shows the average relative performance from timing the various algorithms. The standard MCA implementations are denoted as MCA CPU and MCA GPU. 1D HP implementations are denoted by HP and then a number relating to the reduction. Only 2 of these were considered, HP5 and HP8, which have reduction factors of five and eight respectively. These two were the best performing of the 1D HPs. AHP refers to the adaptive approach and the final two algorithms, HP2D and HP3D refer to implementations that have 2D or 3D layers respectively.

The HistoPyramids reduce the number of computations and performs these computations in parallel, so a large improvement is expected. This approach was found to be 50 to 500 times faster than the MCA to extract the same surface. The benefit of this technique is clear from these results.

The AHP performs better than any of the other implementations. On average it performed 4.5% faster than the next best performing method for each individual case, for the volumes that exceeded $256 \times 256 \times 256$ this improvement increases to an average of 10%. In the best case from the dataset this improvement increased to 20%. The performance of the AHP was also found to improve as the size of the volume increased.

In addition to this the AHP was generally found to be the best method across datasets. In one outlier case, the AHP was not the top performer as it took slightly longer than the HP5 method. In this case, the AHP used a large amount of padding and has not tended to the best formulation of layers. The HP5 method uses less padding for this case. There is at least one better solution to the arrangement of AHP layers since any 1D HP is also a possible solution to the AHP. Using a different algorithm for determining the reduction factors of the AHP would reduce the padding used and improve performance further. This outlier demonstrates that padding is a relevant factor in performance.

The 1D implementation tends to perform better than 2D or 3D alternatives. Using 1D layers eases the constraints on the shape and size of the original volume. Additionally, the index overhead of using 3D or 2D indices for each layer is not present with the 1D implementation. For irregularly shaped data this can be particularly apparent. The AHP tends to use even less padding especially when averaging results over several datasets. It will produce a consistently low amount of padding while with other forms of the HP, the amount of padding can vary significantly from dataset to dataset. This may be because it can model any arbitrary volume easily. Because of this the AHP is particularly fast in the construction phase.

5 CONCLUSIONS

The AHP evaluated was an initial unrefined implementation and yet it regularly performed better than the other methods. It enhances the HP using a variable reduction factor between layers of the data structure. This results in a performance boost of up to 20% with an additional benefit of using less of the finite memory available on a GPU. These performance boosts do not sacrifice the accuracy of the mesh extracted. In the outlier case, where the AHP doesn't give a benefit over a 1D HP, which is quickly calculable before creating any structure, a hybrid solution might be used

	MCA CPU	MCA GPU	HP5	HP8	AHP	HP2D (HP4)	HP3D (HP8)
Average Performance over 9 datasets							
Relative Time to MCA	1.000000	0.244867	0.005974	0.006015	0.005722	0.006558	0.007677
Padding Required			86.56%	139.06%	8.98%	105.93%	258.50%
MRHead (130 x 256 x 256)							
Relative Time to MCA	1.000000	0.341110	0.004098	0.004229	0.004232	0.004216	0.004643
Padding Required			14.85%	98.44%	22.03%	98.44%	98.44%
F_Head (234 x 512 x 512)							
Relative Time to MCA	1.000000	0.229836	0.006884	0.006498	0.005701	0.009239	0.011601
Padding Required			300.45%	243.90%	0.16%	9.48%	119.78%
Wrist CT (251 x 440 x 440)							
Relative Time to MCA	1.000000	0.485414	0.009634	0.010373	0.009393	0.015005	0.018657
Padding Required			0.49%	177.72%	7.60%	38.43%	177.72%

Table 1: The relative time taken for each method relative to the Baseline MCA CPU and the extra padding required for each HP and AHP. The best performing algorithms for each surface are highlighted.

instead that could force the use of the most appropriate conventional HistoPyramid-based approach.

There are some areas in which the AHP could be improved. Namely, the final structure is often not the optimal solution and, moreover, uses more padding than needed. This happens because the method for selecting factors reaches an acceptable solution but there are often better solutions available, possibly by refining an initial guess or by using a more analytical approach. Additionally, the AHP can readily select subsets of the volume. For example, a pre-processing step could crop only important parts of the volume as much of it is empty space. Or a computer vision task might quickly detect subvolumes from larger datasets that might need to be selectively extracted, something that the AHP would easily accommodate. Future work might see a better method for selecting factors while also taking advantage of the AHP's ability to model arbitrary volumes to see more significant gains.

As with the HP, the AHP is a parallel-first approach making it ideal for GPU implementations. This avoids any CPU to GPU transfers which will cause a bottleneck. Each step of the HP consists of only additions or comparisons. It is therefore easy to comply with the SIMD or SIMT models of parallel computing and ensure that maximum use is being made of the GPU.

REFERENCES

- [Ack98] Ackerman, M. "The Visible Human Project". In: *Proceedings of the IEEE* 86 (Mar. 1998), pp. 504–511.
- [Arc11] Archirapatkave, V., Sumilo, H., See, S. C. W., and Achalakul, T. "GPGPU Acceleration Algorithm for Medical Image Reconstruction". In: *2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications*. 2011, pp. 41–46.
- [Che95] Chernyaev, E. V. "Marching Cubes 33: Construction of topologically correct isosurfaces". In: (Nov. 1995).
- [CZ21] Chen, Z. and Zhang, H. "Neural Marching Cubes". In: *ACM Trans. Graph.* 40.6 (Dec. 2021).
- [Dyk08] Dyken, C., Ziegler, G., Theobalt, C., and Seidel, H.-P. "High-speed Marching Cubes using HistoPyramids". In: *Computer Graphics Forum* 27.8 (2008), pp. 2028–2039.
- [Dyk10] Dyken, C. and Ziegler, G. "GPU-accelerated data expansion for the Marching Cubes algorithm". In: *Proc. PGU Technol. Conf.*, 2010.
- [Har07] Harris, M., Sengupta, S., and Owens, J. "Parallel prefix sum (scan) with CUDA". In: vol. 39. Aug. 2007, pp. 851–.
- [ILG22] *ILGPU*. <https://www.ilgpu.net/>.

- [Kik14] Kikinis, R., Pieper, S. D., and Vosburgh, K. G. "3D Slicer: A Platform for Subject-Specific Image Analysis, Visualization, and Clinical Support". In: *Intraoperative Imaging and Image-Guided Therapy*. Ed. by F. A. Jolesz. New York, NY: Springer New York, 2014, pp. 277–289.
- [Lor87] Lorensen, W. and Cline, H. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm". In: *ACM SIGGRAPH Computer Graphics* 21 (Aug. 1987), pp. 163–.
- [McC14] McCormick, M., Liu, X., Jomier, J., Marion, C., and Ibanez, L. "ITK: Enabling Reproducible Research and Open Science". In: *Frontiers in neuroinformatics* 8 (Feb. 2014), p. 13.
- [NVI20] NVIDIA, Vingelmann, P., and Fitzek, F. H. *CUDA, release: 10.2.89*. 2020.
- [Smi12] Smistad, E., Elster, A., and Lindseth, F. "Real-Time Surface Extraction and Visualization of Medical Images using OpenCL and GPUs". In: Jan. 2012.
- [Sta87] *The Stanford Volume Data Archive*. <http://graphics.stanford.edu/data/voldata/>.

Semi-Supervised Learning Approach for Fine Grained Human Hand Action Recognition in Industrial Assembly

Fabian Sturm
Bosch Rexroth AG
Lise-Meitner-Strasse 4
89081 Ulm, Germany
fabian.sturm@bosch.com

Rahul Sathiyababu
Bosch Rexroth AG
Lise-Meitner-Strasse 4
89081 Ulm, Germany

Elke Hergenroether
University of Applied
Sciences Darmstadt
Schoefferstrasse 3
64295 Darmstadt,
Germany

Melanie Siegel
University of Applied
Sciences Darmstadt
Schoefferstrasse 3
64295 Darmstadt,
Germany

Abstract

Until now, it has been impossible to imagine industrial manual assembly without humans due to their flexibility and adaptability. But the assembly process does not always benefit from human intervention. The error-proneness of the assembler due to disturbance, distraction or inattention requires intelligent support of the employee and is ideally suited for deep learning approaches because of the permanently occurring and repetitive data patterns. However, there is the problem that the labels of the data are not always sufficiently available. In this work, a spatio-temporal transformer model approach is used to address the circumstances of few labels in an industrial setting. A pseudo-labeling method from the field of semi-supervised transfer learning is applied for model training, and the entire architecture is adapted to the fine-grained recognition of human hand actions in assembly. This implementation significantly improves the generalization of the model during the training process over different variations of strong and weak classes from the ground truth and proves that it is possible to work with deep learning technologies in an industrial setting, even with few labels. In addition to the main goal of improving the generalization capabilities of the model by using less data during training and exploring different variations of appropriate ground truth and new classes, the recognition capabilities of the model are improved by adding convolution to the temporal embedding layer, which increases the test accuracy by over 5% compared to a similar predecessor model.

Keywords

Human Action Recognition, Industrial Assembly, Semi-Supervised Learning, Transfer Learning, Transformer

1 INTRODUCTION

The full automation of production processes in industrial production lines has shown that the contribution of humans cannot be completely replaced by machines, yet. This adapted attitude toward full automation is primarily due to monetary reasons, such as increased fixed costs due to the maintenance of the machines used in the process, but also to reasons such as lower social acceptance due to the elimination of certain occupational groups. The advantages resulting from and utilized by humans compared to today's machines are primarily evident in manual assembly work. Humans are able to process a high product variance, to adapt to new or optimized processes without additional technical ef-

fort, and to recognize and adjust to unexpected problems and assemble components very accurately and precisely. However, the increasing variance of products and shorter time-to-market for companies is becoming a disadvantage for people due to the increase in workload. This workload leads to a higher error rate, especially when it comes to assembling products. The main reason for this is lack of concentration, especially due to long work shifts, as well as inattention or distraction. In addition, the assembly worker needs more and more in-depth knowledge to assemble the products correctly and has less time for training and instruction. These circumstances lead to a lack of expertise and reduce positive human flexibility. The errors in assembly, which initially go undetected, then continue through the entire production process until they are noticed during quality control or, in the worst case, during operation of the product. The consequences are inconvenience to production, an increased number of defective products or damage to the company's image because of the low quality. One possible solution to avoid the aforementioned problems in assembly without affecting the economic parameters are assistance systems for the worker.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

These assistance systems guide the employee through the assembly process and point out errors. In addition to detecting the objects used, such as tools and components, the first step in this observation of assembly tasks is to observe the fine-grained human actions performed by the hands of the assemblers. For the recognition and description of these fine-granular hand actions, a stacked deep-learning architecture is used. This deep-learning model architecture consists of an existing feature extractor for hand detection and tracking and subsequent classification of the sequences by an existing for this specific use case adapted spatio-temporal transformer model based on the findings of [Liu+21]. Beside the presentation of the architecture and usage of an available dataset for fine-grained human hand actions in industrial environment, [Stu+23] the main focus of this paper is the semi-supervised model training procedure dealing with the general problem in an industrial environment with lots of data but few labels. This circumstance is a relevant point at the latest when the trained basic application needs to be adapted to a new real world use case in order to check the scalability of the model. Especially in an industrial environment, this must be done as quickly and cost-effectively as possible and, without time-consuming labelling effort, as is the case with traditional deep learning methods. In this work, pseudo-labelling transfer learning experiments are therefore investigated to improve the stability of each class and the overall performance and generalization possibilities of the model. A base model trained supervised on different combinations of assembly tasks with the highest F1-Score[GBV20] from the "Industrial Hand Action Dataset V1"[Stu+23] is pretrained to provide these weights as initialization for an industrial real world use case from a laboratory environment. For the subsequent real world use case, it is assumed that partial labels based on the job description and example assembly steps for training new employees exist, which are in this case a combination of novel classes from the "Industrial Hand Action Dataset V1". The pretrained weights are transferred to a new classifier architecture, which is subsequently fine-tuned on the novel classes in a semi-supervised procedure by using at first the labeled example data from the customer and initialize the model for the training on unlabeled data. Afterward, the pseudo-labeling approach follows. This procedure shows how this method of semi-supervised training can positively affect the classification results in an industrial environment. It provides more stability and generalizability compared to supervised approaches for an industrial real world use case and is based on a spatio-temporal transformer network for fine-grained human hand actions. In the following Sections, the training data set is introduced in Section 2 followed by the stacked model architecture in Section 3 existing of the hand detector in Section 3.1 and the spatio-temporal

transformer model for the classification task in Section 3.2. The related work 4 of pseudo-labeling a method of semi-supervised learning is presented in Section 4.1 and the shown approaches are revisited under consideration of the usability in this specific industrial human hand action recognition approach in Section 4.2, before the experiments are examined in Section 5 and finally evaluated, compared and concluded in Section 6 and Section 7.

2 INDUSTRIAL DATASET

The "Industrial Hand Action Dataset V1", a vision based industrial hand assembly dataset introduced in [Stu+23] consists of 12 fine-grained hand action classes for industrial assembly. With 459,180 frames in the basic version and 2,295,900 frames after spatial augmentation, uneven distributed it belongs to one of the larger datasets. It is the first dataset of its kind to tackle the real world issues which occur in an industrial environment. Compared to other freely available datasets tested in [Stu+23], it has an above-average duration and, in addition, meets the technical and legal requirements for industrial assembly lines. Furthermore, the dataset contains occlusions, hand-object interaction, and various fine-grained human hand actions for industrial assembly tasks that were not found in combination in similar examined datasets [Stu+23]. The recorded ground truth assembly classes are selected after extensive observation of real-world use cases. The usability of the dataset for training sequential deep learning models was confirmed in [Stu+23] with a test accuracy of 86.25% before hyperparameter tuning. The architecture is based on an adapted version of the gated transformer network model of [Liu+21], presented in more detail in section 3.2.

3 MODEL ARCHITECTURE

The processing of full image sequences is a very resource intensive task when it comes to deep learning model training. Since the focus of the model architecture in the described industrial use case is on the hands, the decision was made to apply a skeleton-based human hand action recognition approach. Hand coordinates are extracted from frames to get a reduced feature space data sequences. This lowers the computational effort for the subsequent classification task by providing to the model 2.5 dimensional coordinates per frame without any noise from tools, unnecessary objects or other assemblers. This can be seen in Figure 2, or for the classification of unnecessary information of the in industrial environment usual static background, see Figure 1. Subsequently, for the classification of the performed assembly task of the hands, the sequences of coordinates per frame are provided to the adapted Gated Transformer Network by [Liu+21]. They achieved good re-

sults on the initial work on the dataset in [Stu+23] and on comparable datasets in [Liu+21].

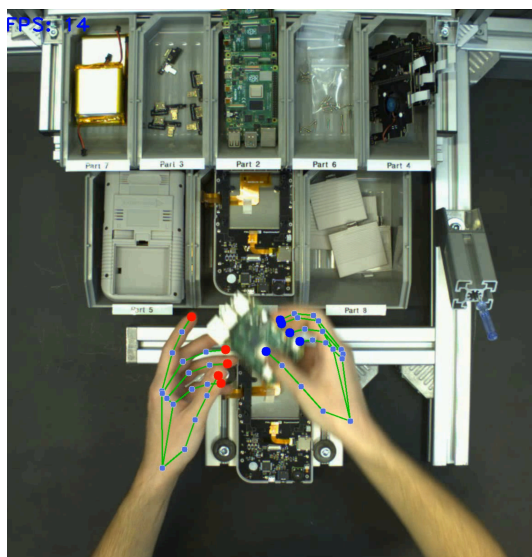


Figure 1: Static Background in Industrial Environment

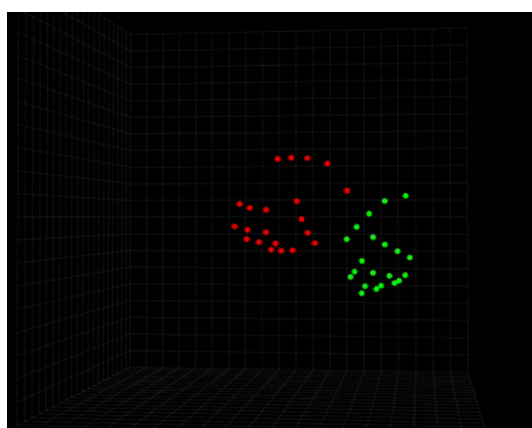


Figure 2: Feature Space for Sequential Model Training

3.1 Hand Keypoint Extractor

The keypoint extraction is based on an approach similar to the architecture of Google's MediaPipe Hands solution, with a palm detection model and a hand feature detector on top [Zha+20a]. The output of this model architecture is able to provide 21 2.5D coordinates for different landmarks, which are used as pre-extracted features or keypoints for the subsequent transformer model architecture. The keypoint extractor starts with a palm detector network, which is a single shot detector optimized for mobile real-time usage [Zha+20a]. It takes advantage of first recognizing the palms through a bounding box, which is a more stable approach for the model than first recognizing the entire hands with fingers due to its square shape. Further, an encoder-decoder feature extractor similar to a Feature Pyramid Network [Lin+16] is used to detect the palm across a

large range of scales. Focal Loss [Lin+17] is used as the loss function, since it handles the imbalance between background segment detections and actual palm detections better by reducing the influence of background detections during training [Zha+20a]. Once a palm has been detected, a cropped image is generated based on the mentioned detection. This includes more image data than the palm bounding box itself in order to contain the entire hand. This cropped image is then provided to a second network consisting of a convolutional neural network to detect the hand landmarks and outputs the 21 hand landmarks with 2.5D coordinates. The values consist of x, y values which are calculated by the size of the frame and a depth value relative to the wrist landmark. Furthermore, a probability of a hand being present and the handedness is provided. Based on the detected landmarks, a new crop area is calculated to try to keep the hand within this area. The next image in the sequence is then cropped to this new value straight away, without the single-shot detector running again. Only if the probability of a hand being present is below a certain threshold, the single-shot detector runs again on the entire image [Zha+20a]. Since the convolutional neural network has to run only on a smaller cropped image and the single-shot detector only has to scan the entire image if a hand has been lost, the number of compute cycles required during inferencing is reduced.

3.2 ConvGTN

The output of the hand keypoint extractor leads to the second part of the architecture. This part classifies the temporal and sequential correlation of the previously extracted time series of keypoints by a gated transformer network from [Liu+21] which is adapted to this use case, see Figure 3. [Liu+21] achieved state-of-the-art results on 13 multivariate time series classification tasks in the domain of Natural Language Processing (NLP), but also human action recognition tasks which are comparable to this use case [Liu+21]. The architecture is based on a two-tower transformer, where the encoder in each tower capture time-step-wise and spatial-channel-wise attention. To merge the encoded feature of the two towers, a learnable weighted concatenation is used as a gate before the final fully connected layers. This gate decides which tower of the network provides more important features for the final classification during backpropagation. For the improvement of the prediction results of the model, an additional Conv1D¹ with kernel size of 5 is implemented. This additional convolution helps the model to find better correlation between the hand keypoints [21*3*2] [key-points per hand*xyz coordinates*hands], in the temporal embedding of the model and leads to better gradients

¹ <https://pytorch.org/docs/stable/generated/torch.nn.Conv1d.html>

in the temporal tower at each time step of the input data, [8,126,100] [batch size, features, sequence length]. The convolutional layer is followed by a linear layer with 512 input and 512 output features and leaky rectified linear unit (LeakyRelu) as activation function [Xu+15], which is added to the temporal embedding layer for better generalization performance, see Figure 3.

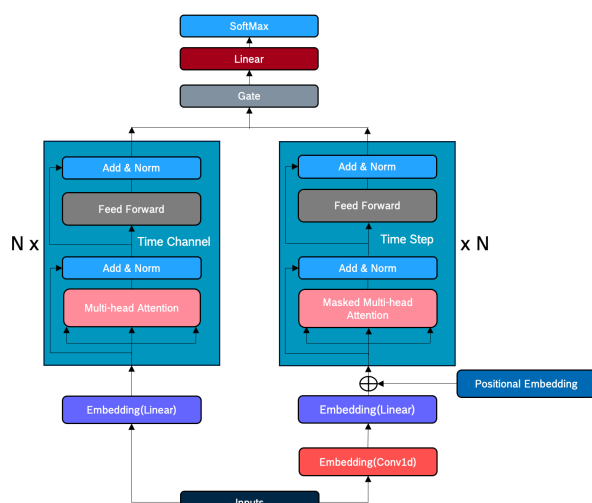


Figure 3: ConvGTN Model Architecture

4 RELATED WORK

The choice of a semi-supervised training method for the presented model architecture is based on the large amount of recurring patterns in the data generated by industrial processes and the precise instructions given to assemblers before starting an assembly task. In the process, the employee is first shown assembly tasks as ground truth via training videos, which he or she is then expected to perform. When performing the assembly steps, a wide variety of tasks occur, especially by different employees. In addition, there are also tasks that are similar but must be performed differently due to the variance of the components. However, these have the same specifications in the task description and can thus cause a high variance and positive influence on the generalizability of the model to be trained. The approach envisaged for this is to train a ground truth on a common specification and use it as a basis to transfer, recognize, and learn the recognition of new or similar assembly steps faster. For this purpose, a base model is first trained on labeled data, and this knowledge is then used as base initialization knowledge for new classes. This is done by replacing and re-training the final classification layers through fine-tuning [Far+21]. In order to remain appropriate to the real condition in the production, only a small portion of marked data is initially used. The unlabeled new data is then iteratively labeled by the model, and the entire model is re-trained from the transferred state to fine tune parts of the model on the new knowledge. For this purpose, a method of semi-supervised self-training [GB04] is used, more specif-

ically pseudo-labeling. The labeled data set is augmented by self-training to detect better relationships in the data through self-supervised learning in the unsupervised domain [Zho+18].

4.1 Self-Training for Classification Tasks

Most of the work in the subcategory of self-training, pseudo-labeling, concentrate on image classification [Yan+21][Wen+21]. Especially in this use case of fine-grained human hand action recognition or skeleton based human action, very few studies exists whether it is based on videos nor previously extracted features [Xu+21][ITP14][Xia+21]. The difference between a training approach with labels and without labels like in this case is that the algorithm uses the model's own trusted predictions to create the pseudo-labels for unlabeled data and can add more training data by using existing optimally self-labeled data to predict the labels of the unlabeled data [Ami+22]. The firstly proposed approach of pseudo-labeling by [Lee13] uses a small set of labeled data to iteratively train a model in combination with a large set of unlabeled data. A small set of labeled data is used to iteratively train a model in combination with a large set of unlabeled data. Using cross entropy loss, this involves training a base model, which is then used to make a prediction on a batch of unlabeled data. These predicted labels are then added to the labeled data set, the model is trained again on the data set, and the next batches are predicted until, in the optimal case, there is no unlabeled data left. The loss function is defined as follows, see Equation 1 [Lee13].

$$L = \frac{1}{n} \sum_{m=1}^n \sum_{i=1}^C L(y_i^m, f_i^m) + \alpha(t) \frac{1}{n'} \sum_{m=1}^{n'} \sum_{i=1}^C L(y_i'^m, f_i'^m) \quad (1)$$

In this equation, n defines the number of mini-batches in labeled data for stochastic gradient descent (SGD) and n' the number of mini-batches for unlabeled data. f_i^m is the output unit of m samples in labeled data, and y_i^m is the label of the samples. $f_i'^m$ is a sample as well, just for unlabeled data, and $y_i'^m$ is the pseudo-label of that. $\alpha(t)$ is a coefficient balancing the supervised and unsupervised loss terms by increasing the influence of the unsupervised loss per each iteration. In general, the pseudo-label approach takes entropy minimization to get the pseudo labels with the highest confidence as the ground truth for unlabeled data [Lee13]. An improved version of the traditional pseudo-labeling is the Meta Pseudo-Labeling (MPL) [Pha+20]. In this approach, a teacher model assigns distributions to the inputs and thus trains the student model. While training the student, the teacher observes the student's performance on a validation set and learns to generate target distributions so that when the student learns from such distributions, the student performs well in validation. The MPL training procedure consists of two alternating

processes. The teacher generates the conditional class distribution for the student's training. This pair is then fed into the student network to update its parameters based on the cross entropy loss. After the student network updates its parameters, the model evaluates the new parameters based on the samples from the validation dataset. Since the student's new parameters depend on the teacher, the gradient of loss can be calculated based on the dependence to update the teacher's parameters. Similar to this two model approach is Cross-Model Pseudo-Labeling, which focuses on the problem of having less labeled data where a single model is not able to provide good enough pseudo labels [Xu+21]. As shown in their experiments, a deeper model can find better spatial correlations, while a smaller model is better at detecting temporal correlations sequentially. This leads to the result that two models that are different sizes can be used for pseudo-labeling to better distinguish their found labels. The models predict each other's pseudo-labels and use them to train or, more precisely, subsequently back propagate the unsupervised loss through themselves to improve performance. [Xia+21] made the same results for spatial and temporal correlations, but in video-based action recognition, especially in cases with fewer data. They improved performance with a blockwise dense alignment strategy and cross modal contrastive learning, focusing the model on the temporal dynamics of videos by computing a temporal gradient. These methodologies shown so far are used in many approaches to pseudo-labeling and require that the training parameters of the model be set to optimal values to achieve optimal performance. The problem with these approaches is that there must be confidence in the model to successfully generate pseudo-labels. This means that the model must generate the correct labels, otherwise incorrect labels are generated, the model subsequently learns these self-generated incorrect labels, and predicts incorrectly again due to worse overall performance by iteratively adding noise to itself to the training process. To avoid these problems, [Ber+19] uses a method called MixUp which guesses low-entropy labels for augmenting unlabeled examples in each batch, and then mixes augmented labeled and unlabeled data together by using traditional regularization methods. The augmentation is made by consistency regularization within the unlabeled and labeled data, and afterward the cross entropy loss is used to minimize the loss between the guessed labels and the unlabeled data. FixMatch, a state-of-the-art method, uses a combination of consistency regularization and pseudo-labeling, by requiring that the predictions of strongly augmented data can be paired with the predictions of weakly augmented data to create a labeled sample [Soh+20]. Beside this approach to prevent wrong labels by augmenting and compare the data, which is in this case not possible since the data is un-

known, the wrong label prediction can be avoided by adding something similar to a threshold like in the case of curriculum labeling [Cas+20]. In this approach, self-paced curriculum principles are applied and additionally, to avoid concept drift [Lu+20], the model parameters before each pseudo-labeling cycle are reinitialized from scratch. [Cas+20] uses successful curriculum learning approaches from [Ben+09], where a model is first trained with simple examples and then iteratively progresses to more difficult examples. As described, the difficulty is to create a curriculum that goes not too fast but also not too slow over the initially simple examples. To successfully learn the general features and to iteratively store the knowledge of the weights, a percentile is used. [Ber+22] uses a similar curriculum approaches for pseudo-labeling of NLP tasks by tracking the generalization and overfitting progress. Regarding their findings, especially in fewer data cases, pseudo-labeling is not successful because of overfitting in an early stage of pretraining with labeled data. The recommendation is to start the semi-supervised training very early in the training process by dynamically controlling the pseudo-labels with curriculum to avoid overfitting and stabilize the model.

4.2 Revisiting Self-Training Methods

Several approaches in the semi-supervised self-training domain of pseudo-labeling exist [Ami+22]. However, many are based on a prior spatial augmentation approach, which is relatively easy to implement when images are available [Soh+20][Ber+19][Wen+21][Yan+21]. Since in this showed use case for human hand action recognition in industrial assembly lines, where the features to be used have already been extracted, this is not directly possible at the image level. Furthermore, after the first feature extraction it is necessary to maintain the structure of the hand keypoints, which is why the individual keypoints may not be augmented without further ado. This shows that there is little experience in the described methods that focus on this particular method for an industrial use case and employ it in combination with a transformer encoder model like the ConvGTN from Section 3. [Pha+20], [Xu+21] and [Xia+21] show that it helps to work with separate models on the spatial and temporal part with different focus like it is already in this model structure and seems like a promising approach. But with this teacher student approach one needs to keep track of the confirmation bias which restricts the performance of the student by the teaching performance of the teacher [Ara+19]. Moreover, especially in the approach of this work, no standard pseudo-labeling method can be applied, because due to the many parameters, of the ConvGTN, in this case over 18M., it cannot be assumed that the correct settings of the hyperparameters can be

applied at the training. [Cas+20] and [Ber+22] proved that it helps to converge in the semi-supervised search space faster by using a threshold as well as curriculum and percentile. The focus of the experiments on these methods was based on the described approaches to self-training for fine-grained human hand action recognition in industrial assembly.

5 SELF-TRAINING EXPERIMENTS

Data split for Experiments

For the experiments, a transfer pseudo-labeling approach with curriculum is implemented and compared with several combinations of base and novel data, always under consideration of 7 base and 4 novel class data distribution.² The decision for this partitioning is made to have a strong pretrained model, for weight initialization of the base classes under consideration of just a few examples of novel classes as it is recommended by [Zha+20b]. Besides, it was recognized, if the novel classes are reduced by one class, not enough data is provided for the semi-supervised training and vice versa for a base model pretraining with good performance. Since there are 35 different distribution possibilities for the 7/4 split, five distributions were selected from [Stu+23] depending on the F1 score of the ground truth results of the supervised training on all 11 classes. The selection was based on the best matching classes to these specific industrial conditions, see table 1. The ground truth supervised model training with the new created class `Assembly_Step7` is added as experiment 1 for comparison.

Supervised Pretraining & Weight Transfer

After the data split, a full ConvGTN model from Section 3.2 is supervised pretrained on several combinations of 7 base classes with a seed of 42, a batch size of 6 and a learning rate of $1e-4$. These model weights are transferred supervised to 10% of the 4 novel classes by freezing the whole model with all layers of the encoder but the feedforward of the encoders and the final classifier after the gate, see for comparison Figure 3. The learning rate in the transfer process is set per layer in the Adam optimizer to $1e-3$ in the encoder, $1e-4$ in the classifier and is tracked for minimization

by a `ReduceLROnPlateau`³ scheduler which tracks the learning rate by patience of 7, factor of 0.1 and minimal decay of learning rate of $1e-9$. These new weight initialization of the feed forward layers of the encoder and the classifier helps afterward the curriculum pseudo-labeling approach to converge faster and was trained in 40 epochs by a batch size of 8.

Pseudo-labeling Based Self-Training

The difference in the semi-supervised self-training to the initial setting is that only one of four layers of the encoder of the model is frozen completely. A similar approach as a method that applies higher learning rates to top layers and lower learning rates to bottom layers is used. This procedure took place since existing experiments using similar encoder architectures like BERT [Dev+18] showed that using the complete network for transfer was not always the most effective choice, because transferring the top pre-trained layers can slow down learning and decrease the performance [Zha+20b]. This can be explained as different layers in transformer structures usually capture different kinds of information. Bottom layers often encode more common, general, and broad-based information, while the top layer closer to the output encodes information more localized and specific to the task on hand [Zha+20b]. This procedure is partly accomplished by setting manually the learning rate of the top layer and using a multiplicative decay rate to decrease the learning rate layer-by-layer from top to bottom [HR18]. For these experiments, the learning rate per layer was set to $1e-3$ in the embedding layers, $1e-2$ in the gate, and $1e-3$ in the final classifier, see Figure 3 for comparison of the layers. The encoder part was trained further than the other layers during pre-training and fine-tuning, therefore the learning rate was set to $0.5e-3$. In contrast, to the recommendation to use SGD for semi-supervised learning approaches, Adam optimizer is used to adapt the optimizer to the transformer architecture as it has proven to be a successful optimizer approach in attention models [Zha+19]. Additionally, L2 regularization is used to avoid that the pre-trained target weights deviate too much from the initial weights. For the same reason, the scheduler patience was reduced to 5. After the pre-trained weight transfer to the 10% novel classes, a first initial iteration is used to predict pseudo-labels on the unlabeled dataset. Therefore, a bottom to top approach is used. First, all the unlabeled data is shown to the model with low threshold more precise a percentile, which is computed in a 0.2 step to find widespread patterns over all the data in 150 epochs per iteration.

² During preprocessing, a too similar distribution between `Assembly_Step7` and `Assembly_Step8` was detected, which was caused by the augmentation of the augmented standard dataset and could therefore not be clearly separated from each other. Therefore, `Assembly_Step7` and `Assembly_Step8` were merged to one class, `Assembly_Step7`. The now appearing majority of the class compared to the other classes was compensated afterward by weight balancing and by combining only each second example of both classes to `Assembly_Step7`.

³ https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

After an iteration the predicted labels are added to the dataset, removed from the unlabeled dataset and the model weights are reset to the initial weights from the fine-tuning with again a completely new classifier. The following iterations are done by freezing the convolutional layer and both embedding layers in the first transformer layer, and unfreeze and train all the other layers that are left, see Figure 3 [LTL19]. This procedure is repeated until the percentile reached 100% and the model is confident to predict the right labels to the data. For the final validation, the best model of the last iteration is used because this model had access to most of the data. The complete self-training architecture with curriculum pseudo-labeling can be seen in Figure 4.

5.1 Experimental Setup

The improved ground truth results of the ConvGTN from Section 3.2 are taken as a base for these experiments, especially the test accuracy and F1-Score per class is therefore under deeper consideration. The 7/4 data split was done as shown in Table 1. Additionally,

Exp.	Class Split	7 Base Assembly_Step	4 Novel Assembly_Step
1	0Best & 4Worst	1,2,3,4,5,7,12	6,9,10,11
2	1Best & 3Worst	1,2,3,4,5,7,9	6,10,11,12
3	2Best & 2Worst	2,3,4,5,7,9,10	1,6,11,12
4	3Best & 1Worst	2,3,4,6,7,9,10	1,5,11,12
5	4Best & 0Worst	3,4,6,7,9,10,11	1,2,5,12

Table 1: Dataset Split for Experiments

to the F1-Score comes the observation which combination of base and novel assembly tasks from [Stu+23] works best for the semi-supervised fine-tuning with curriculum learning approach. It is also assumed that not all classes can provide unique features for generalization and lead to good performance of the model when generalizability is considered.

5.2 Model Training Environment

The model architecture was created in PyTorch and stacked on top of Googles framework MediaPipe Hands. The training and hyperparameter tuning was done in Microsoft Azure on a STANDARD_NC6 with 6 vCPUs and 56 GiB Memory. The final model training was done on a GPU which corresponds to half a K80 card with 12 GiB, and a maximum of 24 data disks and 1 NCiS in a duration of 2 hours and 24 minutes up to 4 hours for the pretrained model and 4 hours and 5 minutes up to 6 hours for the self-trained model both depending on the split of the classes.

6 COMPARISON & RESULTS

For a better comparison of the overall results in Table 3, the ground truth F1-Score of each class by training supervised on 100% of the data is provided in Table 2. With this initial supervised training method, the Con-

vGTN reached an overall ground truth test accuracy of 91.18% on the "Industrial Hand Action Dataset V1" [Stu+23]. This result is 5% higher as the test accuracy without the convolutional layer in [Stu+23]. For the following experiments, these target classes of the dataset are split, into 7 base classes for pretraining and 4 novel classes for the downstream tasks. It can also be assumed from previous literature reviews in Section 4 that a strong base model has a positive effect during training on novel models. The split into base and novel classes depends on the results of the F1 score per class from table 2 by dividing the classes into high (best) and low (worst) F1-scores for the novel downstream task. For the exact class split per experiment, see Figure 1. The final results per experiment are presented in table 3 showing each data split per experiment by column and within each column always 3 different training runs with the F1 score results per sub-column. Training on only 10% labeled data is presented in sub-column "10% Sup" for the comparison that semi-supervised training helps. Sub-column "PreTrained + 10% Sup" showing that pre-trained weights help fine-tune to 10% of new data, and the final target, pre-training on baseline data and fine-tuning to 10% of new data as initial weights, followed by 90% curriculum learning in the sub-column "PreTrained + 10% Sup + 90% SemiSup". As additional information in the last row of the table, the overall test accuracy is presented to compare the overall performance over the different combinations of classes depending on best to worst F1-scores of the novel classes after each training experiment. The goal over all experiments is to reach nearly the same F1-Score with the semi-supervised approach, as with 100% labeled data. These results help to see how the pseudo-labeling improves the scores in the experiments. Additionally, only the F1-Scores of the novel classes of each of the experiments are added to the split classes, since only there an improvement helps for making a final result for providing deeper insides into the proof of scalability in deep learning models in industrial environments.

Effect of Pretraining

Compared to the supervised training on only 10% of labeled data, the additional pretraining of the model on the base classes shows as expected in Table 3 an improvement in the overall test accuracy in all variations of the novel classes and experiments. It is also visible, that with a better F1-Score in the novel class, not only the "Test Acc." improves but also the F1-Score in each experiment. Only some small outliers are visible in `Assembly_Step12`, with a reduction from 0.91 to 0.89 in the second experiment and from 1.00 to 0.80 in the fifth experiment, which can in both cases be traced back to bad features in base and novel training. These bad features could lead to negative transfer learning,

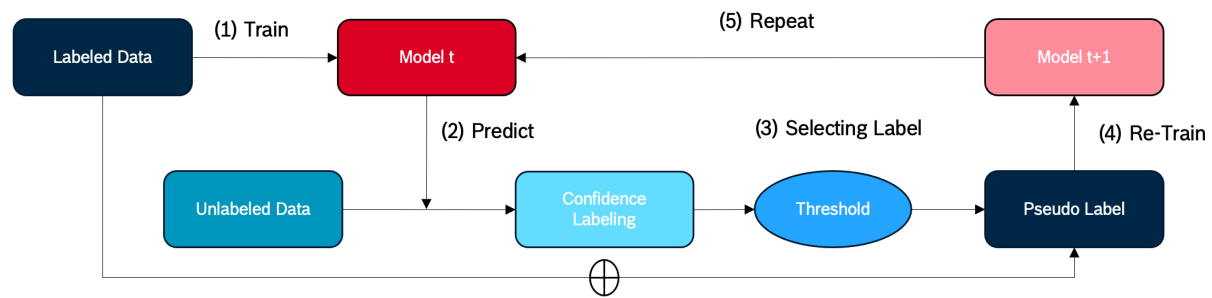


Figure 4: Self-Training Architecture

Assembly_Step	1	2	3	4	5	6	7	9	10	11	12	Total Acc.
F1-Score	0.94	0.93	0.91	0.92	0.93	0.87	0.92	0.91	0.90	0.86	1.00	91.18%

Table 2: Ground Truth Results

like it is experienced beside these experiments in [Wan+18], [PY10] and [Ros+05]. This behavior during model training needs to be examined deeper in future experiments. Nevertheless, the result can be made as expected that robustness of a pre-trained model and amount of data that is used for fine-tuning matters for the overall model performances.

Effect of Pretraining with Curriculum Self-Training

By adding the semi-supervised curriculum learning approach with 90% of unlabeled data, the experiments showed that this procedure gives nearly all models an increase in "Test Acc." by almost reaching the ground truth test acc. over all classes in the last experiment with 86.07% compared to the initial ground truth of 91.18%. The overall test accuracy of different variations of novel classes trained with 90% unlabeled data in the self-training method is always higher than the training on 10% labeled data and adding pretrained model results. Since more data helps in fine-tuning and semi-supervised learning approaches beside the increase of the number of best F1-Score classes in the novel classes. In experiment 5 of the self-training method, the performance increased by 18.32% from 67.75% to 86.07% when compared to experiment 3 with the 4 worst novel classes. But even in these experiments with more data, negative transfer learning is visible. Especially in the cases of experiment 3 and 5 where weak F1-Score classes are added to the novel classes. This is especially the case for class `Assembly_Step6` where the F1-Score dropped from a 0.55 to 0.54 score and from 0.78 to 0.75 score which also affects the final test acc. with 1.4% less. Since this behavior has happened in experiment 5 in 3 out of 4 experiments, the negative transfer learning needs to be further evaluated in the semi-supervised learning approach as well. Additionally, it can not be excluded that the combination of 2 best and 2 worst novel classes leads to bad results in model training because of the possible close relation and similar movements in the actions of the classes.

7 CONCLUSION & OUTLOOK

This approach shows, first, that adding a convolutional layer in the investigation by a spatial tower improves the performance of a spatio-temporal transformer model. Secondly, the main findings in this work are that fine-grained human hand action recognition on a limited amount of novel data can indeed be improved by pre-training of a base model and subsequent usage of a self-training approach based on curriculum labeling to raise the final evaluation results and generalization possibilities of the model. Therefore, it is also important which classes are part of the base model training and how strong and obvious novel classes need to be for the model. Which means having a strong pretrained model helps to improve the results, but also strong novel classes can help if enough data is available. It was also shown that a stable model can be trained even with a small amount of labeled data. This confirms that industrial environments are ideal for scaling deep learning approaches and gives deeper insights for the creation of a pretrained model to prove the scalability in industrial environment. Besides, it shows how the fine-tuning in transformer models needs to happen by freezing only specific layers of the transformer architecture but also use different learning rates per layers. In addition, compared to the traditional methods of using an SGD optimizer in semi-supervised training, Adam was used. Since a lot of human fine-grained hand actions look similar, negative transfer learning was experienced probably because of the similar movements from base to novel classes which will be further evaluated in the following experiments, by investigating approaches to prevent negative learning and to improve the generalizability of the model, by self-attention approaches for the recognition of fine-grained human hand actions in industrial assembly.

REFERENCES

[Liu+21] Minghao Liu, Shengqi Ren, Siyuan Ma, Jiahui Jiao, Yizhou Chen, Zhiguang

	10% Sup PreTrained + 10% Sup PreTrained + 10% Sup + 90% SemiSup				
Class\Experiment	0Best & 4Worst	1Best & 3Worst	2Best & 2Worst	3Best & 1Worst	4Best & 0Worst
Assembly_Step1			0.22 0.67 0.74	0.40 0.74 0.69	0.43 0.57 0.85
Assembly_Step2					0.42 0.70 0.88
Assembly_Step5				0.31 0.50 0.88	0.22 0.70 0.81
Assembly_Step6	0.25 0.55 0.54	0.44 0.57 0.69	0.52 0.78 0.75		
Assembly_Step9	0.67 0.80 0.76				
Assembly_Step10	0.46 0.53 0.59	0.50 0.80 0.72			
Assembly_Step11	0.30 0.47 0.60	0.53 0.70 0.73	0.57 0.74 0.71	0.14 0.71 0.71	
Assembly_Step12		0.91 0.89 0.95	0.83 0.91 0.89	0.50 1.00 0.85	1.00 0.80 0.92
Test Acc. in %	40.54 59.46 63.82	56.67 72.41 74.81	54.84 76.67 75.27	32.41 71.43 77.22	41.94 67.75 86.07

Table 3: Experimental Scores per Class on Pretrained Model on 7 Base Classes + Semi-Supervised 4 Novel Classes

- [Xu+15] Wang, and Wei Song. “Gated Transformer Networks for Multivariate Time Series Classification”. In: *CoRR* abs/2103.14438 (2021). arXiv: 2103 . 14438. URL: <https://arxiv.org/abs/2103.14438>.
- [Stu+23] Fabian Sturm, Elke Hergenroether, Julian Reinhardt, Petar Smilevski Vojnovikj, and Melanie Siegel. *Challenges of the Creation of a Dataset for Vision Based Human Hand Action Recognition in Industrial Assembly*. 2023. DOI: 10 . 48550 / ARXIV . 2303 . 03716. URL: <https://arxiv.org/abs/2303.03716>.
- [GBV20] Margherita Grandini, Enrico Bagli, and Giorgio Visani. *Metrics for Multi-Class Classification: an Overview*. 2020. DOI: 10 . 48550 / ARXIV . 2008 . 05756. URL: <https://arxiv.org/abs/2008.05756>.
- [Zha+20a] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. “MediaPipe Hands: On-device Real-time Hand Tracking”. In: *CoRR* abs/2006.10214 (2020). arXiv: 2006 . 10214. URL: <https://arxiv.org/abs/2006.10214>.
- [Lin+16] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. “Feature Pyramid Networks for Object Detection”. In: *CoRR* abs/1612.03144 (2016). arXiv: 1612 . 03144. URL: <http://arxiv.org/abs/1612.03144>.
- [Lin+17] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. “Focal Loss for Dense Object Detection”. In: *CoRR* abs/1708.02002 (2017). arXiv: 1708 . 02002. URL: <http://arxiv.org/abs/1708.02002>.
- [Xu+15] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. “Empirical Evaluation of Rectified Activations in Convolutional Network”. In: *CoRR* abs/1505.00853 (2015). arXiv: 1505 . 00853. URL: <http://arxiv.org/abs/1505.00853>.
- [Far+21] Abolfazl Farahani, Behrouz Pourshojae, Khaled Rasheed, and Hamid R. Arabnia. “A Concise Review of Transfer Learning”. In: *CoRR* abs/2104.02144 (2021). arXiv: 2104 . 02144. URL: <https://arxiv.org/abs/2104.02144>.
- [GB04] Yves Grandvalet and Yoshua Bengio. “Semi-supervised Learning by Entropy Minimization”. In: *Advances in Neural Information Processing Systems*. Ed. by L. Saul, Y. Weiss, and L. Bottou. Vol. 17. MIT Press, 2004. URL: <https://proceedings.neurips.cc/paper/2004/file/96f2b50b5d3613adf9c27049b2a888c7-Paper.pdf>.
- [Zho+18] Hong-Yu Zhou, Avital Oliver, Jianxin Wu, and Yefeng Zheng. “When Semi-Supervised Learning Meets Transfer Learning: Training Strategies, Models and Datasets”. In: *CoRR* abs/1812.05313 (2018). arXiv: 1812 . 05313. URL: <http://arxiv.org/abs/1812.05313>.
- [Yan+21] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. “A Survey on Deep Semi-supervised Learning”. In: *CoRR* abs/2103.00550 (2021). arXiv: 2103 . 00550. URL: <https://arxiv.org/abs/2103.00550>.
- [Wen+21] Zejia Weng, Xitong Yang, Ang Li, Zuxuan Wu, and Yu-Gang Jiang. “Semi-Supervised Vision Transformers”. In: *CoRR* abs/2111.11067 (2021). arXiv: 2111 . 11067. URL: <https://arxiv.org/abs/2111.11067>.

- [Xu+21] Yinghao Xu, Fangyun Wei, Xiao Sun, Ceyuan Yang, Yujun Shen, Bo Dai, Bolei Zhou, and Stephen Lin. “Cross-Model Pseudo-Labeling for Semi-Supervised Action Recognition”. In: *CoRR* abs/2112.09690 (2021). arXiv: 2112.09690. URL: <https://arxiv.org/abs/2112.09690>.
- [ITP14] Alexandros Iosifidis, Anastasios Tefas, and Ioannis Pitas. “Semi-supervised Classification of Human Actions Based on Neural Networks”. In: *2014 22nd International Conference on Pattern Recognition*. 2014, pp. 1336–1341. DOI: 10.1109/ICPR.2014.239.
- [Xia+21] Junfei Xiao, Longlong Jing, Lin Zhang, Ju He, Qi She, Zongwei Zhou, Alan L. Yuille, and Yingwei Li. “Learning from Temporal Gradient for Semi-supervised Action Recognition”. In: *CoRR* abs/2111.13241 (2021). arXiv: 2111.13241. URL: <https://arxiv.org/abs/2111.13241>.
- [Ami+22] Massih-Reza Amini, Vasilii Feofanov, Loic Pauletto, Emilie Devijver, and Yuri Maximov. *Self-Training: A Survey*. 2022. DOI: 10.48550/ARXIV.2202.12040. URL: <https://arxiv.org/abs/2202.12040>.
- [Lee13] Dong-Hyun Lee. “Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks”. In: *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)* (July 2013).
- [Pha+20] Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V. Le. “Meta Pseudo Labels”. In: *CoRR* abs/2003.10580 (2020). arXiv: 2003.10580. URL: <https://arxiv.org/abs/2003.10580>.
- [Ber+19] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. “MixMatch: A Holistic Approach to Semi-Supervised Learning”. In: *CoRR* abs/1905.02249 (2019). arXiv: 1905.02249. URL: <http://arxiv.org/abs/1905.02249>.
- [Soh+20] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. “FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence”. In: *CoRR* abs/2001.07685 (2020). arXiv: 2001.07685. URL: <https://arxiv.org/abs/2001.07685>.
- [Cas+20] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordóñez. “Curriculum Labeling: Self-paced Pseudo-Labeling for Semi-Supervised Learning”. In: *CoRR* abs/2001.06001 (2020). arXiv: 2001.06001. URL: <https://arxiv.org/abs/2001.06001>.
- [Lu+20] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. “Learning under Concept Drift: A Review”. In: *CoRR* abs/2004.05785 (2020). arXiv: 2004.05785. URL: <https://arxiv.org/abs/2004.05785>.
- [Ben+09] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. “Curriculum learning”. In: *International Conference on Machine Learning*. 2009.
- [Ber+22] Dan Berrebbi, Ronan Collobert, Samy Bengio, Navdeep Jaitly, and Tatiana Likhomanenko. *Continuous Pseudo-Labeling from the Start*. 2022. DOI: 10.48550/ARXIV.2210.08711. URL: <https://arxiv.org/abs/2210.08711>.
- [Ara+19] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O’Connor, and Kevin McGuinness. “Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning”. In: *CoRR* abs/1908.02983 (2019). arXiv: 1908.02983. URL: <http://arxiv.org/abs/1908.02983>.
- [Zha+20b] Tianyi Zhang, Felix Wu, Arzo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. *Revisiting Few-sample BERT Fine-tuning*. 2020. DOI: 10.48550/ARXIV.2006.05987. URL: <https://arxiv.org/abs/2006.05987>.
- [Dev+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: <http://arxiv.org/abs/1810.04805>.
- [HR18] Jeremy Howard and Sebastian Ruder. “Fine-tuned Language Models for Text Classification”. In: *CoRR* abs/1801.06146 (2018). arXiv: 1801.06146. URL: <http://arxiv.org/abs/1801.06146>.

- [Zha+19] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. *Why are Adaptive Methods Good for Attention Models?* 2019. DOI: 10.48550/ARXIV.1912.03194. URL: <https://arxiv.org/abs/1912.03194>.
- [LTL19] Jaejun Lee, Raphael Tang, and Jimmy Lin. “What Would Elsa Do? Freezing Layers During Transformer Fine-Tuning”. In: *CoRR* abs/1911.03090 (2019). arXiv: 1911.03090. URL: <http://arxiv.org/abs/1911.03090>.
- [Wan+18] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime G. Carbonell. “Characterizing and Avoiding Negative Transfer”. In: *CoRR* abs/1811.09751 (2018). arXiv: 1811.09751. URL: <http://arxiv.org/abs/1811.09751>.
- [PY10] S.J. Pan and Q. Yang. “A Survey on Transfer Learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359.
- [Ros+05] Michael Rosenstein, Zvika Marx, Leslie Kaelbling, and Thomas Dietterich. “To Transfer or Not To Transfer”. In: *NIPS 2005*. Jan. 2005.

Position Based Rigid Body Simulation: A comparison of physics simulators for games

Miguel Seabra^{1,2}

Francisco Fernandes²

Daniel Simões
Lopes^{1,2}

João Madeiras
Pereira^{1,2}

¹ IST-UTL
Av. Rovisco Pais
1049-001, Lisbon
Portugal

² INESC-ID
Rua Alves Redol, 9
1000-029, Lisbon
Portugal

{ miguel.l.n.seabra, francisco.fernandes, daniel.s.lopes, joao.madeiras.pereira }@tecnico.ulisboa.pt

ABSTRACT

Interactive real-time rigid body simulation is a crucial tool in any modern game engine or 3D authoring tool. The quest for fast, robust and accurate simulations is ever evolving. PBRBD (Position Based Rigid Body Dynamics), a recent expansion of PBD (Position Based Dynamics), is a novel approach for this issue. This work aims at providing a comprehensible state-of-the-art comparison between Position Based methods and other real-time simulation methods used for rigid body dynamics using a custom 3D physics engine for benchmarking and comparing PBRBD (Position Based Rigid Body Dynamics), and some variants, with state-of-the-art simulators commonly used in the gaming industry, PhysX and Havok. Showing that PBRBD can produce simulations that are accurate and stable, excelling at maintaining consistent energy levels, and allowing for a variety of constraints, but is limited in its handling of stable stacks of rigid bodies due to the propagation of rotational error.

Keywords

Position Based Rigid Body Dynamics, PBRBD, Real-time Physics Simulation, Benchmark.

1 INTRODUCTION

Physical Simulation is a wide field within computer graphics and animation, being crucial for modern animation effects and interactive simulations such as those found in video games. The demand for reliable physical simulation has grown with the popularization of virtual reality, since users interact with everyday objects within simulated environments in more ways than ever before. Physical simulators are usually measured along three metrics: robustness, accuracy and time efficiency. Scientific simulations usually trade efficiency for accuracy, but for interactive applications, prioritizing fast solutions is key. Rather than true physical accuracy, real time simulations just need enough to maintain visual plausibility. Most of the time this means finding approximate solutions as fast as possible in a robust way.

A large body of works exists on how to speed up simulations and finding solutions for a vast range of

physical phenomena and materials [BET14, BMM17, NMK⁺06]. PBD (Position Based Dynamics) is one of these methods, and it stands out from the crowd due to its robustness, visual accuracy, and speed [BMM17], as well as its ability to handle over and under constrained environments gracefully [MMC⁺20]. Its most common applications are the simulation of particle systems which are unable to efficiently simulate rigid bodies. Due to this limitation particle systems, and rigid bodies are often simulated in different physics engines meaning that interaction between both will be heavily limited.

Recently, PBD was expanded upon, creating Position Based Rigid Body Dynamics (PBRBD) which allows rigid bodies and particles to coexist and interact implicitly while remaining fast, robust and stable. This method was first shown alongside a collection of demos showcasing its capabilities and strengths. These demos however did not contain any comparison to other simulators available [MMC⁺20]. Making it a challenge to determine how this method compares in terms of accuracy, speed and robustness to any current methods.

The work will focus on exploring the limits and benchmarking and comparing PBRBD to other methods for rigid body simulation helping future researchers and industry professionals to assess whether they should implement or perform further testing using PBRBD.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

In order to benchmark PBRBD a package was developed containing an implementation of the physics engine that can be used within the unity game engine. Benchmarking software was also developed that is capable of creating equivalent scenarios across three different independent physics engines, the custom PBRBD implementation, Unity's default physics system PhysX and Havok. This software also acquires benchmarking data for the three engines.

2 RELATED WORK

Rigid body dynamics simulation methods fall under three general categories, force, velocity and position based methods[BET14]. Force methods solve collisions using virtual springs to enact forces on bodies and maintain any constraints. Since virtual springs act as natural forces some consider these methods to be more realistic.[BML⁺14]. The most common types of force based method are penalty methods. When a collision is detected a spring is created at the contact point that pushes the bodies into non-colliding positions [MW88]. To simulate friction a spring is created between the contact point that opposes tangential movement [XZB14]. Common issues with this methodology are that resting contacts may suffer from oscillations from the springs [Dru07], and collisions between fast moving or heavy bodies require strong springs to separate them which can lead to numeric instability [Dru07].

Velocity based methods solve collisions by changing velocities directly, Impulse methods do so via the application of impulses which are instantaneous accelerations acting during a single instant [TBV12, MC95]. These methods might require resting contacts to be handled differently as the impulses used for separating objects can lead to jittering.

Position based methods were originally used for particle systems. These methods work on positions directly projecting objects currently colliding into the nearest collision free position using a Gauss-Seidel step to iterate and solve all collisions and constraints. The original Position Based Dynamics method [MHHR07], although fast, robust and simple, had shortcomings that made it harder to work with. The stiffness of constraints was time step dependant. This made arriving at a suitable stiffness parameter and substep count a challenge. The algorithm also had no direct correspondence to real world elastic and dissipation energy potentials [MMC16, BML⁺14] making it hard to simulate real world scenarios. The original method's shortcomings were eventually solved by XPBD (Extended Position Based Dynamics) [MMC16]. Currently, the term PBD is usually interchangeable with XPBD. The extension managed to decouple stiffness from substep count by replacing the concept with compliance, the inverse of stiffness. It also made the method more ro-

bust at handling hard constraints, since they were essentially infinitely stiff. Using compliance, a hard constraint has a compliance of value one, and a fully compliant constraint has a value of zero [MMC16]. It was also extended to receive physical quantities and introduced Lagrange multipliers to its equations, which offers the previously mentioned constraints a force estimate value. This method still had some shortcomings, it is derived from an implicit time stepping scheme and as such suffers from energy dissipation [MMC16]. It has also been stated that Gauss-Seidel solvers can oscillate between solutions rather than converge given non-feasible sets [BML⁺14]. The most recent iteration of PBD extends the method to handle rigid body dynamics. PBRBD (Position Based Rigid Body Dynamics) adds steps to XPBD's algorithm in order to handle orientation and angular velocity, as well as adding angular constraints [MMC⁺20]. It has also been shown that for position based methods it is more efficient and accurate to break down the temporal window of each simulation step and simulate more steps per second handling, even at the expense of only performing one Gauss-Seidel iteration[MSL⁺19].

3 IMPLEMENTATION

3.1 Environment

In order to test and benchmark our approach, we needed to have a development environment providing physics engines to serve as comparison as well as offering rendering, profiling and debugging tools. As such, Unity's game engine using C# scripts to run the simulation was deemed the better option. Since PBRBD requires double precision floating points and most math capabilities provided by the engine only support single precision floating points none of Unity's math classes containing vectors, quaternions and matrixes could be used creating a fully independent package that handles all the simulation and then simply updates the positions Unity uses.

3.2 Simulation loop

Algorithm 1 Simulation Step

```
BroadCollisionDetection()
 $h \leftarrow \Delta time / numSubsteps$ 
for  $numSubsteps$  do
    PositionalUpdate()
    ConstraintSolve()
    VelocityUpdate()
    NarrowCollisionDetection()
    VelocitySolve()
end for
UpdateEnginePositions()
```

The implemented simulation loop is executed once per frame and simulates as many substeps as *numSubsteps*. Increasing this value decreases the size of the temporal window, which has been shown to be the most efficient way of increasing accuracy [MSL⁺19]. The rigid bodies and particles used by the simulation are independent entities from the ones used by the game engine for rendering. As such, one final step is necessary to update the positions and orientations of the simulated bodies.

3.3 Bodies

Particles are defined as a mass, position and velocity. A rigid body on the other hand, also has orientation defined as a quaternion, angular velocity and external torque parameters defined as vectors. In order to properly apply realistic rotations, rigid bodies also require an inertia tensor which refers to mass in rotational terms [MMC⁺20], a 3x3 matrix that contains information regarding the moment of inertia of a rotation along the bodies' principle axes. Since shape and consequent mass distribution of a body has an impact on its rotation. The Inertia tensor is dependent on the orientation of the body, to avoid recalculation the tensor is always defined in local coordinates and any rotation that is applied to the body need to be converted to self coordinates, multiplied by the tensor and converted back to world coordinates.

3.4 Positional Update

The first step within the algorithm's internal loop performs the time integration of the current positions and velocities according to the current velocity and acceleration. During this step the previous position and orientations are updated. The new position and velocity of a body is updated by applying one Euler step [MSL⁺19]:

$$\begin{aligned}x &= x + \vec{V} \cdot h \\ \vec{V} &= \vec{V} + \vec{F} \cdot h\end{aligned}\quad (1)$$

Where x , \vec{V} and \vec{F} are the position, velocity and external force vectors respectively, and h is the time interval being simulated. The above update is sufficient for simulating particles, for rigid bodies the following steps are also required [MMC⁺20]:

$$\begin{aligned}W_Q &= [0, \vec{W} \cdot x, \vec{W} \cdot y, \vec{W} \cdot z] \\ Q &= Q + 0.5 \cdot W_Q \cdot Q \cdot h \\ Q &= |Q|\end{aligned}\quad (2)$$

$$\vec{W} = \vec{W} + h \cdot I^{-1} \cdot (\vec{T} - (\vec{W} \times (I \cdot \vec{W}))) \quad (3)$$

Where q and \vec{W} and \vec{T} are the orientation quaternion, angular velocity in self coordinates and external torque vectors respectively, I represents the inertia tensor. Angular velocity is converted into a quaternion and transformed into world coordinates and scaled by the time h .

Note that since the angular velocity \vec{W} is stored in self coordinates, there is no need to perform any coordinate conversion before applying the tensor.

3.5 Constraints

It is possible to create constraints that simulate a variety of physical effects, the Nonlinear Gauss-Seidel solver is capable of processing different types of constraints since all are solved in the same generalized manner, with positional and angular constraints requiring slightly different approaches.

Constraints vary in how the error and its gradient ΔC is calculated. The gradient is a vector that points in the direction with most impact to the error value and with magnitude proportional to the impact moving the object will have on the error value calculated by $C(x)$. The Lagrange multiplier used to solve the constraint by calculating the positional correction Δx is calculated using the error value and the inverse mass values w_i of bodies affected by it [MMC16].

$$\Delta x = \lambda w_i \Delta C \quad (4)$$

$$\lambda = \frac{C(x)}{\sum w_i \Delta C_i^2 + \alpha / h^2} \quad (5)$$

Constraints use the value of inverse mass to distribute the correction between constrained bodies, as seen in 5 and 4. In practice, a body with twice the mass will suffer half the effect of the constraint's correction, while the lighter body will experience double. Using inverse mass is useful for having infinitely heavy objects that cannot be moved by any correction simply by setting its value to zero. Compliance α determines how rigid a constraint should act. A compliance value of zero corresponds to a rigid constraint where error is fully corrected. More compliant constraints only correct a fraction of the error, leading to a spring like behaviour.

For positional constraints, the Lagrange multiplier calculations are as shown in 5 and applied as in 4. Some positional constraints might not act on the center of mass of the body, in those cases a vector \vec{R} determines the offset from the acted on position and centre of mass, in self coordinates. Furthermore, the movement needs to impact the rotation of the body Q . This is achieved via an extra step, correcting orientation [MMC⁺20]:

$$\begin{aligned}\Delta X &= Q^{-1} \cdot \Delta X \\ rotation &= Q \cdot (I^{-1} \cdot \vec{R} \times \Delta X) \\ Q &= Q + 0.5 \cdot rotation \cdot Q \\ Q &= |Q|\end{aligned}\quad (6)$$

In order to keep energy conservation when transferring positional kinetic energy to rotational kinetic energy, a

different value for inverse mass is used called the generalized inverse mass w_i [MMC⁺20].

$$\begin{aligned} rotation &= \vec{R} \times (Q^{-1} \cdot \Delta X) \\ w_i &= w_i + (rotation \cdot I^{-1}) \cdot rotation \end{aligned} \quad (7)$$

Distance constraints take two bodies, and offsets from the centre of mass, and ensure that the distance between the positions with offset applied are within a certain range. The error of a distance constraint is simply the difference between the real distance between the points and the desired distance. The gradient points in the direction opposite to the other particle. And the magnitude of the gradient has a magnitude of one.

Angular constraints are solved similarly to positional ones. Instead of using inverse mass, the inverse inertia tensor, the rotational equivalent of mass, is used. Corrections come in the form of a vector, which can be broken down into length θ and direction ΔW_n . The new generalized inverse mass used for calculating the Lagrange multiplier is calculated as [MMC⁺20]:

$$\begin{aligned} W_{self} &= Q^{-1} \cdot \Delta W_n \\ w_i &= W_{self}^T I^{-1} W_{self} \end{aligned} \quad (8)$$

The correction of the orientations, with respect to the Lagrange multiplier λ is done as follows:

$$\begin{aligned} P_Q &= [P.x, P.y, P.z, 0] \\ P &= \Delta W_n \cdot \lambda \\ P_{self} &= Q^{-1} \cdot P \\ P &= Q \cdot (I^{-1} \cdot P_{self}) \\ P &= Q \pm 0.5 \cdot P_q \cdot Q \end{aligned} \quad (9)$$

Angular constraints are applied in relation to some axis, defined in the body's self coordinates, labelled a_n . In certain cases, a secondary axis is needed, which takes the form of b_n . A hinge joint works by ensuring that two axes belonging to two bodies remain aligned. The gradient and error of this constraint can be calculated as [MMC⁺20]:

$$\begin{aligned} \Delta C &= \frac{a_{1world} \times a_{2world}}{|a_{1world} \times a_{2world}|} \\ error &= |a_{1world} \times a_{2world}| \end{aligned} \quad (10)$$

Ball joints work by limiting the angle between two axes to be in a certain interval. If the angle between two axes (σ) exceeds the max bound (α) an error is calculated and returned. The gradient is the same as a hinge joint, and the error of the constraint is calculated as:

$$error = |a_{1world} \times a_{2world}| \cdot (\sigma - \alpha) \quad (11)$$

3.6 Collisions

In PBD collisions are handled as constraints, when a collision is found a new constraint is initialized and corrected immediately. The gradient of this constraint is

the vector that can separate the penetrating colliders in the shortest distance coinciding with the contact normal multiplied by the penetration depth at the contact point. To simulate correct restitution and friction, a special step, called the velocity solve, is needed, iterating through collisions and adjusting velocities. When a collision is detected, a collision data structure containing references to both colliders, collision point (p), penetration distance (d), and contact normal (\vec{N}) is created.

3.7 Restitution

To achieve physically accurate conservation of momentum, the velocities resulting from a collision need to be adjusted during the velocity solve step, while taking into account bodies' restitution coefficients e_n . A body with a restitution coefficient of zero absorbs all the energy from a collision impulse, while one with a value of one absorbs no energy.

This step handles collision instances, that have information on both colliders, a collision normal \vec{N} , and the offsets from the colliders' centre of mass and collision point R_n , defined in self coordinates. The velocity solve step begins by calculating the difference between both velocities $\Delta \vec{V}$, the velocity normals \vec{V}_n and tangential velocities \vec{V}_t [MMC⁺20].

$$\Delta \vec{V} = N(-\vec{V}_n + \min(-(e_1 e_2) \vec{V}_t, 0)) \quad (12)$$

This step consist of subtracting the current velocity and replacing it with a reflected velocity \vec{V}_t [MMC⁺20]. The resulting correction to velocity $\Delta \vec{V}$ now needs to be distributed by both bodies according to their masses and distributed in terms of positional and rotational energy. This following step is used whenever a velocity is corrected within the velocity solve [MMC⁺20]:

$$\begin{aligned} P &= \Delta \vec{V} / (w_1 + w_2) \\ \vec{V}_1 &= \vec{V}_1 + P / m_1 \\ \vec{V}_2 &= \vec{V}_2 - P / m_2 \\ W_1 &= W_1 + I^{-1} (\vec{R} \times P) \\ W_2 &= W_2 - I^{-1} (\vec{R} \times P) \end{aligned} \quad (13)$$

3.8 Friction

Friction is a dissipative force that opposes movement between two tangential surfaces. The strength of this force is determined by the amount of force the bodies are exerting on each other (usually the normal force) and the friction coefficients, values referring to the amount of friction produced by the body's material. There are two types of friction, one that acts when initiating motion (static friction) and another that acts after movement is initiated (dynamic friction).

3.9 Static Friction

Static friction is implemented using a positional constraint initialized after separating the contact. It takes the sliding bodies' positions x_n , the offsets from centre of mass to collision points \vec{R}_n and the collision normal used for calculating the collision tangent direction. It then ensures that no tangential movement occurs between the contact points. The force exerted by a constraint can be calculated as [MMC16]:

$$\begin{aligned}\vec{F} &= \lambda \vec{N} / h^2 \\ \tau &= \lambda \vec{N} / h^2\end{aligned}\quad (14)$$

The formula that determines if static friction is applied in respect to the static friction coefficient μ_s and the normal and friction forces is:

$$\vec{F}_{static} \leq \mu_s \vec{F}_{normal} \quad (15)$$

The values of \vec{F}_{static} and \vec{F}_{normal} are proportional to λ_{static} and λ_{normal} respectively, as such the formula above can be implemented as follows:

$$\lambda_{static} \leq \mu_s \lambda_{normal} \quad (16)$$

If the above condition is not met then the constraint responsible for applying static friction is discarded before applying corrections to any bodies and marks the collision for dynamic friction to be applied during the velocity Solve step.

3.10 Dynamic Friction

During the velocity solve step, collisions that have not experienced static friction have dynamic friction applied. The force exerted cannot exceed a certain value, determined by the dynamic friction coefficient and normal force. Using (14) and (1) it is possible to calculate the velocity the dynamic force produces during the current time step using the following formula:

$$\Delta \vec{V} = \left| \frac{\mu_{dynamic} \cdot \lambda_{dynamic}}{h} \right| \quad (17)$$

3.11 Interaction

For behaviours that require input from the user such as using the mouse to drag bodies it is necessary to take into account that the input devices are updated at a lesser frequency than the simulations substeps, as such mouse positions need to be collected and interpolated by each substep to simulate continuous movement of the anchor point.

3.12 Soft Bodies

Soft bodies are simulated as a series of particles connected by constraints, a mesh is then dynamically altered so that it matches the particles' positions

[MHTG05]. Particles are connected via distance and also volume constraints which take four particles and ensure that the tetrahedron formed by their points' volume remains constant. The direction of the gradient is different for all particles, but it is always perpendicular to the plane defined by the three other particles.

3.13 Jacobi Solver

In order to compare the difference between using a Jacobi and the default Gauss-Seidel solver, both solvers were implemented with an option to toggle between them. When using the Jacobi solver corrections are stored, once a substep is finished all the corrections are averaged and then applied to the body.

3.14 Optimizations

Collision detection can be broken down into the broad and narrow phases. The broad phase takes all colliders and tries to identify which pairs of collisions can possibly occur during the next simulation loop, and is executed once per step. The narrow phase iterates through the likely collisions and checks for actual contacts, and is executed once per substep. The collision detection and constraint resolution steps are implemented in parallel, distributing the workload amongst different threads. For the Gauss Seidel solver, mutual exclusion blocks are used to ensure that no constraint that shares a body with another is processed in parallel.

4 EVALUATION AND RESULTS

4.1 Testing Methodology

In order to properly benchmark and compare PBRBD's custom implementation, which will be referred to simply as PBD, with other prominent physics engines available in Unity, the default implementation using PhysX which will be referred simply as Unity and Havok's physics package, as well as a version of PBD using a Jacobi solver labelled as Jacobi and parallel configurations of both versions. A variety of topics of interest were selected, testing the engine's along an array of different attributes. Each topic is then studied by proxy of simulation scenarios designed to expose each engine's performance in each topic using instrumentalized versions of the algorithms and data collectors. All tests were conducted on a Lenovo Legion 5 laptop, using an AMD Ryzen 7 5800H processor, 16GB of RAM, and a GeForce RTX 3060 GPU.

4.2 Momentum conservation in collisions

Conservation of momentum dictates how real world object's velocities are affected by a collision, ensuring that no energy is gained or lost from the collision's impulses. A commonly used mechanism for

demonstrating this phenomenon is the Newton's cradle, a device comprising, usually, of four spheres of equal mass suspended by wires, which, excluding dissipating forces, can remain in perpetual motion. Simulating the contraptions using distance constraints to simulate the strings as in Fig. 1a shows that PBD provides the closest simulation to theoretical results as shown in Fig. 1b.

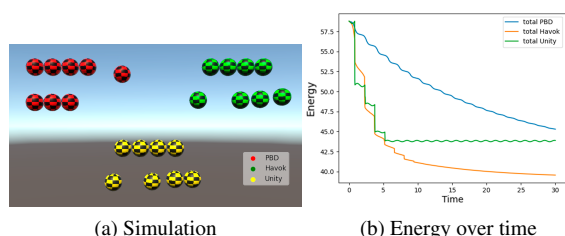


Figure 1: Total energy values of Newton's cradle using constraints to simulate string over 30 seconds. PBD shows a stable energy loss, while Havok and Unity show sudden drops that coincide with the times collisions happen during the simulation. The reason their energy levels stabilize is that wrongful momentum conservation is causing the spheres at the centre to swing, the simulation converges to a state where all four spheres move in tandem.

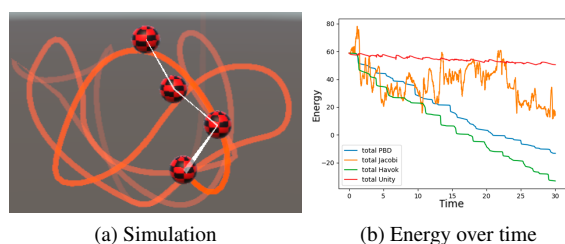


Figure 2: Energy evolution over a 10-second simulation of a triple pendulum using distance constraints. Unity provides greater energy conservation, however it does suffer from some energy being gained which might compromise its simulation. PBD and Havok exclusively lose energy and show similar results. In comparison, Jacobi is very unstable, suggesting the solver is less accurate. Only PBD's simulation was shown in the visual example for clarity.

4.3 Energy conservation in constraints

On the above section, the analysed energy losses were, mostly, a by-product of several collisions in a short time frame. While the triple pendulum (Fig. 2a) is a suitable test case to for the accuracy and stability of distance constraints, it does not test the impact that its corrections had on orientation. In order to test a scenario where proper simulation of the orientation of bodies is crucial, a chain was simulated by attaching the edges of capsules together. At the end of the chain, a heavy

sphere is attached (Fig. 3) in order to weigh the chain down. According to Fig. 4 PBD offers the best results for simulations of 100 capsules, displaying negligible energy gains and acceptable losses. However, when simulating 500 capsules, the simulation breaks and massive amounts of energy are gained. This most likely has to do with the velocity recalculation step, after a large correction was applied to a particle its velocity was recalculated as a massive value leading to unexpected and uncontrollable behaviour.

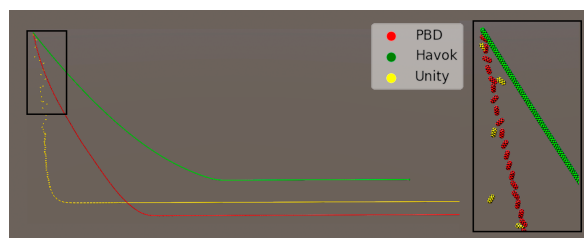


Figure 3: The simulation of a chain made up of 500 capsules connected by constraints. Both PBD and Unity exhibiting wrongful simulation towards the top of the rope, with the latter showing the most error.

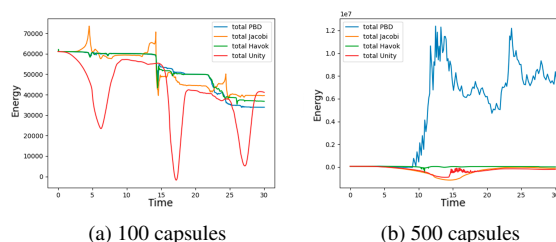


Figure 4: Energy conservation when simulating a chain comprised of a varying number of capsules connected by constraints. All engines used 20 iteration or sub-steps. While PBD is shown to be the most stable choice for 100 capsules, when a scenario is too complex for the current substep count it can diverge. Jacobi which has been shown to be less accurate in Fig. 2b is shown here to be more robust.

4.4 Stability

Stacks of bodies are challenging simulations because error can quickly propagate and bring, what should in theory be a stable stack, to collapse in on itself. To test the stability of a stack, the energy conservation will be analysed as well as how much the body at the top of a stack, which should be stationary, moves (Fig. 5) in order to capture positional error in the form of drift, a known problem with velocity methods [MMC16]. The movement of the top body can be analysed via the difference between the bodies' starting and current positions throughout the simulation. In order to distinguish between error concerning the separation of penetrating bodies and drift, horizontal and vertical deviation from

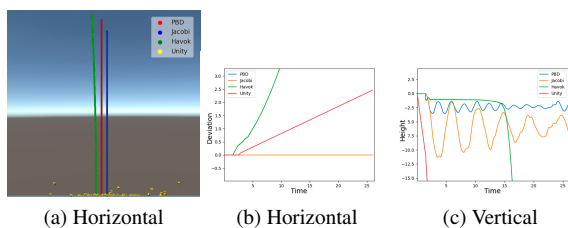


Figure 5: The horizontal and vertical deviation from the top cube's current and starting position over time in a vertical stack. Havok's and Unity's simulation eventually collapses, seen by the large vertical deviation. PBD and Jacobi show no drift, but they both oscillate vertically as they struggle to keep the cubes from interpenetration, at worse, this could result in cubes gaining vertical velocity and being thrown upwards.

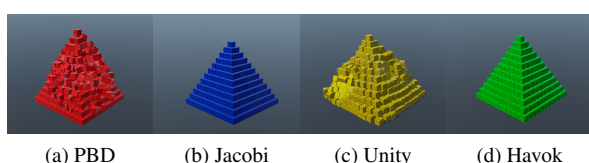


Figure 6: The simulation of 650 cubes organized in a pyramid shape. PBD and Unity's simulation show wrongful behaviour leading to the eventual collapse of the structure, the former's error comes from rotations within the bodies while the latter's is due to drifting.

the starting position is analysed separately. The most basic example of the type of system mentioned above would be a single vertical stack of cubes (Fig. 5a), where the forces experienced by cubes near the bottom of the stack would be immense due to having to support hundreds of other cubes on top of them. There are many ways positional error can impact said structure. The stack may become compressed as the system is unable to keep bodies from interpenetrating, efforts to separate bottom cubes may increase penetration towards the top which may end up with cubes being sent upwards instead of resting, PBD makes no distinction between resting contacts and collisions making it more susceptible to interpenetration.

A more complex type of stack would be to organize the cubes in a pyramid shape (Fig. 6). PBD suffers from rotation being applied to cubes, this happens because each cube has four contacts beneath it, using the Gauss-Seidel solver means that each contact is solved and corrected one at a time, when the first contact is handles, towards one of the bottom corners of the object it applies a rotation as well as a translation, this rotation will cause further penetration in the opposite corner which eventually leads to an overcorrection and destabilizes the stack. When using Jacobi the corrections are stored and averaged together and applied all at once, meaning that the rotations will cancel each other out leading to a more stable stack. Jacobi also suffers from some verti-



Figure 7: Cuboids with uneven axes lengths

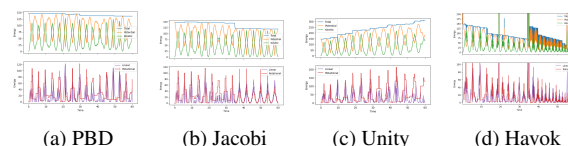


Figure 8: The energy of the systems throughout a 30-second simulation of a plane cuboid.

cal oscillations, similar to what was shown in the previous example. Neither method achieves a stable stack at every simulation, but Jacobi has a higher rate of success. Unity's simulation is vertically and rotationally stable but suffers from horizontal drift, when the cubes are too spread out the structure collapses. Finally, Havok provides a stable simulation and could still provide stable results with more than five times as many elements being the strictly better choice.

4.5 Linear and Rotational Kinetic Energy

When calculating the energy values of a system, two separate values make up the total energy value, potential energy is dependent on the height of an object, and kinetic energy on its velocity. Kinetic energy has two components as well, linear kinetic energy and rotational kinetic energy. Following a collision, a falling object might transform potential energy into kinetic energy, and that energy needs to be properly distributed by its linear and rotational components. This distribution is a potential source of error for simulations. Errors in the distribution between both types of kinetic energy are more noticeable in objects with significantly different moments of inertia for each primary axis, as would be the case in a cuboid with different lengths for each axis, as seen in Fig. 7.

PBD and Jacobi are both able to maintain stable energy levels, but show some mild signs of energy being gained. Unity and Havok suffer from significant losses and show energy spikes as well, although Havok's is able to quickly fix errors due to the use of error caches. The errors present in both Unity and Havok's simulations seem to originate from too much rotation being applied. Most likely originating from having to solve the positional error and apply restitution by applying an impulse. Unity is the most unstable out of the tested engines since it shows energy rising continually, this error is visually apparent as well as the plane gradually

reaches greater heights. Havok's shows some energy gain, but still converges into a low energy state. This might be because Havok detects ambiguous situations where energy gains due to rotation are likely, and always minimizes energy.

4.6 Physically Impossible Scenarios

Within the regular use of a game engine by a developer, or during the execution of a gameplay environment, it is likely that at some point a physically impossible situation arises, either from the developer setting up impossible starting conditions or the gameplay enforcing a specific state. It is an important factor when choosing a physics engine that it is capable of remaining stable in these situations while minimizing physical inaccuracy.

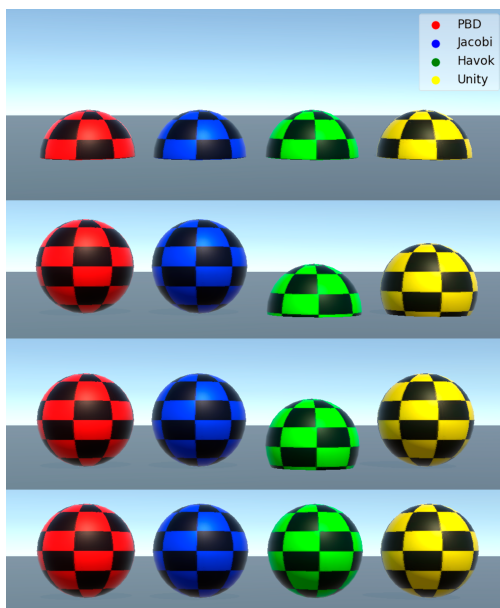


Figure 9: Collision solve of a scenario with colliding starting positions. The velocity based methods used by Unity and Havok cause the collision to be resolved over several frames rather than immediately.

While in this scenario, positional methods appear to be superior, further testing reveals that depending on the penetration depth and scenario they might get worse results. Repeating the same experiment but with a pyramid of penetration depth of 0.4 (nearly half of the cubes' height) shows similar results for Havok, Unity's top layers get a vertical velocity, PBD and Jacobi the pyramid is dismantled within the first simulation step with cubes belonging to the base and the middle layer all at the same height. This happens because while the bottom layer is being processed its members are projected vertically in order to correct the penetration with the plane, however this correction puts them in a position where they penetrate cubes in the middle layer. For lesser starting penetration depths this causes no issues, but with higher starting penetrations the resulting inter-

mediate state where the bottom and middle layer collision have so much penetration in the vertical axis that the shortest correction distance between each colliding cube is horizontal, leading to both layers expanding to each side, and dismantling the structure.

4.7 Unsolvble Constrained Scenarios

In any engine dealing with constrained systems, it is possible to create a configuration that is unsolvable due to having constraints with mutually exclusive solution domains. As was tested in Fig. 10. Havok provides a stable approximation of a state that averages each constraint in order to minimize error, the chain is still straight, and each link remains static, being the best option for this scenario. Jacobi manages a stable and static simulation as well. Finally, Unity and PBD oscillate in their solution with Unity even showing signs of divergence, with capsules moving to seemingly random positions before stabilizing at an oscillatory state switching between both positions at each step. This is a known issue with the Gauss-Seidel solver which Jacobi avoids, and it can lead to behaviour that is visually striking. The issue can be mediated by adding some compliance to each constraint, in which case the system converges to a stable and error minimizing state.

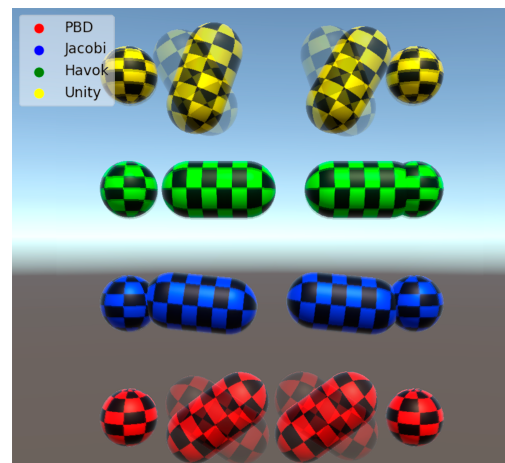


Figure 10: A chain with each end attached to a static point whose distances are greater than the chains' length, leading to a scenario with incompatible constraints. Havok and Jacobi can converge at a stable state, while Unity and PBD oscillate.

4.8 Performance

In order to test how the performance of each engine scales, three scenarios were tested with increasing number of elements, the chain (Fig. 3) to test the impact constraints have on performance, the pyramid (Fig. 6), meant to test the performance impact of resting contacts, and a new scenario consisting of a pile of capsules which will be referred to simply as capsule. Note

that both Havok and Unity's physics engine are implemented in C++, meanwhile our custom PBD implementation is in C# and has the overhead of being processed as part of a Unity MonoBehaviour being at a disadvantage. Measuring the performance of the different engines is further complicated by the fact it is hard to decouple the performance of collision restitution, time integration and constraint solving from aspects that are independent of simulation method, such as collision detection and entity systems.

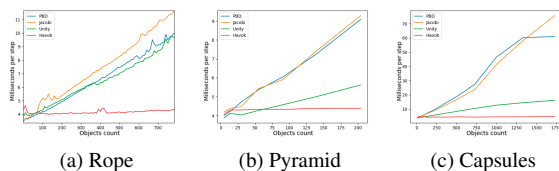


Figure 11: Milliseconds per physics substep of simulations.

While PBD and Jacobi's simulation seem to have similar performance to Unity when simulating constraints, the implemented collision detection is not able to compete with the one provided by Unity and Havok. The most important factor in this analysis is that all engines scale linearly, meaning that none is inherently more complex. The slope of each result will be largely impacted by low level optimizations, implementation language and parallelization and not so much by the simulation method itself. When it comes to sheer performance, Havok is unmatched in its handling of large scenarios.

5 DISCUSSION

After thoroughly testing all engines in different scenarios, it is clear no options is strictly better than the other, with each engine presenting unique strengths and issues. Table 1 combines the observations from all scenarios regarding problems with each method.

	Simulator			
	PBD	Jacobi	Unity	Havok
Velocity transfer	Accurate	Accurate	Inaccurate	Inaccurate
Constraint error	Noticeable	Noticeable	Very Noticeable	None
Inter-penetration	Significant	Very Significant	None	None
Velocity drift	None	None	Very Significant	Significant
Stacking	Rotational Error	Somewhat	Drifting	Stable
Rotational kinetic energy	Stable	Stable	Increase	Loss
Stable friction	No	No	No	Yes

Table 1: Physics Engine Comparisons

In terms of maintaining stable energy levels without producing any extra energy, PBD is the most reliable

option. Along with Jacobi it is the only method capable of accurately and quickly transferring velocity over a row of objects simulating a Newtons cradle with ease. It is stable handling of collision contacts is further supported by the results of (8a) being able to simulate objects with uneven axes lengths. When dealing with constraints, it loses some energy gradually but remains more stable, showing no energy being gained. When dealing with scenarios too complex for its current substep count (which dictates accuracy) the system will diverge causing massive energy gains. This is because PBD is meant to excel at correcting small errors, hence the use of substepping. When it comes to scenarios with stacks of bodies, PBD can allow for some inter-penetration of bodies and suffer from rotational error destabilizing the stack.

Using a modified version of PBD to use a Jacobi solver rather than Gauss-Seidel proves to offer little benefit, since the method cannot produce the same kind of stable energy. The issues of inter-penetration also present in PBD are more pronounced as well. However, in some cases it can be a better choice, when dealing with a scenario too complex for the current substep count PBD was shown to diverge, while Jacobi managed to maintain a more stable simulation as shown in (4b). Jacobi also suffers from less rotational error on resting stacks, but it is more vulnerable to destabilization due to the more severe interpenetration.

Unity's PhysX based physics engine's performance in the tests conducted revealed itself to be similar to Havok's. Some issues are shared by both engines, such as velocity transfer error. Both show velocity drifting, but Unity's was more severe. Unity is also prone to energy gains when tested using long cuboids (8c). The one advantage Unity was shown to have over Havok was better conservation of energy in constrained scenarios without collision, being able to maintain motion for a much longer time.

Havok's simulation proved itself to be incredibly stable and fast, being able to handle scenarios that were more complex and rarely showing any signs of divergence or overcorrection. However, it does suffer from velocity drift and loses a lot more energy than other engines. This is due to Havok detecting situations that could lead to energy being gained and always chooses the option that minimizes energy.

6 CONCLUSION

PBRBD can simulate scenarios with great accuracy showing great conservation of momentum and no energy gains and is a solid choice for any game engine, not only due to its accuracy but also due to being a unified solution merging particle systems and rigid body dynamics in a single engine and allowing for many different types of constraints. Its shortcomings become more

prevalent when simulating stacks, where the Gauss-Seidel solver struggles with rotational error. Switching the default solver by a Jacobi solver can increase robustness and handling of scenarios with mutually exclusive constraint solution domains, but decreases the accuracy of the simulation. In short, this novel method will facilitate development of scenarios mixing particle system or destructible objects while providing accurate and stable results while avoiding some known issues with velocity based methods such as velocity drifting.

A C++ implementation with a focus on code optimization and GPU parallelization to understand how performant PBRBD can get could further establish it as a solid choice for rigid body dynamics. The algorithm requires double precision floating points, which GPUs are not optimized to handle, leading to possible challenges. Furthermore, the parallelization of the Gauss-Seidel step can be achieved in a variety of ways, comparing the impacts on performance and accuracy of different techniques could be a source of future work. Due to the variety of supported constraints, PBRBD also seems adequate to create controls such as buttons, cranks, levers and knobs commonly seen in VR applications which could be further studied with user tests.

7 ACKNOWLEDGMENTS

The work reported in this article was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020.

8 REFERENCES

- [BET14] Jan Bender, Kenny Erleben, and Jeff Trinkle. Interactive simulation of rigid body dynamics in computer graphics. In *Computer Graphics Forum*, volume 33, pages 246–270. Wiley Online Library, 2014.
- [BML⁺14] Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. Projective dynamics: Fusing constraint projections for fast simulation. *ACM transactions on graphics (TOG)*, 33(4):1–11, 2014.
- [BMM17] Jan Bender, Matthias Müller, and Miles Macklin. A survey on position based dynamics, 2017. In *Proceedings of the European Association for Computer Graphics: Tutorials*, pages 1–31. 2017.
- [Dru07] Evan Drumwright. A fast and stable penalty method for rigid body simulation. *IEEE transactions on visualization and computer graphics*, 14(1):231–240, 2007.
- [MC95] Brian Mirtich and John Canny. Impulse-based simulation of rigid bodies. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 181–ff, 1995.
- [MHHR07] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.
- [MHTG05] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. Meshless deformations based on shape matching. *ACM transactions on graphics (TOG)*, 24(3):471–478, 2005.
- [MMC16] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, pages 49–54, 2016.
- [MMC⁺20] Matthias Müller, Miles Macklin, Nuttapong Chentanez, Stefan Jeschke, and Tae-Yong Kim. Detailed rigid body simulation with extended position based dynamics. In *Computer Graphics Forum*, volume 39, pages 101–112. Wiley Online Library, 2020.
- [MSL⁺19] Miles Macklin, Kier Storey, Michelle Lu, Pierre Terdiman, Nuttapong Chentanez, Stefan Jeschke, and Matthias Müller. Small steps in physics simulation. In *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 1–7, 2019.
- [MW88] Matthew Moore and Jane Wilhelms. Collision detection and response for computer animation. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 289–298, 1988.
- [NMK⁺06] Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically based deformable models in computer graphics. In *Computer graphics forum*, volume 25, pages 809–836. Wiley Online Library, 2006.
- [TBV12] Richard Tonge, Feodor Benevolenski, and Andrey Voroshilov. Mass splitting for jitter-free parallel rigid body simulation. *ACM Transactions on Graphics (TOG)*, 31(4):1–8, 2012.
- [XZB14] Hongyi Xu, Yili Zhao, and Jernej Barbic. Implicit multibody penalty-based distributed contact. *IEEE transactions on visualization and computer graphics*, 20(9):1266–1279, 2014.

On the Importance of Scene Structure for Hardware-Accelerated Ray Tracing

Martin Kacerik
Czech Technical
University in Prague
Karlovo namesti 13
121 35, Prague, Czech
Republic
kacerna2@fel.cvut.cz

Jiri Bittner
Czech Technical
University in Prague
Karlovo namesti 13
121 35, Prague, Czech
Republic
bittner@fel.cvut.cz

ABSTRACT

Ray tracing is typically accelerated by organizing the scene geometry into an acceleration data structure. Hardware-accelerated ray tracing, available through modern graphics APIs, exposes an interface to the acceleration structure (AS) builder that constructs it given the input scene geometry. However, this process is opaque, with limited knowledge and control over the internal algorithm. Additional control is available through the layout of the AS builder input data, the geometry of the scene structured in a user-defined way. In this work, we evaluate the impact of a different scene structuring on the run time performance of the ray-triangle intersections in the context of hardware-accelerated ray tracing. We discuss the possible causes of significantly different outcomes (up to 1.4 times) for the same scene and identify a potential to reduce the cost by automatic input structure optimization.

Keywords

real-time ray tracing, acceleration structures, bounding volume hierarchies

1 INTRODUCTION

Graphics APIs with access to ray tracing hardware features, such as Vulkan or DirectX, utilize internally built two-level acceleration data structures. These data structures are used during the ray tracing to accelerate a ray-triangle intersection. The data layout, as well as the build algorithm, is provided by a driver vendor and is typically opaque to the user of the API (except for several open-source driver implementations).

When an AS is being constructed for an existing scene, certain organized layout of the scene data is expected as an input for the algorithm. In this paper, we refer to this input layout as a *scene structure*. Intuitive transfer of a scene organized in a scene graph to this scene structure is natural, but likely a suboptimal approach, especially for more complex and dynamic scenes. Nevertheless, preserving the information from the scene graph is beneficial, as it connects the geometric and material properties, including UV coordinates, or object instancing.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

This work explores multiple available options for construction of an opaque acceleration structure for hardware ray tracing. We evaluate the impact of scene structuring in conjunction with different construction algorithm hyper-parameters on the traversal performance of the final acceleration structure. Finally, we reason about the differences in performance and outline a way to derive a better input scene layout automatically.

The paper is structured as follows: Section 2 presents relevant research in the field. Section 3 gives a brief introduction to the acceleration structure API. Section 4 explores the input data layout for bounding volume hierarchy (BVH) construction. Section 5 describes the evaluation process and presents the measured data. In Section 6, we discuss the results and draw an explanation for the measurements. Finally, Section 7 concludes the paper.

2 RELATED WORK

An exhaustive study on the topic of bounding volume hierarchies for ray tracing was recently presented by Meister et al. [Mei21a].

Our work is directly related to massively parallel GPU-accelerated BVH construction algorithms. One of the fastest among these techniques is the top-down, binning-based construction algorithm *linear BVH* (LBVH) proposed by Lauterbach et al. [Lau09a]. The LBVH was then extended to the *hierarchical LBVH*

by Pantaleoni and Leubke [Pan10a], who employed a *surface area heuristic* (SAH) for the upper levels of the hierarchy. The SAH was originally introduced by Goldsmith and Salmon [Gol87a] as a metric approximating the likelihood of a ray-volume intersection and it is employed for driving the vast majority of the BVH build algorithms. A different approach was taken by Meister and Bittner [Mei17a], who proposed a GPU-accelerated bottom-up build algorithm using agglomerative clustering: parallel locally-ordered clustering (PLOC). Recently, *PLOC++* by Benthin et al. [Ben22a] has been proposed, addressing certain technical weaknesses of the original PLOC, such as the number of dispatched GPU kernels.

The refinement of an existing BVH structure of a lower or deteriorated quality back to a near-optimal state was investigated by Benthin et al. [Ben17a], who presented a process of *partial re-braiding* to reduce overlaps in the AS and improve the SAH quality of the BVH. In a similar fashion, Hendrich et al. [Hen17a] proposed a *progressive hierarchical refinement*, a method of improving the outcome of a fast but low-quality BVH builder, such as LBVH. This method could be used to perform a build directly from the scene graph hierarchy.

In past years, major chip vendors, starting with NVIDIA, later joined by AMD and Intel, introduced ray tracing acceleration to their mainstream GPU lineup. Although the internal functioning is mostly hidden, there were attempts to improve the overall ray tracing performance by reorganizing the data used for the ray tracing process, while considering the ray tracing API as a black box. In particular, Meister et al. [Mei20a] investigated the possibilities of ray reordering, while Wald et al. [Wal20a] focused on exploiting the API design to improve the AS hierarchy for a special case of long and thin geometries. Our work aims to lay a foundation for further improvements by performing scene graph restructuring prior to passing the data to the ray tracing API.

3 ACCELERATION STRUCTURE API

In this work, we consider following ray tracing APIs: DirectX Raytracing (DXR), Vulkan and NVIDIA OptiX. Although they might differ in specific capabilities and naming conventions, they all conform to similar programming model.

Acceleration structure defined in a particular API is an opaque data structure utilized in subsequent queries to accelerate ray-object intersection. In general, the layout of the AS, as well as the related algorithm to build the AS, is internal to a specific implementation. Based on the public interface for an AS manipulation, we can derive the knowledge discussed in this section.

3.1 Data layout

The AS is formed in two logical levels - a *bottom level acceleration structure* (BLAS) and a *top level acceleration structure* (TLAS). BLAS nodes consist of geometry data (multiple disjoint geometries can be easily merged into one BLAS), while TLAS nodes reference the BLAS nodes and include their respective transformation and shading data (using a relevant shader index). The TLAS enables storage of geometry in a local coordinate system and supports geometry instancing, as illustrated in Fig. 1.

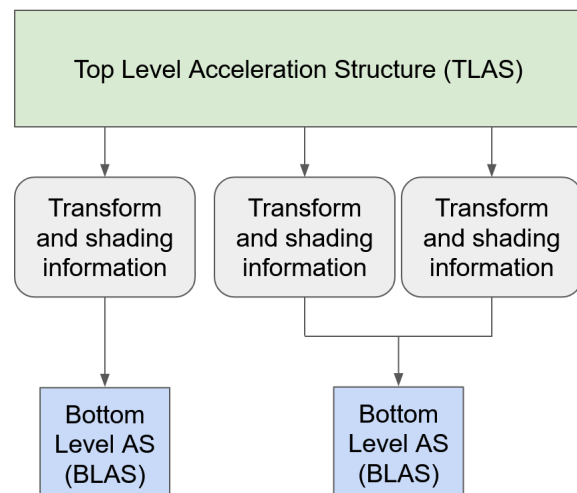


Figure 1: Logical representation of the acceleration structure design as exposed by contemporary GPU APIs.

Terms BLAS and TLAS are directly utilized in DXR and Vulkan APIs, while OptiX chooses terms *geometry acceleration structure* (GAS) and *instance acceleration structure* (IAS).

3.2 Build/update algorithms

The API distinguishes between two modes when building a new AS, either building from scratch or updating the existing AS. AS update corresponds to bounding volume refit, a technique available when specific conditions regarding the topology of the updated geometry are met. Specifically, only instance definitions, transform matrices, and vertex or axis aligned bounding box (AABB) positions are allowed to change during the update. Refit is fast but also likely to deteriorate the quality of the AS over time.

In accordance of the two-level logical hierarchy, both modes are executed in two steps. In the first step, BLASes are built. As long as there is enough memory available for auxiliary buffers required by BLAS builders, multiple BLAS builds can be scheduled to run concurrently without any additional synchronization. When all BLASes are available, the TLAS build

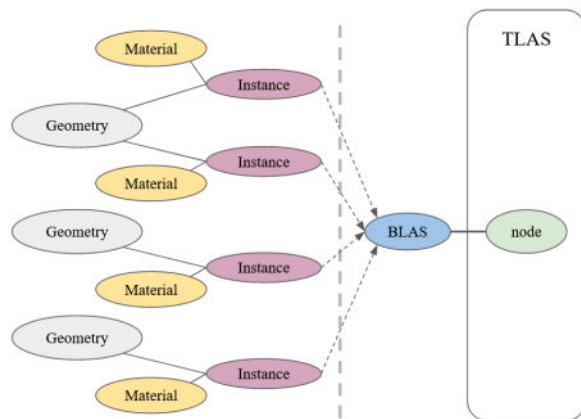


Figure 2: Input scene structure for AS construction, created by mapping a scene graph structure (grey, yellow, and pink nodes) to one joined BLAS node. Material and transformation information is not stored in the AS.

can start. Once finished, the AS is ready to be deployed for hardware-accelerated ray tracing.

Coarse control over the building process is allowed in the form of flags, hinting our preference to the builder. Notably, flags such as `PREFER_FAST_TRACE`, `PREFER_FAST_BUILD` (available in all mentioned APIs), or `LOW_MEMORY` (available in DXR and Vulkan) indicate that the underlying implementation is expected to utilize multiple different algorithms with different trade-offs in terms of the AS build speed, runtime traversal cost, or overall memory consumption.

4 MAPPING A SCENE GRAPH TO THE AS BUILD LAYOUT

3D scenes are typically stored in a scene graph hierarchy, maintaining all vital information about the relations of scene objects: i.e. instances, their geometries, materials, and transformations.

A valid method to submit such data for an AS build is to transform all geometries in place, merge them into one BLAS node, and assign it to a TLAS node with a unit transformation, see Fig. 2. Under such conditions, the BLAS builder is likely to execute the job to its full potential and build the AS of the best quality, thanks to global knowledge of the scene in one place. However, all the benefits of TLAS are given up, including any possibility of fast partial rebuilds and refits of the AS, geometry instancing, or referencing of a specialized shader for defined material.

Another approach to submit scene data to an AS building API is to map geometry nodes to BLAS nodes and object instances to TLAS nodes, see Fig. 3. However, as we show in the next section, submitting the scene graph data organized from a scene designer perspective can have significant performance consequences and will likely translate to a suboptimal acceleration structure.

5 EVALUATION

We evaluated several different ray tracing setups using eight static test scenes. The list of evaluated scenes with some of their parameters is shown in Table 1. All scenes were taken from Morgan McGuire's online archive [McG17a]. Default scene graph hierarchy was loaded from the scene source file. The measurements were performed on a computer equipped with Intel i9-10900X CPU, 128GB RAM, and NVIDIA RTX3080Ti GPU (driver v525.89.02).

Scene	Triangles	#Instances	Overlap
Fireplace room	143 173	51	4.1
Chestnut	316 880	5	3.6
Sibenik cathedral	75 284	1087	21.1
Crytek Sponza	262 267	393	11.1
Bistro interior	1 046 609	2062	22.8
Bistro exterior	2 832 120	1591	22.5
Power plant	12 759 246	57	8.7
San Miguel	9 980 699	2135	226.3

Table 1: List of used scenes and their geometric complexity. Overlap metric represents an overlap of axis aligned bounding boxes of instances in the scene and is computed in a following way: for each pair of AABBs of instances, we compute a surface area of their overlapping region. The overlap value is then the sum of these areas divided by a surface area of the scene's AABB.

Performance evaluation was done in a custom path tracing engine running on a Vulkan API backend. For each scene, we traced five 1920x1080 views with 2048 samples per pixel and a maximum recursion depth of eight. To focus the measurement on the AS performance, we employed simple Lambertian BRDF for surface interaction, avoiding any complex shading computations.

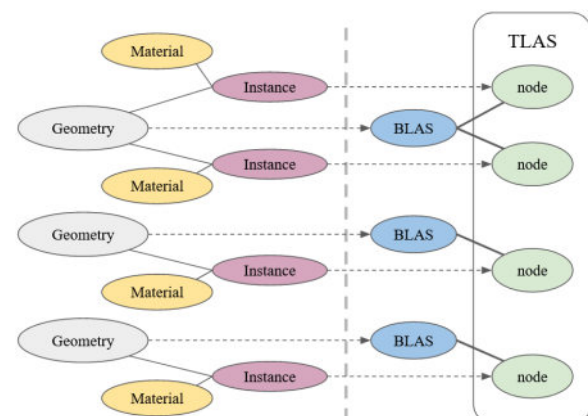


Figure 3: Input scene structure for AS construction, created by mapping a scene graph structure (grey, yellow, and pink nodes) to one BLAS node per scene geometry, with the possibility of instancing. Material and transformation information is stored in TLAS nodes.

Every sample tracks the number of traced rays internally, which is then added to an atomic counter. The measurements showed that the atomic add has an insignificant impact on the performance. Time is tracked through the timestamp query API with advertised nanosecond precision.

We measured the following configurations of the scene: input scene structures mapped as described in chapter 4: (1) one BLAS per geometry node in the scene and (2) one BLAS for the whole scene. Both configurations are then measured with three different opaque AS builder options: `PREFER_FAST_TRACE`, `PREFER_FAST_BUILD`, and `LOW_MEMORY` in the build AS mode (AS is built from scratch). The rendered output is always identical. The measured values are presented in Figures 4, 5, and 6. Reported values represent sum of costs of both AS levels, where TLAS times or memory consumption are negligible compared to BLAS values.

6 DISCUSSION

The measured results support our initial assumption that the scene structure provided to the opaque AS builder significantly impacts the final ray tracing performance. The relation between the geometric complexity and overlap of the acceleration structure nodes seems like a good final performance predictor. This correlation can also be observed in the visualizations shown in Tab. 2 and the traversal performance in Fig. 4. The superior traversal performance of one BLAS per whole scene is especially pronounced in the scenes with an otherwise high overlap of instance AABBs.

The hypothesis outlined in section 4 states, that the global scene knowledge available for the AS builder in one BLAS node will lead to superior AS quality and thus better runtime performance. We can conclude that this hypothesis was proven right in almost all cases, except the Fireplace scene with the fast trace setting on. The behavior in the Fireplace scene is unexpected and it suggests that, in some cases, a better acceleration structure can be found when the scene graph holds useful structural information that is not found by the AS builder. As an anomaly, this case is interesting, and it will be the subject of further investigation.

On average, the trace speed of the BLAS per scene is 1.3 times higher in the `PREFER_FAST_TRACE` case, 1.41 times higher for the `PREFER_FAST_BUILD` case, and 1.37 times higher for the `LOW_MEMORY` case than that of the one BLAS per geometry.

In contrast to the trace speed, the build speed of the AS builder itself, reported in Fig. 5, remains mostly in favor of multiple smaller BLAS nodes compared to one big BLAS node. This is expected due to the hardware ability to run the construction of multiple nodes concur-

rently, as well as due to the known $O(n \cdot \log(n))$ complexity of the AS construction algorithms.

We consider minimization of the node overlap as one of the key steps in the potential scene restructuring process that would optimize the trace speed while keeping the high level scene structure. The problematic nodes, causing unnecessary overlap, have to be identified and, based on the local decision, cut or joined. This decision has to be guided by the local complexity, possibly predicted by metrics like SAH.

A possible further research direction is the analysis of the influence of BLAS orientation. Axis-aligned bounding boxes, which serve as an underlying bounding volume for the AS, are not invariant to rotation of bounded geometry. Discovering the optimized initial rotation for the geometry, minimizing the SAH cost and likely the overlap of its AABB, can thus benefit the overall performance.

7 CONCLUSION

This paper discussed the problem of the dependence of the hardware accelerated ray tracing performance on the AS construction input scene structure. We showed that although the benefits of a two-level acceleration structure are numerous (instancing, local transformations, material referencing, fast refit), the performance penalty for suboptimally organized scenes can be significant.

As indicated by the Fireplace scene, we believe that it is possible to find a scene structure that keeps the TLAS benefits and also minimizes the performance impact. This paper aims to provide the groundwork necessary for the following research of a scene graph layout restructuring to achieve superior results with an opaque AS builder.

8 ACKNOWLEDGMENTS

This research was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS22/173/OHK3/3T/13 and the Research Center for Informatics No. CZ.02.1.01/0.0/0.0/16_019/0000765.

9 REFERENCES

- [Ben17a] Benthin, C., Woop, S., Wald, I., and Áfra, A. T. Improved two-level bvhs using partial re-braiding. In Proceedings of High Performance Graphics (New York, NY, USA, 2017), HPG '17, Association for Computing Machinery, pp. 1-8.
- [Ben22a] Benthin, C., Drabinski, R., Tessari, L., and Dittebrandt, A. Ploc++: Parallel locally-ordered clustering for bounding volume hierarchy construction revisited. Proc. ACM Comput. Graph. Interact. Tech. 5, 3 (Jul 2022), pp. 1-13.

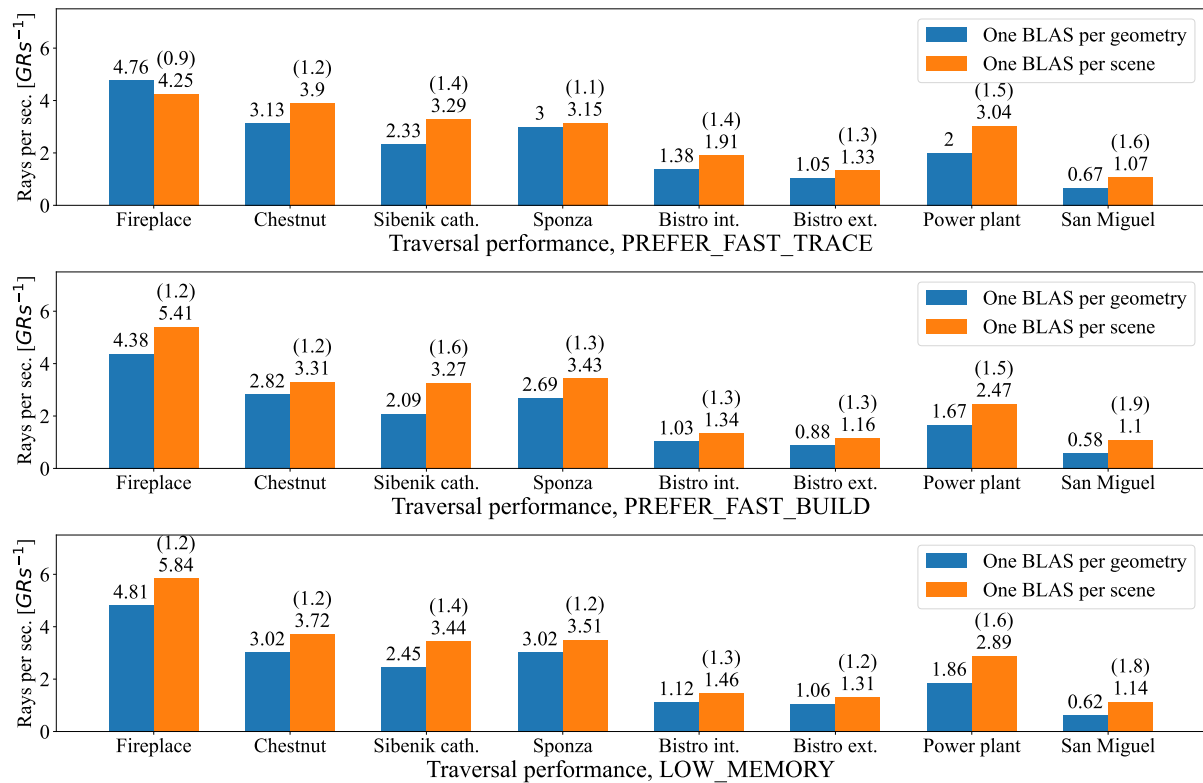


Figure 4: Ray tracing speed of acceleration structures constructed with six different configurations. The trace speed is measured in GigaRays per second (the higher the better).

- [Gol87a] Goldsmith, J., and Salmon, J. Automatic creation of object hierarchies for ray tracing. *IEEE Computer Graphics and Applications* 7, 5 (May 1987), 14-20.
- [Hen17a] Hendrich, J., Meister, D., and Bittner, J. Parallel bvh construction using progressive hierarchical refinement. *Computer Graphics Forum (Proceedings of Eurographics 2017)* 36, 2 (2017), 487-494.
- [Lau09a] Lauterbach, C., Garland, M., Sengupta, S., Luebke, D., and Manocha, D. Fast bvh construction on gpus. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 375-384.
- [McG17a] McGuire, M. Computer graphics archive, July 2017. <https://casual-effects.com/data>. Accessed: 2023-03-27.
- [Mei17a] Meister, D., and Bittner, J. Parallel locally-ordered clustering for bounding volume hierarchy construction. *IEEE transactions on visualization and computer graphics* 24, 3 (2017), 1345-1353.
- [Mei20a] Meister, D., Boksansky, J., Guthe, M., and Bittner, J. On ray reordering techniques for faster gpu ray tracing. In *Symposium on Interactive 3D Graphics and Games (New York, NY, USA, 2020)*, I3D '20, Association for Computing Machinery, pp. 1-9.
- [Mei21a] Meister, D., Ogaki, S., Benthin, C., Doyle, M. J., Guthe, M., and Bittner, J. A survey on bounding volume hierarchies for ray tracing. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 683-712.
- [Pan10a] Pantaleoni, J., and Luebke, D. Hlbvh: hierarchical lbvh construction for real-time ray tracing of dynamic geometry. In *Proceedings of the Conference on High Performance Graphics* (2010), pp. 87-95.
- [Wal20a] Wald, I., Morrical, N., Zellmann, S., Ma, L., Usher, W., Huang, T., and Pascucci, V. Using hardware ray transforms to accelerate ray/primitive intersections for long, thin primitive types. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 3, 2 (2020), 1-16.

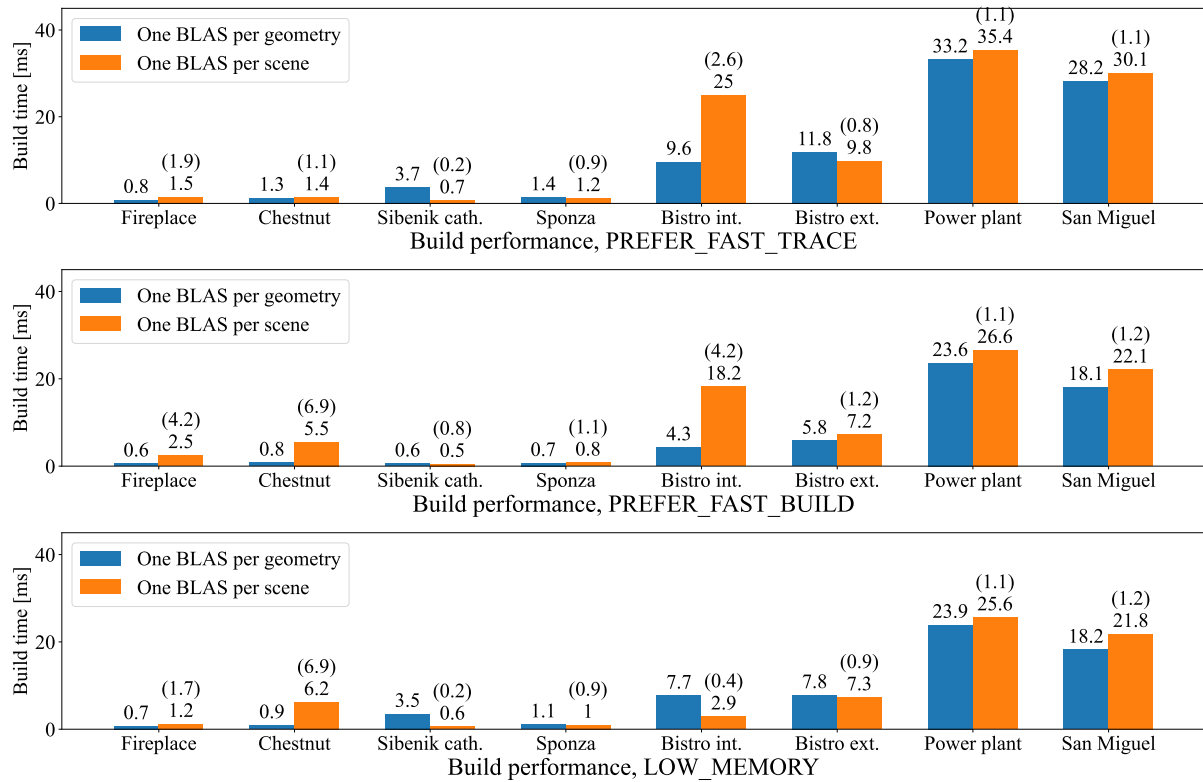


Figure 5: Build times of acceleration structures constructed with six different configurations, measured in milliseconds (the lower the better).

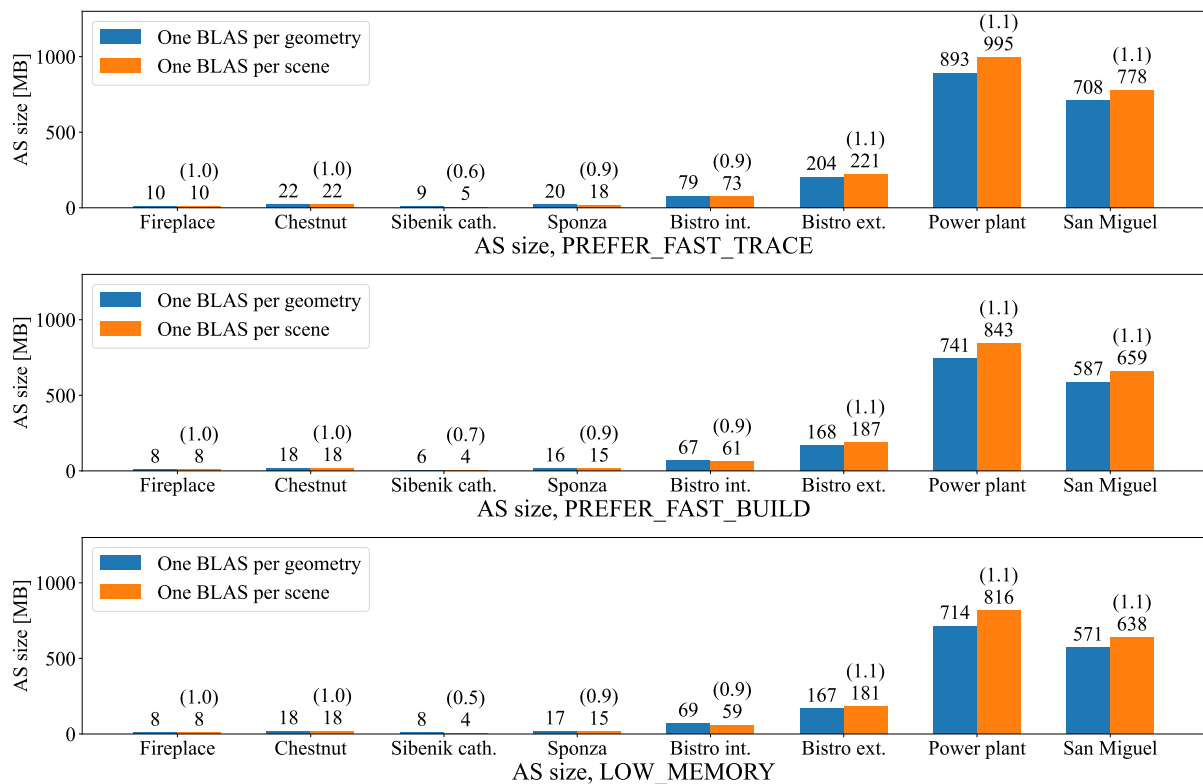


Figure 6: Memory consumption of acceleration structures constructed with six different configurations, measured in MegaBytes (the lower the better).

Scene	Diffuse view	Instance view	Instance AABB overlap
Fireplace room			
Chestnut			
Sibenik cathedral			
Crytek Sponza			
Bistro interior			
Bistro exterior			
Power plant			
San Miguel			

Table 2: Evaluated scenes, with visualization of complexity of the scene for "One BLAS per geometry" case. The middle column shows ID buffers with unique colors for each object instance, and the right column shows heat maps indicating the overlap of bounding boxes of instances. The lower and upper bounds of the heat map scale defined are shown in the top left and top right corners of the heat maps, respectively.

Real-Time Visual Analytics for Remote Monitoring of Patients' Health

Maryam Boumrah^[1]
Boumrah.maryam@inpt.ac.ma

Samir Garbaya^[2]
Samir.garbaya@ensam.eu

Amina Radgui^[1]
Radgui@inpt.ac.ma

^[1] Centre d'études doctorales
Télécoms et Technologies de
l'Information (CEDOC-2TI), INPT,
Av. Allal Alfassi, Rabat, Morocco

^[2] Laboratoire END-ICAP,
INSERM UMR1179, Arts et
Métiers Institute of Technology,
CNAM, LIFSE, HESAM
University, F-75013

ABSTRACT

The recent proliferation of advanced data collection technologies for Patient Generated Health Data (PGHD) has made remote health monitoring more accessible. However, the complex nature of the big volume of medical generated data presents a significant challenge for traditional patient monitoring approaches, impeding the effective extraction of useful information. In this context, it is imperative to develop a robust and cost-effective framework that provides the scalability and deals with the heterogeneity of PGHD in real-time. Such a system could serve as a reference and would guide future research for monitoring patient undergoing a treatment at home conditions. This study presents a real-time visual analytics framework offering insightful visual representations of the multimodal big data. The proposed system was designed following the principles of User Centered Design (UCD) to ensure that it meets the needs and expectations of medical practitioners. The usability of this framework was evaluated by its application to the visualization of kinematic data of the upper limbs' movement of patients during neuromotor rehabilitation exercises.

Keywords

Visual analytics; Patient Remote Monitoring; Brain Stroke; Rehabilitation; User-Centered Design; Kafka.

1. INTRODUCTION

The exponential growth of digital technologies designed for health data capture, such as sensors and wearable devices, has enabled real-time collection of big amounts of PGHD during remote monitoring. This big data has the potential to provide valuable insights to various healthcare stakeholders, particularly clinicians. The clinicians must effectively gather and integrate data from multiple sources, synthesize a comprehensive medical interpretation based on the patient's medical record and make informed decisions [1]. However, to take full benefits of this data, it is crucial to address the challenges of its complexity, including the need for rapid processing, analysis, interpretation, and understanding. The development of new methods and tools to address these challenges is of utmost importance in this era of rapidly advancing technology.

Thus, by implementing a visual analytics solution, the massive amount of data is transformed into meaningful knowledge [2]. The use of visual analytics of medical Big Data plays an important role for taking proactive medical decision and providing better

healthcare of patients. Visual analytics is the concept of combining data analysis, visual representation, and human interaction to extract insights and make decisions. Multiple areas of expertise are involved in the visual analytics process including data mining, machine learning and advanced graphic representations. Visual Analytics first emerged from the necessity of upgrading from confirmatory data analysis, which was used to represent the results, to the exploratory analysis involving the interaction with the data [3]. As defined by Keim et al., visual analytics is the integration of computerized analytics methods and visual interaction tools to gain helpful perception and decision based on big and heterogeneous data [4]. Furthermore, the advent of real-time data processing has given rise to dynamic visual analytics, which refers to the integration of knowledge discovery and interactive visual interfaces. This facilitates the analysis of data streams and enables situational awareness in real-time [5]. Over the past decade, dynamic visual analytics has garnered significant attention from research teams across multiple fields due to its potential for supporting immediate decision-making and raising awareness.

Despite the notable efforts in recent years to implement visual analytics technologies in healthcare sector, the adoption of real-time visual analytics solutions in patient health monitoring remains a challenge. The high pace and time sensitivity of medical data, along with the significant impact of any delay on clinical decisions and the safety of patients, requires that visual analytics systems must operate with minimal latency. The current state of the art falls short in offering a comprehensive solution that encompasses all essential features such as scalability, capability to handle heterogeneous data, low latency, independence from treatment systems, user-friendly visualization, cost-effectiveness, and extensiveness. Taking into consideration the important features for real-time visual analytics in patient health monitoring, this paper proposes a unified framework based on optimized open-source technologies such as Apache Kafka and Dash [6][7]. This framework is presented to efficiently manage real-time monitoring of a considerable number of patients, each with the necessary number of sensors. The usability of the proposed approach was proved by its application for monitoring post-stroke patients during in-home rehabilitation therapy. The system was developed following the User-Centered Design (UCD) approach, with three iterations for refining the user requirements.

The research reported in this paper highlights the potential of visual analytics technologies for improving the quality of patient health monitoring and allowing appropriate clinical decision making. The proposed unified framework offers a promising solution to meet the challenges facing real-time patient health monitoring. The paper is organized as the following: Section 2 provides a comprehensive review of the state-of-the-art research in visual analytics solutions, the challenges faced in the interpretation of health data and the UCD approach for patient health monitoring. Section 3 describes the proposed framework. The development process is described in the section 4. Finally, the conclusion and the research perspectives are presented in section 5.

2. RELATED WORK

The application of advanced visual analytics, incorporating big data analysis, interactive graphical representations, and AI algorithms plays a vital role in monitoring patient's health. Despite its potential benefits, challenges arise in gaining valuable insights from patient-generated data. These challenges encompass data, human, and tool-related limitations. In particular, the quality of health data, restrictions in data access, diversity of data sources, and scarcity of emergency data pose significant difficulties [8]. Moreover, the aggregation of massive data sources

into a unified platform, maintenance and storage of growing data volume, integration and interoperability of diverse data types and structures, and increased data analysis time with respect to data volume are key challenges in extracting information from big health data [9]. The complexity of health data and the lack of data standardization may also pose a challenge to the user and could result in misinterpretation [10]. On the user side, the understanding and interpretation of visual representations of health data are subject to the personal perception, leading to differences in interpretation of data and feedback on the visualized information between patients, clinicians, and data analysts [11]. Additionally, the insufficiency of data analysts and the lack of appropriate IT expertise among healthcare practitioners for processing, visualizing, and interpreting patient-generated data is a critical issue that needs to be addressed [12]. Furthermore, health stakeholders may not be prepared to adopt new systems and acquire the necessary skills for data handling [13]. The implementation of visual analytics tools for healthcare data, their development, and the creation of different methods of healthcare data representation remain significant challenges in this field.

In the domain of health monitoring and assessment, the implementation of secure and reliable information systems is of utmost importance. To achieve this, it is necessary to include in these systems appropriate management, analysis, and visualization tools [14] [15]. Feller et al. presented a visual analytics tool for pattern recognition in patient-generated data, which aimed to aid clinicians in identifying systematic and clinically meaningful patterns, and reducing perceived information overload [16]. Similarly, Vu et al. proposed a visual analytics approach for identifying informative temporal signatures in continuous cardiac monitoring alarms, using retrospective evaluation of a middleware alarm escalation software database in conjunction with visualization [17]. Dagliati et al. introduced the MOSAIC dashboard system, they used predictive modeling and longitudinal data analytics to support clinical decision-making. Their system integrated multiple-source data and uses visual and predictive analytics to enhance the management of chronic diseases, such as type two diabetes, through the successful implementation of the learning cycle of healthcare system [18].

The accuracy of health data exploration results relies heavily on the visualization tools and the used data processing interfaces. To address this, the field of human-machine interface has adopted the approach of User-Centered Design (UCD) which prioritizes the end user's requirements and needs in the development cycle of these tools. UCD follows an iterative design process, where end users are involved in the

evaluation of the outcome of the system. This approach has been widely adopted in health data processing and representation, as evidenced by several studies in the literature. For instance, Hobson et al. applied UCD to develop a telehealth system named TiM, for collecting information from motor neuron disease patients which was reviewed by care providers [19]. Similarly, Griffin et al. [20] used UCD to develop an mHealth approach for colorectal cancer monitoring in elderly patients using virtual human technology. Raghau et al. developed SMARTHealth, a UCD-based mobile application for Clinical Decision Support in cardiovascular disease risk [21]. Backonja et al. used UCD to study the visualization of health data [22], while Petersen et al. developed a mobile system using UCD for collecting and analyzing data from older adults [23]. David et al. developed a virtual reality-based interactive system for upper extremity rehabilitation of post-stroke patients [24]. Caporaso et al. studied the usefulness of biomechanics and neuroscience in designing a personalized monitoring system for hand rehabilitation [25]. Osborne et al. proposed a UCD-based mobile health application for neuro rehabilitation monitoring of stroke survivors [26]. Wentink et al. used UCD to analyze the requirements of users in eRehabilitation for post-stroke patients [27].

Despite existing visual analytics solutions in patient health monitoring, there's a need for a framework that effectively addresses the limitations of the existing systems and challenges. By adopting user-centered design principles and incorporating open-source technologies, the platform *could be reliable* and cost-effective solution for patient health monitoring, handle diverse data types and structures, integrate multiple data sources, and prioritize user needs.

The objective of the study reported in this paper is to develop a reliable architecture for a unified real-time patient health monitoring framework using dynamic visual analytics. To design the system and determine its components, a thorough review of the literature was conducted to answer two main research questions:

1. What are the main features of an efficient visual analytics system for real-time in-home patient monitoring?
2. What are the fundamental components of a new system that integrates the identified features?

A systematic review [28] of the most relevant literature on visual analytics for real-time monitoring of stroke patient health was conducted as part of this project. The literature was analyzed to identify the most critical features of an efficient system that aligns with the objectives of this study. The main identified features were scalability, independency, extensiveness, the ability to handle heterogeneous data, real-time interaction, and patient health status prediction. Additionally, the system should be accessible on a distributed platform to users who might not have access to high performance computing devices. Thus, by adopting user-centered design principles and incorporating open-source technologies, the platform *could be reliable* and cost-effective solution for patient health monitoring, handle diverse data types and structures, integrate multiple data sources, and prioritize user needs.

The results of the review led to the development of a new framework based on Lambda architecture [29]. This framework is described in the following section.

3. PROPOSED FRAMEWORK

The proposed framework combines both streamed and batch data into a unified pipeline, allowing real-time visualization, analysis, and data optimization.

As illustrated in Figure 1, the proposed framework consists of five main components. The data captured by wearable sensors are ingested into the pipeline via an event hub. The use of an event hub, which is scalable and easy to manage, is crucial for incorporating data captured by multiple and different types of sensors into the system. Upon ingestion into the framework, the data are simultaneously directed to both the streaming and batch layers. The batch layer primarily includes a storage unit, which continuously accumulates data volume to provide historical patient data for reference. It is used to develop a reliable machine learning models for health data optimization through feature extraction during the preprocessing phase. In contrast, the streaming layer is responsible for delivering real-time patient data with minimal latency to the therapist dashboard. Prior to display, the streamed data undergoes preprocessing for direct visualization and analysis. The machine learning models from the batch layer will be used to analyze the data in the streaming layer, and a postprocessing step is necessary to reduce false positives and extract the significant information to display in a dynamic visualization mode. In addition, the system includes a service layer that serves as an intermediary between the historical data and the streaming layer. The machine learning models generated from the historical data are forwarded to the streaming layer through this

service layer. The system also incorporates a webserver interface between the streaming layer and the application layer, providing a seamless integration of the two layers. The application layer, located on the user side, presents a dashboard for interaction of the user with the various components of the monitoring

system. This layer enables the user to visualize real-time data before and after the optimization process through a web application. This feature provides the user with a comprehensive understanding of the health status of the patient being monitored.

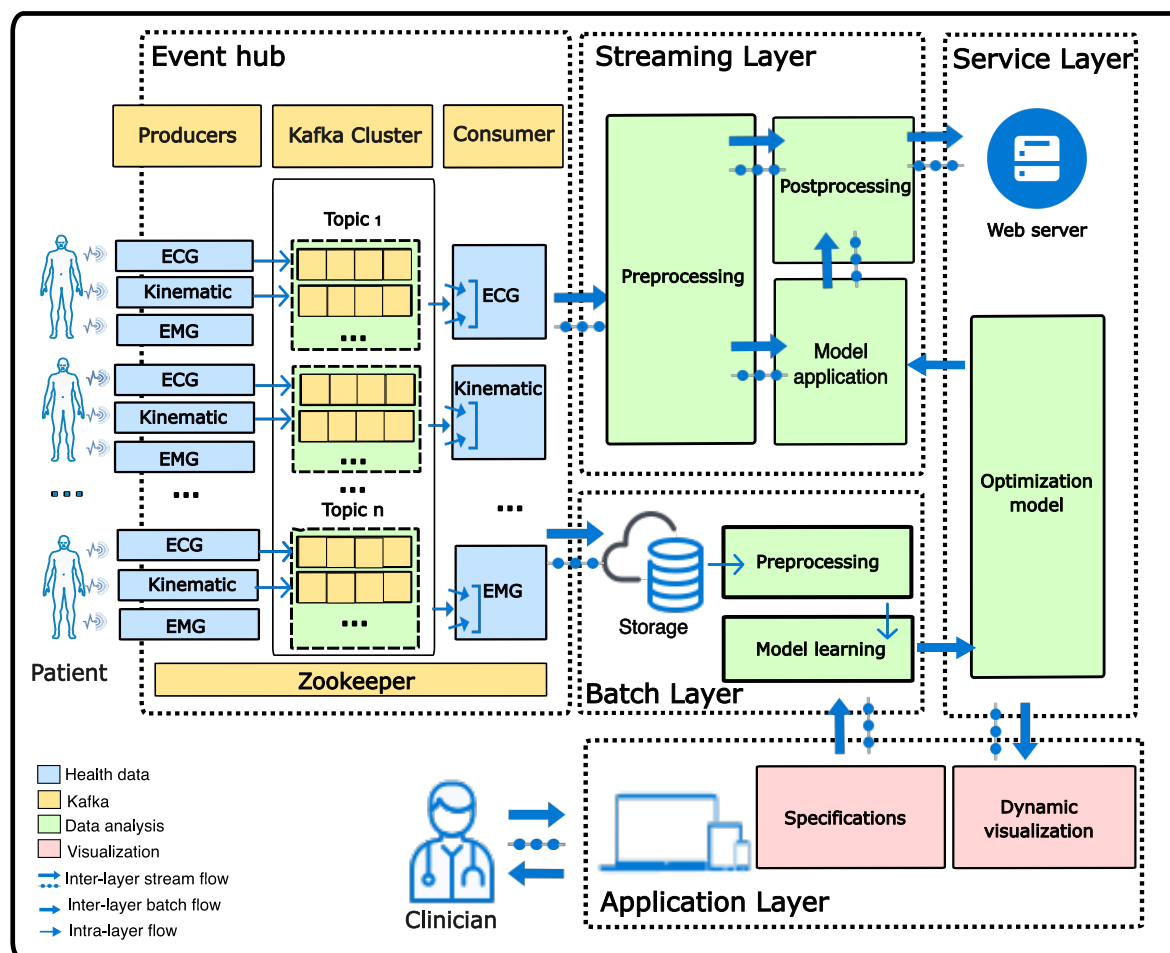


Figure 1. Architecture of the real time visual analytics framework for remote health monitoring and the structure of Kafka pipeline

The proposed system facilitates the interaction through customizable display modes and adjustable specifications of relevant information. For instance, in the context of post-stroke rehabilitation monitoring, the user can select the motor function data to visualize, and the specific optimized alerts sent to the therapist. This could enhance the evaluation of the patient's performance and health status, leading to a more personalized therapy approach. The multiple selective alerts enable real-time feedback to the clinician, providing crucial support in the case of emergency situations and facilitating the adjustment of rehabilitation protocol as necessary. This feature brings an important added value for the effectiveness and efficiency of the real-time patient health monitoring framework.

4. APPLICATION OF THE PROPOSED FRAMEWORK TO POST STROKE IN-HOME REHABILITATION

The objective of the development of this framework is to propose a real-time visual analytics system to assist physiotherapists in monitoring post-stroke patients when they practice rehabilitation services at home. The system described in this paper is focused on real-time visualization of medical data streams. The development of the framework was carried out following the principle of user-centered design (UCD) to ensure its effective usability. This process aims to analyze the application context and user requirements, design and develop the prototypes and evaluate the,

effectiveness of the system. Thus, by **repeating** the activities flow until the usability is redeemed by the rehabilitation experts who are the target end users of the system. The UCD process was performed by generating three iterative cycles: analysis, design,

prototyping, testing, and design refinement (Figure 2). This section provides a detailed description of the research activities and the methods used in each iteration.

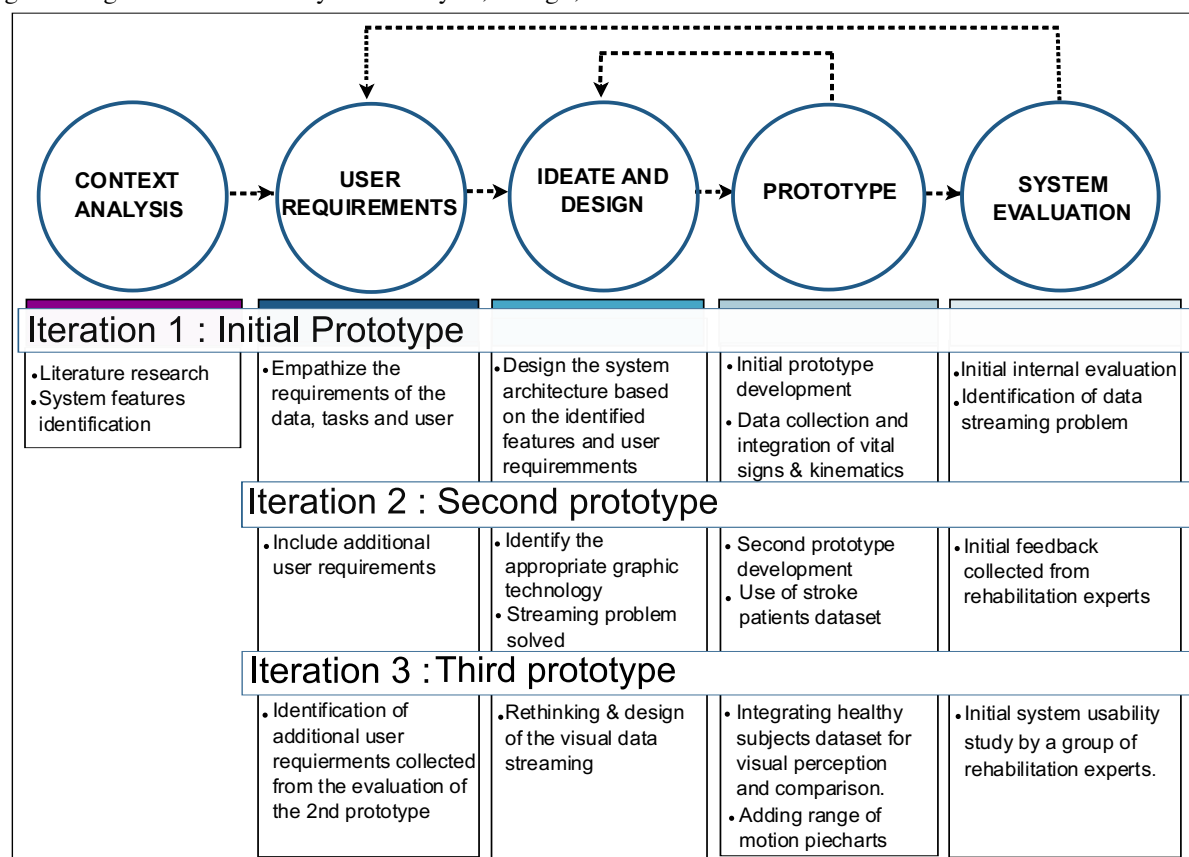


Figure 2. The workflow of the development process of the visual analytics framework

First Iteration

After analyzing the context of the proposed framework, it was critical to identify the requirements of the rehabilitation experts as they are the main end-users of the system.

In the preliminary iteration of the system design, emphasis was made on the identification of the requirements of the rehabilitation experts. This task was carried out by the research team members who have expertise in neuromotor rehabilitation practices, and they were in regular interaction with physiotherapists and neurologists in the hospital. Thus, for a more exhaustive understanding of the user requirements, the design triangle: Data–Users–Tasks was identified [30] and adopted as guideline:

Data requirements: The classification of patient-generated health data (PGHD) obtained during post-stroke rehabilitation sessions is crucial to provide meaningful insights into the patient's well-being. To this end, the PGHD are categorized into four key

outcome measures: quality of life measures, activity measures, balance measures, and motor function measures [31]. The quality-of-life measures provide a holistic evaluation of the patient's physiological and psychological well-being, including vital signs and emotions. The activity measures evaluate the patient's ability to perform rehabilitation exercises, while the balance measures inform about the patient's ability to maintain stability. Finally, the motor function measures monitor physical capabilities such as range of motion and muscle strength. In light of the aforementioned categorization, the health **data generated during post-stroke rehabilitation sessions** is primarily numeric, time-oriented, and multivariate in nature, making it challenging to manage through conventional systems due to its alignment with the requirements of Big Data such as high volume, rapid velocity, diverse variety, complex variability and time-sensitive nature.

Tasks and user requirements: The proposed system is intended for rehabilitation specialists such as rehabilitation physicians, physiotherapists, and human

movement scientists. The main objective is to enable remote monitoring of stroke patients during rehabilitation sessions. To achieve this objective, the system must provide users with access to real-time and historical health data through intuitive graphic representations and an interactive dashboard. This will enable rehabilitation experts to assess patients' physical abilities to perform the prescribed exercises and monitor their health status during rehabilitation practices. Additionally, the system should enable the analysis, interpretation, and utilization of collected data to support clinical decision-making and detect potential abnormalities. Furthermore, in order to optimize the therapy protocol, the users expressed the need for comparing real-time data with historical datasets from previous sessions for the same patient or with comparison with other patients. To fulfill these requirements, the system's dashboard and data representation tools must be user-friendly, enable simple and straightforward manipulation of data to extract the desired information. The goal is to ensure that the system is accessible and easy to understand, allowing users to effectively analyze and utilize the data.

After the identification of the user requirements, the design phase was initiated. It was focused on the selection of open-source technologies to implement a unified framework for post-stroke rehabilitation monitoring. To meet the specified needs, Apache Kafka was selected as the central event hub, with Python libraries for data processing and Dash was used for graphical visualization. Apache Kafka has an open and distributed architecture. It can handle real-time data ingestion from multiple producers, scale up to handle high-volume data, offer low latency, and ensure robust reliability [32]. These features made it well suited for this application. In a hypothetical scenario where there are no network delays, the processing is efficient, and the hardware is properly configured, a Kafka system with *five* brokers and 12GB of memory could sustain around 55,000 events per second [33]. If each patient wore *five* sensors generating 50 events per second, the system could theoretically monitor up to 220 patients:

$$220 \text{ patients} = \frac{55,000 \text{ events per second}}{50 \text{ events per second} \times 5 \text{ sensors per patient}}$$

However, it's important to keep in mind that this is a simple approximate estimation but the actual performance depends on various factors such as the complexity and size of the events, the processing load on the system, CPU resources, hard disk I/O speed, network bandwidth, message size and frequency, and the configuration parameters [34].

The development of the first prototype of the system was carried out using Python 3.8.2, Kafka 2.13-2.8.0, and Dash 2.0.0. The Dash framework was chosen for data visualization due to its flexibility and easy connectivity with Kafka using Python. Dash is built on the Flask web providing a simple and straightforward interface for creating rich visualizations and dashboards based on complex data and allows users to interact with the data through various controls and filters. As shown in Figure 1, the data collected from each type of sensor for a single patient was represented by a producer. The data collected from each type of sensor for a specific patient were transmitted by a producer to the corresponding partitions within a unique topic representing the patient. This organizational structure facilitated the subsequent processing and analysis of data. The data was then consumed based on their type but not on the patient ID, as they were organized into consumer groups to facilitate the analysis in the following step. The processed data were then transmitted to the Dash application via a web server for real-time display.

The evaluation of the system's functioning is important for the preliminary prototyping stage. In order to assess the system's performance, two datasets related to rehabilitation were selected for analysis. The first dataset consists of hand kinematics data of 22 healthy subjects while performing daily activities [35]. The second dataset includes vital sign recordings of 30 healthy individuals, including ECG, blood pressure, and respiration rate [36]. To mimic real-time data flow, the datasets were continuously looped and ingested into the prototype. The user dashboard provides a visual representation of the processed data, as illustrated in Figure 3. This approach allowed for the thorough evaluation of the system's capabilities, providing valuable insights into the system's performance, and enabling further improvements.

An initial usability evaluation of the system was conducted by the research team to assess its performance. The evaluation involved integrating four data streams into the prototype pipeline and assessing their real-time display.

Results of the evaluation of the first prototype:

During the running of the framework, the system initially displayed the data properly with an acceptable latency for a short time but after few minutes, and as the number of streams increased, the live graph display was slowed down reaching a latency of *three* seconds and later the flow of data was completely suspended at the application layer of the prototype. This issue had a significant impact on the aspect of real-time data visualization which had to be addressed in the second iteration. The results of this evaluation provided important insights into the limitations of the system and areas for improvement, highlighting the

need for further optimization of the prototype to meet the requirements of real-world applications and the needs of the target user. The objective was to obtain

smooth, scalable, and low-latency data visualization experience.

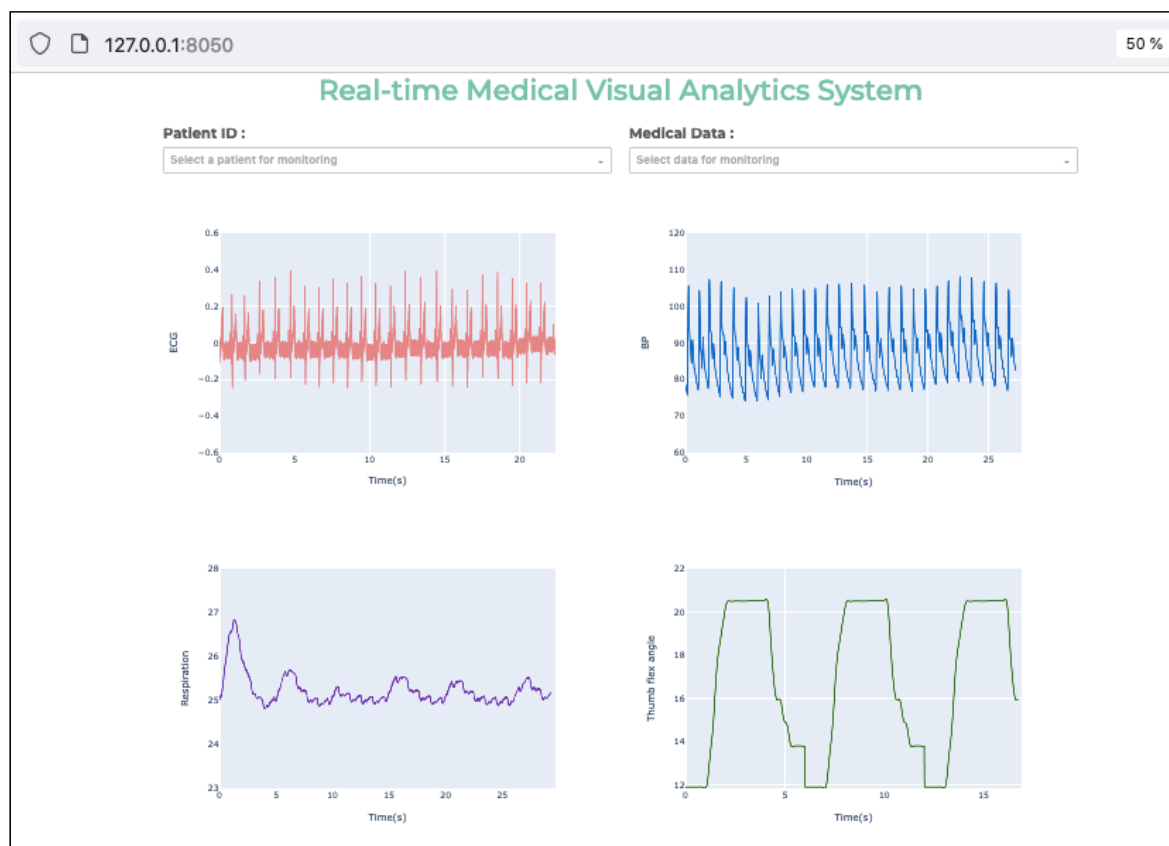


Figure 3. First prototype: visualization of the multimodal health data streams in the dashboard

Second Iteration

In order to address the limitations of the first prototype, a second development iteration was necessary. Then, an evaluation of the system's performance was carried out using real datasets of post-stroke patients practicing rehabilitation exercises.

To fulfil the new identified requirements, the technology used for graphical display using Dash was reviewed in order to avoid the blockage of data stream in time and ensure smooth display of the graphical representations. In the first prototype, the graphs were based on SVG (Scalable Vector Graphics) and multiprocessing. The multiprocessing in Python guarantees the parallelism of multiple flows as [processes do not share the same memory space](#). However, multithreading could be a better choice for the current system because the threads are lighter and less likely to cause overload. Additionally, the SVG is an easy option for rendering high-quality vector graphics, but its performance is limited. As an alternative solution, the WebGL provides a JavaScript API that allows to create GPU-accelerated graphics. The second prototype was developed by replacing the

old tools and inbuilding the combination of Multithreading method and WebGL technology. [As a result, the visualization of the data flow in the implemented system is currently smooth and continuous in real-time. It supports a significant amount of data streams, including nine types of heavy streams that are updated every 100ms. Additionally, the system maintains a very acceptable latency that does not exceed one second.](#) In order to assess the performance of the prototype, it was essential to conduct tests using real data that simulate a typical rehabilitation session for post-stroke patients. Various datasets from real stroke patients have been reported in the literature.

The recent dataset of post-stroke upper limb kinematics of daily living tasks (UZH) was deemed to be the most suitable in the existing datasets of stroke rehabilitation due to its corresponding number of subjects (20 stroke patients and [five](#) healthy individuals) and the variety of the covered upper limb rehabilitation activities (30 exercises). [This data was collected by Averta et al. as part of U-limb, which is a large and multi-modal database that was released to](#)

help the research contribution towards the assistive rehabilitation of the upper extremity of post stroke survivors [37]. To acquire the upper limb kinematic data, inertial wearable motion capture sensors of the type of Xsens MVN Awinda were used. The sensors were attached to various locations on the subject's body, including above the sternum, shoulder blade, upper arm, forearm, and back of the hand (Figure 4). The subjects undergoing the rehabilitation exercise were instructed to perform 30 daily living activities with both sides of their upper limb, performing each activity three times [37]. The kinematic measures used to evaluate the performance of the second prototype in this study included shoulder flexion/extension, shoulder abduction/adduction, shoulder internal/external rotation, elbow flexion/extension, elbow pronation/supination, wrist flexion/extension, wrist abduction/adduction and wrist pronation/supination.

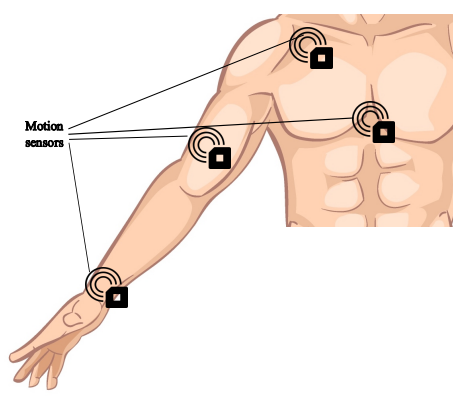


Figure 4. Motion sensors attached to the patient's body for data capture [38].

These measures provided a comprehensive evaluation of the system's ability to accurately stream and visualize the upper limb movements of stroke patients during rehabilitation activities.

The second prototype features a user interface that displays live-streamed data from the patient alongside the standard deviation (SD) of data from healthy subjects used as a reference for the targeted performance of the patient during rehabilitation exercises. Figure 5 illustrates an example of a graph of shoulder flexion/extension for a stroke patient performing the task of reaching and grasping a glass of water, drinking for three seconds, and returning the glass to its initial position. This prototype was initially evaluated by a panel of four rehabilitation experts to collect end users' feedbacks.

Results of the evaluation of the second prototype:

The experts identified two main requirements for the system improvement. The first issue pertained to the difficulty of comparing the performance of the stroke patient and healthy subject with the naked eye while the patient data graph was updating in real-time. The experts recommended implementing visual aids to illustrate the comparison and facilitate more accurate analysis of the patient's performance. The second requirement highlighted by the panel of experts was the need for control over the pace of data display. This would enable the user to slow down or pause the data stream to obtain thorough analysis and relevant information extraction. This could also allow the option to speed up the stream to keep pace with real-time updates. This level of control over the pace of data display would provide greater flexibility and usefulness for the rehabilitation therapist.

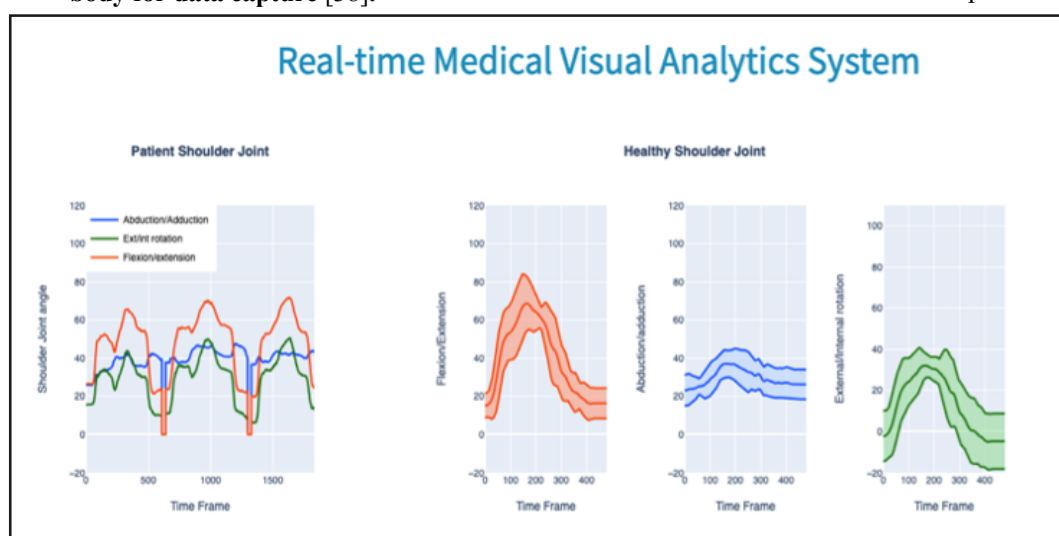


Figure 5. Second prototype: visualization of real-time data flow of the shoulder joint angle of stroke patient Vs healthy subjects

Third Iteration

Based on the feedback received from the evaluation of the second prototype, additional requirements were integrated to improve its functionality. To address the challenge of comparing the performance of patients with healthy subjects, the real-time data of the patients were superimposed with the pre-recorded data and the SD of healthy subjects. This could provide better visual comparison of the patient's performance during rehabilitation activities. The range of motion of the patient and the healthy subjects were calculated and displayed through pie charts in the dashboard. The user interface also displayed important information about the patient, including their ID, age, gender, the time since the brain stroke took place, and an indication about which limb is affected by the stroke. A description of the current exercise instructions was also included in the dashboard. Figure 6 shows an example of the user interface, displaying live stream data of a patient performing an activity of daily living with upper limb data of the monitored patient mapped to healthy subjects' data.

Results of the evaluation of the third prototype:

In order to evaluate the third version of the prototype, a group of 16 rehabilitation practitioners used the dashboard of the framework to monitor the streamed data used in the previous iterations. The subjects were requested to map the displayed data on the kinematics of the patient's limb and interpret his/her progress with reference to healthy subjects. The feedback of experience was collected by means of questionnaire about the appreciation of the visualization system by

the users. The analysis of the collected data revealed that the rehabilitation practitioners considered that the system allowed them to interpret the visualized data. They also declared that the system is beneficial for monitoring patients receiving rehabilitation therapy in home settings. Additionally, they expressed their desire to use the final version of the system in their protocols for patient care.

5. CONCLUSION AND RESEARCH PERSPECTIVES

The objective of this research was to develop and evaluate a unified real-time visual analytics framework for remote patient monitoring. The development process of the proposed framework was carried out in three iterations and following the principle of User Centered Design (UCD). The system was applied for monitoring post-stroke survivors receiving neuromotor rehabilitation treatment. In order to comply with the principle of involving end-users since the design stage, each iteration was evaluated, and the collected feedbacks were analyzed and used for the refinement of the framework. The final prototype of the developed framework allowed a group of target users to monitor patient data during the practice of rehabilitation exercises. Real-time monitoring is not to be limited to the visualization of the current health data of patients. It should also allow predicting the evolution of patients' health during and after the rehabilitation sessions. This aspect will be addressed in the future of this research by the development of an intelligent module based on machine learning for health prediction.

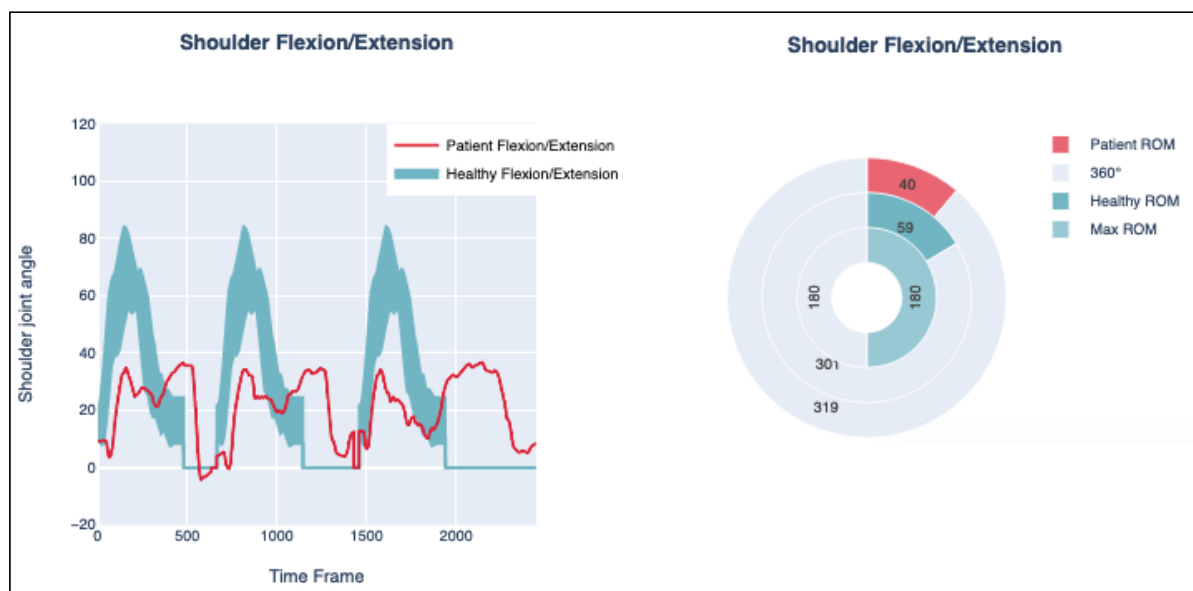


Figure 6. Third prototype: Shoulder's Flexion/Extension and Range of motion related to the current exercise for both patient (red) and the average of healthy subjects

REFERENCES

- [1] J. J. Caban and D. Gotz, "Visual analytics in healthcare - opportunities and research challenges," *J. Am. Med. Informatics Assoc.*, vol. 22, no. 2, pp. 260–262, 2015, doi: 10.1093/jamia/ocv006.
- [2] N. Kamal, "Big Data and Visual Analytics in Health and Medicine: From Pipe Dream to Reality," *J. Heal. Med. Informatics*, vol. 05, no. 05, pp. 998–1000, 2014, doi: 10.4172/2157-7420.1000e125.
- [3] J. W. Tukey, "Exploratory Data Analysis," in *The Concise Encyclopedia of Statistics*, New York, NY: Springer New York, 2008, pp. 192–194.
- [4] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, *Mastering The Information Age – Solving Problems with Visual Analytics*. Eurographics Association, 2010.
- [5] P. C. W. John Dill, Rae Earnshaw, David Kasik, John Vince, *Expanding the Frontiers of Visual Analytics and Visualization*. Springer Science & Business Media, 2012.
- [6] "Apache Kafka." <https://kafka.apache.org/>.
- [7] "Dash." <https://dash.plotly.com/dash-html-components>.
- [8] S. Park and A. D. Flaxman, "Understanding data use and preference of data visualization for public health professionals : A qualitative study," *Public Health Nurs.*, no. December 2020, pp. 1–11, 2021, doi: 10.1111/phn.12863.
- [9] G. Harerimana, S. Member, and B. Jang, "Health Big Data Analytics : A Technology Survey," *IEEE Access*, vol. 6, pp. 65661–65678, 2018, doi: 10.1109/ACCESS.2018.2878254.
- [10] K. Sawhney and D. F. Sittig, "Information Overload and Missed Test Results in EHR-based Settings," vol. 173, no. 8, 2014, doi: 10.1001/2013.jamainternmed.61.Information.
- [11] P. West, M. Van Kleek, R. Giordano, M. J. Weal, and N. Shadbolt, "Common Barriers to the Use of Patient-Generated Data Across Clinical Settings," *CHI Conf. Hum. Factors Comput. Syst.*, pp. 1–13, 2018.
- [12] M. Musterman *et al.*, "Big Data in Health Care : What Is So Different About Was ist so anders am Neuroenhancement ?," *Data Inf. Manag.*, vol. 2, no. 3, pp. 175–197, 2018, doi: 10.2478/dim-2018-0014.
- [13] S. Kliment, F. Information, and C. Technologies, "Big Data Analytics in Medicine and Healthcare Abstract.," *J. Integr. Bioinform.*, pp. 1–5, 2018, doi: 10.1515/jib-2017-0030.
- [14] K. K. Khedo *et al.*, "Health Data Analytics: Current Perspectives, Challenges, and Future Directions," in *IoT and ICT for Healthcare Applications*, N. Gupta and S. Paiva, Eds. Cham: Springer International Publishing, 2020, pp. 117–151.
- [15] A. Seals *et al.*, "Are They Doing Better In The Clinic Or At Home?: Understanding Clinicians' Needs When Visualizing Wearable Sensor Data Used In Remote Gait Assessments For People With Multiple Sclerosis," 2022, doi: 10.1145/3491102.3501989.
- [16] D. J. Feller *et al.*, "A visual analytics approach for pattern-recognition in patient-generated data," *J. Am. Med. Informatics Assoc.*, vol. 25, no. 10, pp. 1366–1374, 2018, doi: 10.1093/jamia/ocy054.
- [17] A. L. Vu, "Visual Analytics: Identifying Informative Temporal Signatures in Continuous Cardiac Monitoring Alarms from a Large Hospital System," Diss. The Ohio State University, 2017.
- [18] A. Dagliati *et al.*, "A dashboard-based system for supporting diabetes care," *J. Am. Med. Informatics Assoc.*, vol. 25, no. 5, pp. 538–547, 2018, doi: 10.1093/jamia/ocx159.
- [19] E. V. Hobson *et al.*, "The TiM system : developing a novel telehealth service to improve access to specialist care in motor neurone disease using user-centered design," *Amyotroph. Lateral Scler. Front. Degener.*, vol. 19, no. 5–6, pp. 351–361, 2018, doi: 10.1080/21678421.2018.1440408.
- [20] L. Griffin *et al.*, "Creating an mHealth App for Colorectal Cancer Screening : User-Centered Design Approach," *JMIR Hum. factors*, vol. 6, 2019, doi: 10.2196/12700.
- [21] A. Raghu, D. Praveen, D. Peiris, L. Tarassenko, and G. Clifford, "Engineering a mobile health tool for resource-poor settings to assess and manage cardiovascular disease risk : SMARThealth study," *BMC Med. Inform. Decis. Mak.*, pp. 1–15, 2015, doi: 10.1186/s12911-015-0148-4.
- [22] U. Backonja, S. C. Haynes, K. K. Kim, and K. K. Kim, "Data Visualizations to Support Health Practitioners ' Provision of Personalized Care for Patients With Cancer and Multiple Chronic Conditions : User-

- Centered Design Study,” *JMIR Hum. factors*, vol. 5, pp. 1–20, 2018, doi: 10.2196/11826.
- [23] C. L. Petersen *et al.*, “Using Natural Language Processing and Sentiment Analysis to Augment Traditional User-Centered Design : Development and Usability Study,” *JMIR mHealth uHealth*, vol. 8, pp. 1–13, 2020, doi: 10.2196/16862.
- [24] L. David, G. Bouyer, and S. Otmame, “Towards a low-cost interactive system for motor self-rehabilitation after stroke,” *Int. J. Virtual Real.*, vol. 17, no. 2, pp. 40–45, 2017, doi: 10.20870/IJVR.2017.17.2.2890.
- [25] T. Caporaso, S. Grazioso, D. Panariello, G. Di Gironimo, and A. Lanzotti, “Understanding the Human Motor Control for User-Centered Design of Custom Wearable Systems: Case Studies in Sports, Industry, Rehabilitation,” in *Design Tools and Methods in Industrial Engineering*, 2020, pp. 753–764.
- [26] C. L. Osborne, S. B. Juengst, and E. E. Smith, “Identifying user-centered content , design , and features for mobile health apps to support long-term assessment , behavioral intervention , and transitions of care in neurological rehabilitation: An exploratory study,” *Br. J. Occup. Ther.*, 2020, doi: 10.1177/0308022620954115.
- [27] M. Wentink *et al.*, “How to improve eRehabilitation programs in stroke care? A focus group study to identify requirements of end-users,” *BMC Med. Inform. Decis. Mak.*, vol. 3, pp. 1–11, 2019.
- [28] M. Boumrah, S. Garbaya, and A. Radgui, “Real-time visual analytics for in-home medical rehabilitation of stroke patient — systematic review,” *Med. Biol. Eng. Comput.*, pp. 889–906, 2022, doi: 10.1007/s11517-021-02493-w.
- [29] J. Marz, Nathan; Warren, *Big data : principles and best practices of scalable real-time data systems*. Simon and Schuster, Shelter Island, NY : Manning, 2015. - 308 p, 2015.
- [30] S. Miksch and W. Aigner, “A matter of time: Applying a data-users-tasks design triangle to visual analytics of time-oriented data,” *Comput. Graph.*, vol. 38, no. 1, pp. 286–290, 2014, doi: 10.1016/j.cag.2013.11.002.
- [31] G. L. Dimaguila, K. Gray, and M. Merolli, “Person-generated health data in simulated rehabilitation using kinect for stroke: Literature review,” *JMIR Rehabil. Assist. Technol.*, vol. 20, no. 5, 2018, doi: 10.2196/rehab.9123.
- [32] B. R. Hiranman, M. C. Viresh, and C. K. Abhijeet, “A Study of Apache Kafka in Big Data Stream Processing,” *2018 Int. Conf. Information, Commun. Eng. Technol. ICICET 2018*, pp. 1–3, 2018, doi: 10.1109/ICICET.2018.8533771.
- [33] H. Jafarpour, R. Desai, and D. Guy, “KSQL: Streaming SQL engine for Apache Kafka,” *Adv. Database Technol. - EDBT*, vol. 2019-March, pp. 524–533, 2019, doi: 10.5441/002/edbt.2019.48.
- [34] Apache Software Foundation, “Kafka 3.4 Documentation.” <https://kafka.apache.org/documentation/#performance>.
- [35] N. J. Jarque-Bou, M. Vergara, J. L. Sancho-Bru, V. Gracia-Ibáñez, and A. Roda-Sales, “A calibrated database of kinematics and EMG of the forearm and hand during activities of daily living,” *Sci. Data*, vol. 6, no. 1, pp. 1–11, 2019, doi: 10.1038/s41597-019-0285-1.
- [36] S. Schellenberger *et al.*, “A dataset of clinically recorded radar vital signs with synchronised reference sensor signals,” *Sci. Data*, vol. 7, no. 1, pp. 1–11, 2020, doi: 10.1038/s41597-020-00629-5.
- [37] G. Averta *et al.*, “U-Limb: A multi-modal, multi-center database on arm motion control in healthy and post-stroke conditions,” *Gigascience*, vol. 10, no. 6, p. giab043, Jun. 2021, doi: 10.1093/gigascience/giab043.
- [38] A. Schwarz *et al.*, “Characterization of stroke-related upper limb motor impairments across various upper limb activities by use of kinematic core set measures,” *J. Neuroeng. Rehabil.*, vol. 19, no. 1, pp. 1–18, 2022, doi: 10.1186/s12984-021-00979-0.

On Unguided Automatic Colorization of Monochrome Images

Andrzej Śluzek^[0000-0003-4148-2600]

Warsaw University of Life Sciences - SGGW
Institute of Information Technology
ul. Nowoursynowska 159, Bld. 34
02-776, Warsaw, Poland
andrzej_sluzek@sggw.edu.pl

ABSTRACT

Image colorization is a challenging problem due to the infinite RGB solutions for a grayscale picture. Therefore, human assistance, either directly or indirectly, is essential for achieving visually plausible colorization. This paper aims to perform colorization using only a grayscale image as the data source, *without* any reliance on metadata or human hints. The method assumes an (arbitrary) *rgb2gray* model and utilizes a few simple heuristics. Despite probabilistic elements, the results are visually acceptable and repeatable, making this approach feasible (e.g. for aesthetic purposes) in domains where only monochrome visual representations exist. The paper explains the method, presents exemplary results, and discusses a few supplementary issues.

Keywords

Image colorization, decolorization, *rgb2gray* models, visual plausibility, color models.

1. INTRODUCTION & MOTIVATION

Image colorization, i.e. reconstructing color images from monochrome ones, is an ill-posed problem due to the infinite number of RGB solutions for a grayscale picture. Nonetheless, this topic holds notable practical and commercial significance, particularly in the restoration of historical photos and movies being the primary application, e.g. [Zeg21], [Sal22].

In the past two decades, many papers have proposed diverse algorithms for reconstructing color images or movies from their monochrome counterparts. Initially, the methods were mainly semi-automatic. Users would provide exemplary images to guide the algorithm in coloring images with similar contents and contexts, primarily using similarly textured patches, e.g. [Iro05] and [Gup12]. Alternatively, monochrome images can be manually “scribbled” to indicate approximate colors over a number of significant locations, e.g. [Lev04], [Lag08].

More recently, advanced machine learning has enabled fully automated image colorization, with coloring patterns learned from relevant images rather than human-provided hints.

Typically, the patterns are derived from images of specific domains, e.g. [Des15], [Hwa16], [Zha16].

A more comprehensive system is described in [Iiz16], which learns scene recognition, local priors, and mid-level features from nearly 2.5 million training images. The results are impressive on test images of scenes (if their semantics are correctly recognized). Existing commercial systems (e.g. [Sal22]) generally follow the same concepts.

In an alternative approach, machine learning can be used to identify colorization statistics (instead of direct coloring), as in [Des15] and [Roy17]. Automatic image colorization across multiple domains (transfer learning) is more challenging, and [Lee22] is the first work with limited but convincing results.

In summary, all the methods mentioned above (and many other approaches not discussed here) are assisted by humans, either by providing colorization hints or relevant training data for ML algorithms. Therefore, the proposed objective of this paper seems slightly audacious (if not impossible). Our intention is to develop a mechanism for unguided automatic image colorization *without* additional metadata, assistance, learning processes, or domain identification. In other words, we aim to create an acceptable colored counterpart using only a grayscale image as the data source. By “acceptable,” we mean visually attractive results that are statistically repeatable and deliver convincingly rich sensations of colors (excluding pseudo-coloring, as in thermographic cameras).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

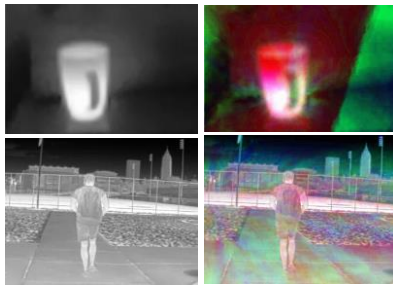


Figure 1. Examples of visually convincing colorizations of infrared images.

In domains of visual-frequency grayscale images, such a problem is rather marginal, but there are areas where only single-channel visualizations actually exist, such as IR/UV/US/MRI/X-ray images. In such cases, we would like to see hypothetical RGB versions of those "gray worlds" for various reasons, even if it is only for aesthetic purposes (see Fig. 1).

Section 2 of the paper discusses a number of assumptions and models adopted in the proposed solution. Further implementation details of the developed algorithms are included in Section 3. Section 4 presents diverse examples of obtained results, with corresponding explanations. The final Section 5 contains conclusions, discusses some supplementary issues, and highlights directions for future work.

2. PROPOSED METHODOLOGY

Using *rgb2gray* models in colorization

Colorization methods generally assume that grayscale image values represent the luminance channel of the colored outputs, requiring reconstruction of only two chrominance channels. However, few papers on image colorization consider the opposite question: *how the original RGB image (real or hypothetical) was decolorized to obtain a grayscale image*.

Standard RGB-to-grayscale models (YUV and YIQ) apply linear functions of primary colors:

$$Y = k_R R + k_G G + k_B B \quad (1)$$

where $k_R = 0.299$, $k_G = 0.587$ and $k_B = 0.114$ (or $k_R = 0.2126$, $k_G = 0.7152$, $k_B = 0.0722$).

Other *rgb2gray* models with arbitrarily assumed k_R , k_G and k_B coefficients (subject to $k_R + k_G + k_B = 1$) produce alternative monochrome images (see Figs 2b,c,d) from which various re-colorizations can be hypothetically reconstructed (Figs 2e,f,g).

Therefore, in the proposed colorization scheme we first assume that:

Monochrome images are derived from (real or hypothetical) color images by an *rgb2gray* model with arbitrarily assumed k_R , k_G and k_B coefficients.

Such an assumption is justified because for problems with only hypothetical existence of color images (as in Fig.1), any *rgb2gray* model can be assumed, as long as the colorization results are visually appealing.

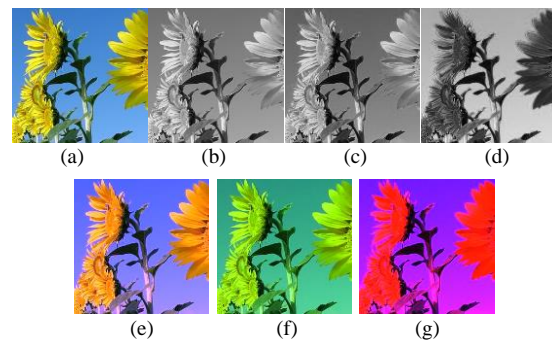


Figure 2. B/w versions of (a) by various *rgb2gray* models (b, c, d). Perfect re-colorizations of (b) using alternative *rgb2gray* models (e, f, g).

Colorization of pixels

2.2.1. Individual pixels

Assume that colorized monochrome images are obtained using known *rgb2gray* model.

Given a single (x,y) pixel with $I(x,y)$ intensity from **[0:255]** discrete range, its colored counterpart should approximately satisfy (subject to color discretization):

$$I(x,y) \approx k_R R(x,y) + k_G G(x,y) + k_B B(x,y) \quad (2)$$

Since the adopted *rgb2gray* model might be inaccurate, we can use reduced numbers of colors (e.g. 32 levels instead of 256) without affecting significantly Eq. 2.

Eventually, all $32^3 = 32768$ colors are assigned to various intensities, based on the smallest error in Eq. 2. The numbers of colors assigned to a single intensity are non-uniformly distributed. Fig. 3 contains the actual numbers for two exemplary *rgb2gray* models: **[0.299, 0.587, 0.114]** and **[0.69, 0.12, 0.19]**.

It shows the widest selection of color options for mid-range intensities, with the numbers gradually dwindling for darker/lighter values to, eventually, a deterministic choice for extremely dark/light intensities. With no prior information provided, all available colors should be considered equally probable, i.e. $p(C_j | I) = 1/N$, where N indicates the number of colors assigned to I value.

Fig. 4 displays the pool of colors (under two *rgb2gray* models) for selected values.

2.2.2. Neighboring pixels

If a pixel at (x,y) has an intensity of I but has not been assigned a color yet, the probabilities of colors that could be assigned to I should be influenced by the

presence of a neighboring pixel with an intensity I_1 and its already assigned C_{I_1} color.

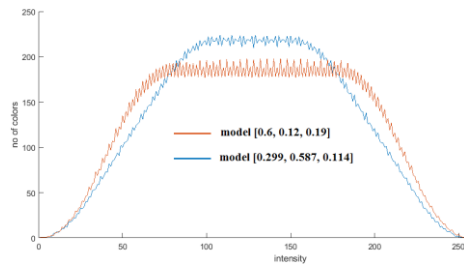


Figure 3. Numbers of RGB colors (out of the total number of 32768) assigned to intensities in two *rgb2gray* models.

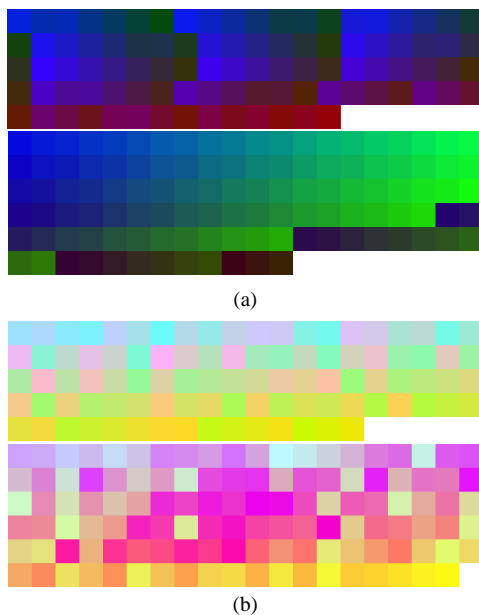


Figure 4. Colors assigned to (a) 46 and (b) 208 intensities in the [0.299, 0.587, 0.114] and [0.69, 0.12, 0.19] *rgb2gray* models. Note the inconsistencies with human perception of brightness in the second model.

Therefore, we propose a simple heuristic rule:

The greater the difference in brightness between adjacent pixels, the higher the likelihood that their assigned colors will also differ significantly.

Let's consider the pool of colors available for the intensity level I : $\{C_I^1, C_I^2, \dots, C_I^N\}$. They are arranged in a monotonically increasing order based on their distance from the color C_{I_1} of the adjacent pixel.

Fig. 4b shows the ordered lists for $I = 208$, assuming that a neighboring pixel of an intensity $I_1 = 46$ was assigned the RGB color $C_{I_1} = [20, 42, 137]$.

Then, we select the color C_I from the list using a uniform distribution which is defined over certain

fragments of the list, depending on the difference in intensity levels $abs(I - I_1)$. We tested several options, but eventually implemented a heuristic approach where the C_I color is randomly selected from the $\langle C_I^{i_{\min}}, C_I^{i_{\max}} \rangle$ range specified by the indexes i_{\min} and i_{\max} given by Eq. 3.

$$\begin{aligned} i_{\min} &= \max(1, \text{round}(\alpha \cdot \text{diff})), \\ i_{\max} &= \min(N, \text{round}(\text{diff})), \\ \text{where } \text{diff} &= N \cdot \min\left(1, \frac{abs(I - I_1)}{128}\right) \end{aligned} \quad (3)$$

In some cases, the choice is deterministic (formally represented by the $i_{\max} \leq i_{\min}$ condition), e.g.:

- White/black pixels are always colored using the brightest/darkest color.
- If neighboring pixels have the same brightness their colors are also the same (this may later change later as discussed below).

In the implementation of the method, images are colored incrementally (see details in Section 3) and it may happen that an uncolored pixel has several already colored neighbors. Then, the color selection can be performed several times for that pixel, and the final choice is a weighted sum of the colors obtained from all colored neighbors.

$$C_I = \frac{1}{M} \sum_{j=1}^M C_{I_j} \quad \text{where } M = 1, 2, 3 \text{ or } 4 \quad (4)$$

In this way, we can get more colors than a limited pool of 32768 colors initially assumed in Section 2.2.1.

3. IMPLEMENTATION DETAILS

Initialization procedure

Colorization of monochrome images is performed incrementally, starting from a number of initially colored pixels. In the simplest case, it can be even a single pixel.

The proposed options that do not require human assistance for the initial list (queue) of colored pixels are:

- The darkest/brightest pixel of the image. Because its color is usually deterministic (see Section 2), no human assistance is needed.
- As in (a), but the list contains all darkest or brightest pixels (or both).

Image colorization

The image colorization method is actually a randomized variant of a popular *flood-fill* algorithm (in the queue-based version).

We randomly select a pixel from the current list L of colored pixels and colorize its uncolored neighbors using the method outlined in Section 2.2.2. This way,

the colored patch grows randomly, avoiding unnecessary regularities in the colorization process (see Fig. 5).

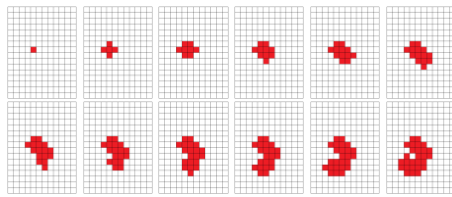


Figure 5. Random growth of the colored patch.

4.3.1. Final touch ups

As highlighted in Section 2.2.1, pixels are initially colored using a limited set of **32768** colors.

However, additional colors can be introduced when the colors assigned to pixels are averaged by Eq. 4 (i.e. a colorized pixel has more already colored neighbors).

We further increase the diversity of colors by projecting them on planes of *rgb2gray* models.

Given a pixel with the original I intensity and the assigned color C_I , we find its closest counterpart $C_{I_{mod}}$ on the selected *rgb2gray* plane (Eqs 1 and 2).

Such modifications may not noticeably change colorized images when projected onto the plane of the original *rgb2gray* model (Figs 6a and 6b), but they can enhance the visual plausibility of colorized images when projected onto the YUV plane (Fig. 6c).

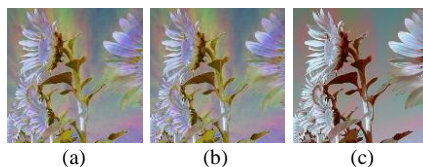


Figure 6. The colorization results (a) and their projection on the original *rgb2gray* plane (b) and on the YUV plane (c).

4. EXPERIMENTAL RESULTS

The proposed methodology incorporates several heuristics, arbitrary assumptions, and probabilistic schemes. Therefore, evaluating its performance and practicality requires extensive experimentation. Unfortunately, popular image similarity metrics cannot be used because the ground-truth color images are assumed to be nonexistent.

Therefore, we evaluate the results using subjective criteria as a preliminary approach. We consider two evaluation criteria:

- Visual plausibility, without considering domain-based realities (e.g., grass of any color can be accepted if convincingly rendered).
- Repeatability under the same the same *rgb2gray* model (see Section 2) and the same initialization mode (see Section 3).

Due to page limitations, we include only a selection of results in this section to illustrate the presented conclusions. For example, we only consider three *rgb2gray* models. A more extensive summary of the results is provided in the supplementary materials.

Datasets

We use a diversified collection of monochrome visual-frequency, IR, and other images. For the visual-frequency images, we show their *ground-truth* colors (if available) for information purposes only and do not use them to evaluate colorization quality.

The images are sourced from personal resources and public databases. As no benchmarks (to the best of our knowledge) exist for the discussed topic, we have selected the databases somewhat arbitrarily. The visual-frequency images (converted to monochrome) mainly come from well-known UKBench and SUN databases. The IR images are primarily selected from CAMEL [Geb18] and SMD [Pra17] datasets, while examples of other non-visual images (e.g., MRI, X-ray, etc.) come from various sources.

Parameters of colorization

Altogether, 4851 *rgb2gray* models were considered, corresponding to a sampling of the model coefficients with a 0.001 increment, but in the end, only three models were selected for the experiments reported in the paper, namely:

- $k_R = 0.299$, $k_G = 0.587$ and $k_B = 0.114$, i.e. the standard YUV model accurately converting colors into a subjective perception of brightness.
- $k_R = 0.301$, $k_G = 0.387$ and $k_B = 0.302$, which is similar to a simple mean of primary colors.
- $k_R = 0.69$, $k_G = 0.12$ and $k_B = 0.19$, a model with deliberately unrealistic coefficients.

As discussed earlier, three initialization variants are considered, i.e. (a1) a single darkest pixel, (a2) a single brightest pixel and (b) all darkest and brightest pixels. By considering three initialization variants, a single run of the colorization algorithm can produce *nine* results (all possible combinations of *rgb2gray* models and initialization options).

Visual plausibility

The YUV-based *rgb2gray* model produces the best plausibility in the sense that all details from grayscale images are equally clearly seen (sometimes even overexposed) in their colored counterparts. This is not surprising because this model provides the best compatibility between colors and their brightness perception by human eyes.

Results of the second model are still acceptable, but not all details of the original contents can be as clearly seen as in the first model.

For the third *rgb2gray* model, colors are usually assigned to intensities in such a way that human eyes can hardly identify the image details.

Fig. 7 shows exemplary colorization results for selected IR and visual-frequency images by the three models. It can be noticed that the richness of colors is satisfactory in all three models.

Altogether, we can preliminarily conclude that the plausibility of colorization depends strongly on the selected *rgb2gray* model; the more “natural” the model, the better.

Plausibility by repeatability

We found that the plausibility of colorization can be improved by averaging several runs of the algorithm with the same *rgb2gray* model and initialization.

Surprisingly, such averaged images do not converge to grayscale, as intuitively expected (since colors are assigned to intensities using probabilistic heuristics with uniform distributions).

Instead, as seen in Fig. 8, we get visually attractive images with diversified coloristics (although the colors are usually less saturated than those from individual runs of the algorithm).

What is even more interesting, the averaged images obtained with the same *rgb2gray* model (regardless of the initialization) are usually quite similar. Examples are provided in Fig. 9.

Thus, we cautiously hypothesize (and preliminary theoretical results seem to confirm this hypothesis) that unique image colorizations for the selected *rgb2gray* model might objectively exist. Nevertheless, further experimental and theoretical research on this topic is needed.

5. CONCLUDING REMARKS

In this paper, we attempted to handle the ill-posed problem of colorizing grayscale images without any (direct or indirect) human assistance. We only assume that a hypothetical decolorization model is given. Initially, we use a limited number of 32^3 colors, but the algorithm can subsequently use the full sRGB gamut of colors.

The colorization process is performed using a randomized *flood-fill* method, starting from the darkest/brightest pixels for which the choice of color is deterministic. Subsequently, simple probabilistic heuristics are applied to incrementally colorize other pixels.

In spite of the heavy presence of randomizing factors, the results are surprisingly repeatable, depending on the adopted *rgb2gray* model and (to a rather insignificant extent) on the applied initialization mode. We even cautiously hypothesize that for the adopted *rgb2gray* model, unique optimum colorizations may exist for monochrome images (possibly with some additional limitations).



Figure 7. Selected b/w images and their colorized samples for three *rgb2gray* models. For visual-frequency images, the original color versions are also included.

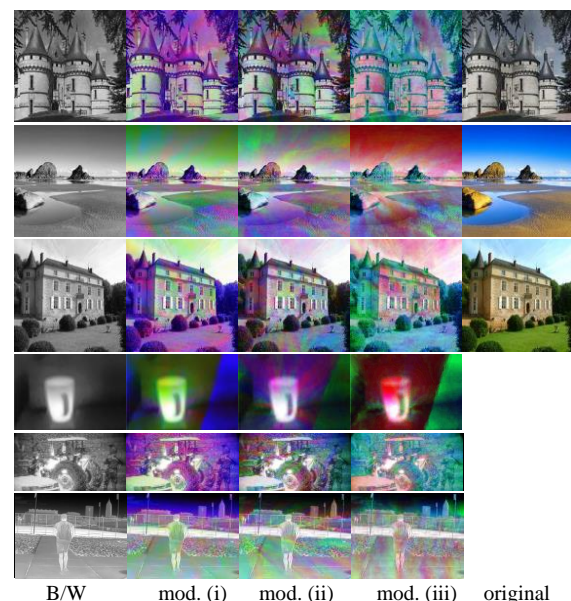


Figure 8. Results for Fig. 7 with colors averaged over 10 runs (with the same model and initialization mode).

The method is primarily intended for colorizing grayscale images for which there are no physical color counterparts. In other words, we aim to produce convincingly rich colorized versions of “gray worlds”. This may be required for various reasons, even if only aesthetic.

Nevertheless, many visual-frequency images are used in the experimental work to better highlight the

differences between our approach and the “traditional” re-colorization.

In the future work we intend to focus on the following aspects of the project:

- A formal analysis of the statistical properties of the method (including alternative probability distributions used in the adopted heuristics).
- The development of metrics for objectively estimating the quality of colorization results (e.g. [Has03]).
- Extension of the method to unguided colorization of monochrome movies.

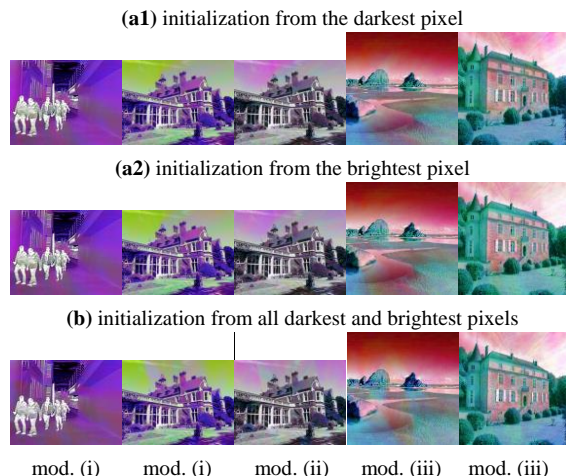


Figure 9. Colorizations for various initializations using the same *rgb2gray* models. The results are averaged from 75 runs of the algorithm.

6. REFERENCES

- [Des15] Deshpande, A., Rock, J., Forsyth, D. Learning large-scale automatic image colorization, 2015 IEEE Int. Conference on Computer Vision (ICCV), pp. 567-575, 2015. DOI: 10.1109/ICCV.2015.72
- [Geb18] Gebhardt, E., Wolf, M. CAMEL dataset for visual and thermal infrared multiple object detection and tracking, IEEE Int. Conf. on Advanced Video and Signal-based Surveillance (AVSS), pp. 1-6, 2018. DOI: 10.1109/AVSS.2018.8639094
- [Gup12] Gupta, R.K., Chia, A., Rajan, D., Ng, E.S., Zhiyong, H. Image colorization using similar images. In: Proc. 20th ACM International Conference on Multimedia MM’12, pp. 369–378, 2012. DOI: 10.1145/2393347.2393402
- [Has03] Hasler, D., Suesstrunk, S.E. Measuring colorfulness in natural images. Proc. SPIE 5007, Human Vision and Electronic Imaging, pp. 87-95, 2003. DOI: 10.1117/12.477378.
- [Hwa16] Hwang, J., Zhou, Y. Image colorization with deep convolutional neural networks. Stanford U. Tech. Rep., http://cs231n.stanford.edu/reports/2016/pdfs/219_Report.pdf, 2016.
- [Iiz16] Iizuka, S., Simo-Serra, E., Ishikawa, H. Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM Trans. Graph., vol. 35(4), Article 110, 2016. DOI: 10.1145/2897824.2925974
- [Iro05] Irony, R., Cohen-Or, D., Lischinski, D. Colorization by example. In: Eurographics Symposium on Rendering (eds K. Bala and Ph. Dutre), 2005. DOI: 10.2312/EGWR/EGSR05/201-210
- [Lag08] Lagodzinski, P., Smolka, B. Digital image colorization based on probabilistic distance transformation. LNCS 5197 (CIARP 2008) (eds. J. Ruiz-Shulcloper and W.G. Kropatsch), pp. 626–634, 2008.
- [Lee22] Lee, H., Kim, D., Lee, D., Kim, J., Lee, J. Bridging the domain gap towards generalization in automatic colorization. LNCS vol. 13677 (ECCV 2022), (eds Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T.), pp. 527-543, 2022. DOI: 10.1007/978-3-031-19790-1_32
- [Lev04] Levin, A., Lischinski, D., Weiss, Y. Colorization using optimization. In: ACM SIGGRAPH 2004 Papers (ed. J. Marks), pp.689–694, 2004.
- [Pra17] Prasad, D.K., D. Rajan, D., Rachmawati, L., Rajabaly, E., Quek, C. Video processing from electro-optical sensors for object detection and tracking in maritime environment: A survey. IEEE Trans. Intelligent Transportation Systems, vol. 18 (8), pp. 1993-2016, 2017. DOI: 10.1109/TITS.2016.2634580
- [Roy17] Royer, A., Kolesnikov, A., Lampert, Ch.H. Probabilistic image colorization. Proc. BMVC’17 Conf., pp. 85.1-85.12, 2017. DOI: 10.5244/C.31.85
- [Sal22] Salmona, A., Bouza, L., Delon, J. DeOldify: A review and implementation of an automatic colorization method. Image Processing On Line, vol. 12, pp. 347–368, 2022. DOI: 10.5201/ipol.2022.403
- [Zeg21] Žeger, I., Grgic, S., Vuković, J., Šišul, G. Grayscale image colorization methods: Overview and evaluation. IEEE Access, vol. 9, pp. 113326-113346, 2021. DOI: 10.1109/ACCESS.2021.3104515
- [Zha16] Zhang, R., Isola, P., Efros, A.A. Colorful image colorization. LNCS vol. 9907 (ECCV 2016), (eds Leibe, B., Matas, J., Sebe, N., Welling, M.), pp. 649–666, 2016. DOI: 10.1007/978-3-319-46487-9_40

Justice expectations related to the use of CNNs to identify CSAM. Technological interview.

Wojciech Oronowicz-Jaśkowiak

Faculty of Computer Science
Polish-Japanese Academy
of Information Technology

Koszykowa 86
Poland, Warsaw 02-008

oronowiczjaskowiak@pjwstk.edu.pl

Piotr Wasilewski

Intelligent Systems Laboratory
Systems Research Institute
Polish Academy of Sciences

Newelska 6

Poland, Warsaw 01-447

piotr.wasilewski@ibspan.waw.pl

Mirosław Kowaluk

Faculty of Mathematics,
Informatics and Mechanics
University of Warsaw

Banacha 2
Poland, Warsaw 02-008

kowaluk@mimuw.edu.pl

ABSTRACT

A technological interview was conducted with representatives of the judiciary to determine their expectations and beliefs related to the technological solution (involving detection of child sexual abuse materials using CNNs), being developed. The obtained results lead to the following conclusions: 1. Representatives of the judiciary recognize the advantages of the technological solution being created in the form of accelerating the work of experts and minimizing the risk of mistakes. 2. Representatives of the judiciary see the limitations of the technological solution being created in the form of the inability to replace court experts and emphasize that it also depends on the stage of the case. 3. The selection of pornographic materials from a specific set for later verification by a forensic expert is of the greatest importance. Coded excerpts of the participants' statements in the form of raw results have been published in the OSF repository (DOI: 10.17605/OSF.IO/RU7JX).

Keywords

CSAM; computer vision; forensic sexology

1. INTRODUCTION

Child sexual abuse material (CSAM) is widely distributed online. The *Directive of the European Parliament and of the Council on combating the sexual abuse and sexual exploitation of children and child pornography* [EU00a] provides the following definition of child pornographic material: “(i) any material that depicts a child engaging in real or simulated sexually explicit conduct; or (ii) any depiction of the sexual organ of a child for primarily sexual purposes; or (iii) any material depicting a child-looking person engaged in real or simulated sexually explicit conduct and depicting the sexual organs of child-looking persons for primarily sexual purposes; or (iv) realistic images of a child engaged in sexually explicit conduct or actual images of a child's genital organs, whether or not they exist, for primarily sexual purposes”.

In the field of experts' specializations, it is necessary to determine the degree of sexualization of the victims, indicate whether, from the sexual perspective, the content can be considered pornographic, indicate whether the content includes such materials in which animals were used and/or cruelty towards the victims, and estimation of the age of the persons presented in the materials [Car00a,

Qua00a]. The tasks set for experts are detailed and require many hours of analyses of materials secured during prosecution proceedings. Depending on the number of secured materials, such an analysis may take from several to several dozen hours for one criminal case. The research underlying the opinion, however, must be conducted fairly, as the further course of the matter may depend on the conclusions contained in the opinion [Moz00a]. The above causes that the judicial and sexological opinions on possible pornographic materials with the participation of children are burdensome in terms of time and weight of the opinions issued.

For the above reasons, solutions are sought that would ease the burden on court experts in the field of sexology by automating some of their classifications. One of them is the use of machine learning methods to create a technological solution that allows some classification of pornographic content, with the use of machine learning [Vit00a, Wer00a]. However, before such solutions are introduced into practice, it is important to conduct a technological interview to determine the preferred functionalities of the solution and the expectations of the judiciary about the solution being created.

2. METHOD

A questionnaire was created containing statements and questions related to the functionality of the created technological solution. The questions posed to the participants of the study were open-ended and closed-ended questions. Invitation to participate in the research was sent to 30 randomly selected prosecutor's offices, courts and provincial police headquarters. Due to the epidemic limitations, the survey was conducted using one of the three methods, i.e. a telephone conversation, an Internet survey and interviews conducted at the participant's workplace. The results were analysed in terms of quantity and quality. The SPSS 25 statistical software was used for quantitative analysis, while the MAXQDA 2022 software was used for qualitative analysis.

3. RESULTS

The minimum size of both groups was estimated on the basis of similar studies [San00a]. It was assumed that the size of the groups would be necessary to observe possible effects with a power of 0.85, therefore 20 representatives of the judiciary participated in the survey. The mean age was 42.4 years (SD = 12.10). The average work experience was 18.65 years (SD = 11.45). The research sample was equal in terms of gender, i.e. there were 10 women and 10 men among the participants. Four participants were assessors (20%), ten prosecutors (50%) and six police officers of criminal departments (30%). Descriptive statistics for quantitative variables were calculated. The results are shown in Tables. Table 1. shows questions relating to the detailed expectations of a technological solution. The answers were given on a seven-point scale, where 1 was described as "not important" and 7 "very important".

Table 1.

No.	Question	Median (Me)	Mean (M)	Standard Deviation (SD)
1.	Is the photo a thumbnail?	2	2.35	0.87
2.	Is the photo clear?	5	5.45	1.09
3.	Is there a human in the photo?	6	5.40	1.18
4.	Is there any nudity in the photo?	6	6.10	1.02
5.	Are there any children in the photo?	7	6.65	0.58
6.	In what	4.5	4.3	1.62

	period of life are the children?			
7.	What is the degree of child sexualization?	4.5	4.55	0.75
8.	Is there a girl in the photo?	3.5	3.3	1.21
9.	Is there a boy in the photo?	2	2.35	0.93
10.	Is there a baby in the photo?	2	2.10	0.96

Friedman's F variance analysis was performed for dependent samples ($F = 138.94$; $df = 9$; $p < 0.001$). The results of the pairwise comparisons are shown in Table. The significant values for many tests were corrected by the Bonferroni method.

Table 2. shows beliefs regarding the expected effects of applying a technological solution. The answers were given on a seven-point scale, where 1 was described as "not important" and 7 "very important. Table 3. shows beliefs regarding the expected effects of applying a technological solution.

Table 2.

No.	Question	Me	M	SD
1.	The introduction of an IT solution for the automatic detection of pornographic materials will allow for the acceleration of the work of law enforcement agencies.	7	6.60	0.50
2.	Relying on IT solutions may lead to fewer errors than in the case of self-evaluation by an expert.	6	5.85	0.98
3.	The use of the material assessment tool will be replaced by the need to consult an expert.	3	2.95	1.23
4.	The use of tools that will automatically classify pornographic material may lead to the overlooking of pornographic material, thus not identifying the perpetrator or the crime or not convicting him in a lawsuit.	3	3.05	0.99

Table 3.

Comparison	Statistic	p	Comparison	Statistic	p
10 - 9	.078	1	1 - 8	-1.514	1
10 - 1	.235	1	1 - 6	-3.264	.049

10 - 8	1.749	1	1 - 7	-3.473	.023
10 - 6	3.499	.021	1 - 2	-4.857	< 0.001
10 - 7	3.708	.009	1 - 3	-5.066	< 0.001
10 - 2	5.092	< 0.001	1 - 4	-6.293	< 0.001
10 - 3	5.301	< 0.001	1 - 5	-6.998	< 0.001
10 - 4	6.528	< 0.001	8 - 6	1.749	1
10 - 5	7.233	< 0.001	8 - 7	1.958	1
9 - 1	.157	1	8 - 2	3.342	.037
9 - 8	1.671	1	8 - 3	3.551	.017
9 - 6	3.421	.028	8 - 4	4.778	< 0.001
9 - 7	3.630	.013	8 - 5	5.483	< 0.001
9 - 2	5.013	< 0.001	6 - 7	-.209	.835
9 - 3	5.222	< 0.001	6 - 2	1.593	.111
9 - 4	6.450	< 0.001	6 - 3	1.802	.072
9 - 5	7.155	< 0.001	6 - 4	3.029	.002
2 - 3	-.209	.835	6 - 5	3.734	< 0.001
2 - 4	-1.436	.151	7 - 2	1.384	.166
2 - 5	-2.141	.032	7 - 3	1.593	.111
3 - 4	-1.227	.220	7 - 4	2.820	.005
3 - 5	-1.932	.053	7 - 5	3.525	< 0.001
4 - 5	-.705	.481			

Friedman's F variance analysis was performed for dependent samples ($F = 50.50$; $df = 3$; $p < 0.001$). In pairwise comparisons, statistically significant differences were found for all pairs ($p < 0.001$). Significance values for many tests were corrected by the Bonferroni method. Table 4. shows beliefs regarding the expected effects of applying a technological solution.

Table 4.

No.	Statement	Me	M	SD
1.	Indication, from the entire set of secured photos and videos of the suspect or accused person, 100 files that are most likely to be pornographic material with the participation of minors	7	6.4	0.94
2.	Indication, on the selected photo or video, of anatomical areas that would justify the "decision" made by the neural network (example – the neural network classified a given photo as representing a person at the age of 15, additionally indicating the area of the breast by marking those anatomical fragments that from the biological point of view are of the greatest importance in age differentiation)	4.5	4.6	1.75
3.	Indication of those materials that most	6	5.	1.09

	likely show a high level of aggression towards minors (e.g. use of violence) or with the participation of animals	9		
	Search for photos that are most visually close to each other (example – search for a photo with children in a specific "pose")	3.5	3	1.38

Friedman's F variance analysis was performed for dependent samples ($F = 33.71$; $df = 3$; $p < 0.001$). In pairwise comparisons, statistically significant differences were found for two pairs: comparing the fourth statement with the third ($p < 0.001$) and the fourth statement with the first ($p < 0.001$). The other pairwise comparisons were not statistically significant ($p > 0.05$). Significance values for many tests were corrected by the Bonferroni method. Table 5. shows expectations for a hypothetical pornographic material criminal case.

Table 5.

Description	Me	M	SD
Statement 1.	6	6.00	1.07
Statement 2.	6	6.10	0.78
Statement 3.	6	6.00	0.79

Friedman's F variance analysis was performed for dependent samples ($F = 0.375$; $df = 2$; $p = 0.829$). Pairwise comparisons were not performed because the results were not statistically significant. Descriptive statistics were calculated for the presented technological description on a scale from 1 to 7, where 1 was described as "not exhaustive" and 7 "is exhaustive" ($M = 6.70$. $Me = 7$, $SD = 0.47$).

4. DISCUSSION

The results were analysed quantitatively as well as qualitatively. These results lead to some basic conclusions.

Firstly, a list of functionalities of the future technological solution was created, which is characterized by high compliance of the competent judges. The obtained result justifies the use of the list created in this way in clinical practice. According to the judges' assessment, the most important information was the determination of the degree of sexualization of children, the differentiation between adolescence and whether nudity is visible in the photo. The finding that there is only one infant in the photo was considered the least useful feature. It is surprising that for the competent judges, the functionality of the network related to differentiating clear from blurred images (allowing for human age to be assessed) was not considered significant and was

finally assigned the third to last position. It might seem that this functionality could be of greater importance in the application of a solution in the administration of justice due to the possibility of the defense questioning that the material in question is of low quality, which would not constitute grounds for a possible conviction. As expected, the determination of the degree of sexualization of children was considered to be an important functionality of the technological solution. It seems to be a very important premise for the judiciary, as it indirectly determined the classification of an act. The degree of sexualization of children, according to the COPINE scale [Qua00a], indirectly answers several important questions. One of them is the differentiation between *pornography* and “child *eroticism*”. Another is the statement that there are animals on a given material, which changes the classification of the act and increases the size of the possible penalty. It should be noted that the assessments of all study participants regarding the expected functionalities of the neural networks are consistent with the assessments made by the competent judges. These results were expected because the assessments were made by the same study participants, but in a smaller number. The most important element for all participants of the study was to determine whether the material contains children, and the least important whether the material is a thumbnail consisting of other photos, and whether the material shows a boy.

Secondly, it seems that representatives of the judiciary are generally optimistic about the proposed technological solution, seeing in it more advantages than disadvantages. The respondents largely agreed that the introduction of an IT solution for the automatic detection of pornographic materials will allow for the acceleration of the work of law enforcement agencies. As expected, they did not agree that the use of tools that would automatically classify pornographic material could lead to the overlooking of pornographic material, thus not identifying the perpetrator of the crime or not showing it in a lawsuit. It is worth noting that the respondents also see it as an opportunity to reduce the number of errors made by court experts.

Thirdly, among the set functionalities concerning the technological solution, it was surprising that for the participants of the study it was less significant to indicate anatomical areas that would justify the “decision” made by the neural network in the selected photo or video. From the point of view of increasing the explainability and the possibility of interpreting the operation of neural networks, it could seem that this information will be

significant for practitioners. However, the results show that the most important thing is to select a given number of materials from a given set of photos that are most likely to be pornographic material with the participation of minors.

When answering the question about the independence of the technological solution being created, experts drew attention to several aspects. Experts emphasized that the stage of applying the tools is important. It seems that the created technological solution could be used without the participation of a forensic expert in the field of sexology, but only at the stage of investigation by the police. At further stages of the case, if the prosecutor’s office considers the collected material sufficient, the technological tool used could also be applied, but with the necessary participation of a court expert. It was pointed out that a specialist would be necessary because it would be difficult to treat the result of the tool as independent evidence. Moreover, the issue of responsibility for the submitted opinions was mentioned.

5. REFERENCES

- [EU00a] Directive EU COM/2010/0094. Accessed 03.10.2022 from <https://eur-lex.europa.eu/legal-content>.
- [Car00a] Carline, A., Palmer, E., Burton, M., Kyd, S., Jooman, P. Assessing the Implementation of the Sentencing Council’s Sexual Offences Definitive Guideline (2018).
- [Moz00a] Mozgawa, M. Kozłowska, P. Prawnokarne aspekty rozpowszechniania pornografii. *Prokuratura i Prawo*, 3 (2002).
- [Vit00a] Vitorino, P., Avila, S., Perez, M., Rocha, A. Leveraging deep neural networks to fight child pornography in the age of social media. *Journal of Visual Communication and Image Representation*, 50, pp. 303-31 (2018).
- [Wer00a] Wehrmann, J., Simões, G. S., Barros, R. C., Cavalcante, V. F. Adult content detection in videos with convolutional and recurrent neural networks. *Neurocomputing*, 272, pp. 432-438 (2018).
- [Qua00a] Quayle, E. The COPINE project. *Irish Probation Journal*, 5, pp. 65-83 (2008).
- [San00a] Sanchez, L., Grajeda, C., Baggili, I., Hall, C. A practitioner survey exploring the value of forensic tools, ai, filtering & safer presentation for investigating child sexual abuse material (CSAM). *Digital Investigation*, 29, pp. 124-142 (2019).

The use of artificial intelligence for automatic waste segregation in the garbage recycling process

Janusz Bobulski

Czestochowa University of Technology
Department of Computer Science
69 Dabrowskiego str.
42-201 Czestochowa, Poland
januszb@icis.pcz.pl
orcidID: 0000-0003-3345-604X

Mariusz Kubanek

Czestochowa University of Technology
Department of Computer Science
69 Dabrowskiego str.
42-201 Czestochowa, Poland
mkubanek@icis.pcz.pl: orcidID:
0000-0001-9651-9525

ABSTRACT

The problem of recycling secondary raw materials remains unresolved, despite many years of work on this issue. Among the many obstacles that arise is also the difficulty of sorting individual waste fractions. To facilitate this task and help solve this problem, modern computer vision and artificial intelligence techniques can be used. In our work, we propose constructing an intelligent garbage bin containing a camera and a microcomputer along with software that uses these techniques to sort waste. The role of the software is to recognize the type of waste and assign it to one of five main categories: paper, plastic, metal, glass and cardboard. The proposed method uses image recognition techniques with a convolutional neural network. The results confirm that using artificial intelligence methods significantly helps in sorting waste.

Keywords

recycling; environmental protection; artificial intelligence; computer vision; intelligent garbage bin.

1 INTRODUCTION

Out of all the materials that flow through the Polish economy, only one-tenth are recyclable, indicating the low circularity of the economy [Pace22]. This situation needs to change as soon as possible, both from economic and environmental perspectives. To address this, experts from the Polish Institute for Innovation and Responsible Development of Innowo, the Norwegian strategic agency Natural State, and the Dutch non-profit organization Circle Economy have undertaken the ambitious task of estimating the extent to which the Polish economy operates as a closed circuit. Their work is summarized in the "Circularity GAP Report Poland," which is funded under the EEA and Norway Grants programs. This report is the first of its kind on the Polish economy [Eurostat22, Stat21]. They discovered that the economy utilizes a total of 610 million tonnes of materials per year, with the consumption of virgin raw materials amounting to 520 million tonnes or 14 tonnes per

person per year. Consequently, only 10% of the materials are recycled.

To improve this situation, modern image processing techniques supported by artificial intelligence can be used. The article proposes using these methods to develop an intelligent waste bin that supports waste sorting and recycling.

Our main contribution to the work is the proposal of a waste recognition method that, when combined with a microcomputer and camera, can be an element of an intelligent waste bin.

2 RELATED WORKS

Almost all countries have recognized the issue of waste management and are taking the problem seriously [Han13]. When designing smart cities and promoting sustainable urban development, designers and scientists put a significant effort into building efficient Waste Management Systems (WMS). The demand for creating effective waste management systems is high, as it has a significant impact on protecting the environment and public health [Bag19]. Currently, the focus is on using cutting-edge technologies to improve and automate services, such as the Internet of Things (IoT), information technology, and Machine Learning (ML) [Bob13]. These technologies have drastically improved the efficiency of various WMS processes, enabling the forecasting of waste, collection, transport, sorting, and recycling [Now20, Ko22].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

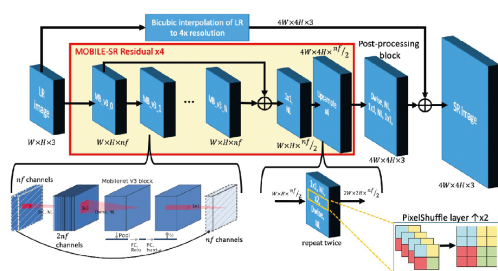


Figure 1: The structure of MobileNet-V3 [How19]

Several pre-existing CNN networks, such as ResNet-50 [Ma22], YOLO [Val19], MobileNet-V2 [Zio20], and a Hybrid of CNN and multilayer perceptron, have been used for waste classification tasks. Waste classification is an essential activity that separates different types of waste and dramatically improves the recycling process's efficiency. A project offers a waste bin equipped with a camera, microcontroller, and servo motor to separate various types of waste. The bin hardware is controlled by non-standard software based on the ResNet-34 algorithm [Ko22]. DL methods significantly impact the recycling process by detecting different types of materials and items for segregation, making the recycling process more efficient in recovering materials.

3 PROPOSED METHOD

To develop a method of automatic waste sorting, we proposed using a neural network that would quickly and efficiently meet our expectations. We used a particular type of convolutional neural network because, in our opinion, it is the best proposal for recognizing this type of image. We chose the MobileNetV3 Large neural network due to its simple implementation and high efficiency.

The neural network based on mobile models was built on more efficient structural elements. MobileNetV3 uses a combination of convolutional layers as building blocks to build the most effective models. Layers are also updated with modified 'swish' nonlinearities. They use a sigmoid that can be inefficient for computation and questioning accuracy retention in fixed-point arithmetic. Therefore, it was replaced with a hard sigmoid [How19]. The construction of the MobileNetV3 network is shown in Figure 1.

Recognition of many objects in images is possible thanks to the YOLO network. YOLO is a method of identifying and recognizing objects in real-time in photos. It stands for You Only Look Once (YOLO) [Red16]. YOLO delivers state-of-the-art results by using an entirely new approach to object recognition, easily surpassing previous real-time object detection methods. The YOLO method divides the image into N grids, each having an equal SxS dimensional sector.



Figure 2: Scheme of forecasting and frame reduction [Red16].

Each of these grids N is responsible for detecting and locating the object it contains.

YOLO skips all bounding boxes with lower probability scores in attenuation. This is done by examining the probability scores associated with each option and selecting the highest score. This process continues until the repeating bounding boxes are removed.

We decided to use the YOLOv4 Tiny network. The Tiny version is less computationally demanding and works perfectly for our tasks. To prepare the training data, it was necessary to mark the waste on each image. For labelling, we used a specialized tool to mark objects and assign them to the appropriate classes quickly. An example of marking objects is shown in Figure 2.

In order to avoid incorrect assignment of objects to classes during recognition, we used two cameras positioned at different angles to the observed object, as in the case of the MibileNetv3 network.

4 EXPERIMENT

The input data is a set of photos showing the waste assigned to five classes: glass, metal, paper, plastic and cardboard. The data was divided in proportion to learning/testing: 90/10, 80/20, 70/30 and 60/40 per cent of all photos within each class. In each experimental case, 10% of the training data was extracted as validation data. This allowed for more precise observation of the learning process and the analysis of the possibilities of the trained network to work in real conditions.

4.1 Datasets

The problem with the input data for convolutional network training is that quite often glass is similar to plastic, cardboard may look more like a background (piece of furniture) than the actual internal appearance of the box, paper may show in the photo e.g. metal or plastic, etc. It is important to collect as many photos as possible in individual classes so that, despite the definitely long learning process, the accuracy of the already learned system is at an acceptable level. Preparing input data for the learning and testing phase is an essential element of the research process. For deep neural

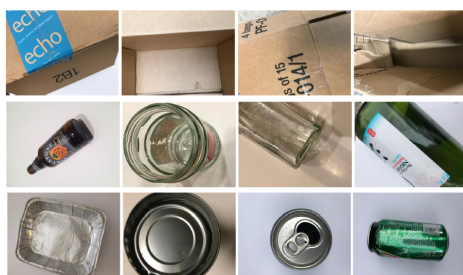


Figure 3: Sample images from dataset [Chan18]

networks, collecting the largest amount of data for each defined class is necessary. Preliminary research showed that the minimum number of images per class should be greater than 500. All the collected photos showed objects classified into the five considered classes: paper, cardboard, glass, plastic and metal. These images come from the database Trashnet (2527 images) [Chan18], and their samples can be seen in Fig. 3. To increase the number of images in individual classes and obtain a similar number of images for each category, geometric transformations of images rotation, reflection and shift, were used. Such a set of data (about 1,000 simulated images per class) was enough to teach the prepared network structures properly. Objects causing the greatest difficulty in recognizing, i.e. transparent, textured and cardboard, were also selected for the experiment. The pictures were taken by placing the object on a white background and using natural light or bulb lighting. The pictures have been resized down to 512 x 384. The data acquisition process involved using a white posterboard as a background and taking pictures of trash and recycling around homes. The lighting and pose for each photo is not the same, which introduces variation in the dataset. Data augmentation techniques were performed on each image because of the small size of each class. These techniques included random rotation of the image, random brightness control of the image, random translation of the image, random scaling of the image, and random shearing of the image. These image transformations were chosen to account for the different orientations of recycled material and to maximize the dataset size.

4.2 Results

The learning process of the neural networks was carried out on a computer: AMD Ryzen 9 5950X 16-Core Processor 3.40 GHz, RAM 128 GB, NVIDIA GeForce RTX 3090 GPU. division of input data into training and testing. Four divisions were used: - 90% (training data) - 10% (test data), 80% (training data) - 20% (test data), 70% (training data) - 30% (test data) and 60% (test data) training data) - 40% (test data). The results of the experiments are presented in Table 4.1. Analyzing the obtained results, it can be seen that the MobileNetV3

network is quite good at recognizing objects with much greater use of training data concerning test data. The less training data, the more complicated the recognition process becomes, which illustrates the decline in the correct verification of objects. The advantage of this network is undoubtedly the learning time. Using two cameras and activating the network on each of the cameras allows you to avoid situations where, for example, an object on a newspaper presents a plastic bottle, and in fact, it is a newspaper, i.e. paper.

In the case of the YOLOv4 Tiny network, a much higher accuracy of correct object recognition can be noticed even with a reduced training data set. This is due, firstly, to a different approach to teaching the neural network and, secondly, to a relatively low threshold of object recognition, which with the standard settings of the YOLO network, is 25%. Increasing this threshold skips recognized objects but prevents misclassification. Unfortunately, the learning time of the network has increased significantly. Still, after learning the operation process runs without real-time delays, we can say that the FPS resolution is about 25 frames. MobileNetV3 performs better in processing, but our misuse of GPU computing may generate this data. That's why we didn't include the FPS scores in the table. Our experiment results are similar to other methods using CNN - 96% in [Udd22] and [Gol19]. This confirms the validity of the idea of using artificial intelligence in the field of environmental protection and the usefulness of the presented method.

5 CONCLUSIONS

The obtained results show that our method copes quite well with a specific waste classification, especially when the training data are good examples. For bad patterns, it is necessary to increase the training data within each class and train the network with these bad patterns. The accuracy of the presented system at the level of over 96% for the YOLO4 Tiny network and over 94% for the MobileNetV3 network at the first assumed division into training and test data is acceptable for the proper functioning of the system and allows us to assume that the proposed method has a good chance of commercial use in real sorting plants waste. We are particularly pleased with the correctness of distinguishing waste by the YOLO network in a situation where there is more than one object within the camera's field of view.

6 ACKNOWLEDGMENTS

The project financed under the program of the Polish Minister of Science and Higher Education under the name Regional Initiative of Excellence in the years 2019-2023 project number 020/RID/2018/19 the amount of financing PLN 12 000 000.

No of division	Type of division	MobileNetV3 accuracy [%]	Training time [min]	YOLOv4 Tiny accuracy [%]	Training time [min]
1	90% - 10%	94.27	169	96.82	268
2	80% - 20%	89.14	164	96.25	224
3	70% - 30%	81.05	135	92.24	199
4	60% - 40%	74.65	122	90.09	183

Table 1: Results of experiment

7 REFERENCES

- [Bag19] M. Bagheri and R. Esfilar and M. S. Golchi and C. A. Kennedy, A comparative data mining approach for the prediction of energy recovery potential from various municipal solid waste, *Renew. Sustain. Energy Rev.*, 116, Dec, 2019. doi=10.1016/j.rser.2019.109423
- [Bob13] Bobulski, Janusz, Hidden Markov models for two-dimensional data, *Advances in Intelligent Systems and Computing*, 226, pp. 141-149, 2013. doi = 10.1007/978-3-319-00969-8-14
- [Bob22] Bobulski, Janusz and Kubanek, Mariusz, Robot for plastic garbage recognition, *International Journal of Electrical and Computer Engineering*, 12 (3), pp. 2425-2431, 2022. doi = 10.11591/ijece.v12i3.pp2425-2431
- [Chan18] C. Chang, Garbage Classification, Kaggle, 2018. url=https://www.kaggle.com/ds/81794, doi=10.34740/KAGGLE/DS/81794,
- [Eurostat22] Eurostat, Annual enterprise statistics by size class for special aggregates of activities (NACE), 2022, <https://ec.europa.eu/eurostat/databrowser/view/>, last access 2022-10-21.
- [Gol19] Golbaz, S., Nabizadeh, R.; Sajadi, H.S. Comparative study of predicting hospital solid waste generation using multiple linear regression and artificial intelligence, *J. Environ. Health Sci. Eng.*, 17, pp. 41-51, 2019. doi.org/10.1155/2022/2073482
- [Han13] M. A. Hannan and M. Arebey and R. A. Begum and A. Mustafa and H. Basri, An automated solid waste bin level detection system using Gabor wavelet filters and multilayer perception, *Resour. Conserv. Recycl.*, 72, pp. 33-42, 2013., doi=10.1016/j.resconrec.2012.12.002
- [How19] A. Howard and M. Sandler and G. Chu and L. Chen and B. Chen. and M. Tan and W. Wang and Y. Zhu and R. Pang and V. Vasudevan and Q. Le and A. Hartwig, Searching for MobileNetV3, *Computer Vision and Pattern Recognition*, 11, 2019.
- [Ko22] Koehler, A. and Rigi, A. and Breuss, M., Fast Shape Classification Using Kolmogorov-Smirnov Statistics, *Computer Science Research Notes*, 3201, pp. 172-180, 2022 doi: 10.24132/CSRN.3201.22.
- [Ma22] Ma, X. and Li, Z. and Zhang, L., An Improved ResNet-50 for Garbage Image Classification, *Tehnicky vjesnik*, 29(5), pp. 1552-1559, 2022. doi=doi.org/10.17559/TV-20220420124810
- [Now20] P. Nowakowski and T. Pamula, Application of deep learning object classifier to improve e-waste collection planning, *Waste Manag.*, 109, pp. 1-9, 2020. doi=10.1016/j.wasman.2020.04.041
- [Pace22] Platform for Accelerating the Circular Economy (PACE), The Circularity Gap Report - Poland, Circularity Gap Report, 2022, <https://circularweek.org/online/circularity-gap-report-poland/>, last access 2022-10-21.
- [Red16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, 2016. doi = doi: 10.1109/CVPR.2016.91,
- [Stat21] Statistics-Poland, Yearbook of labour statistics 2020, 2021, <https://stat.gov.pl/en/topics/statistical-yearbooks/statistical-yearbooks/yearbook-of-labour-statistics-2020,10,8.html?>, last access 2022-10-21.
- [Udd22] Uddin, Ziya and Kshirsagar, Pravin R. et al., Artificial Intelligence-Based Robotic Technique for Reusable Waste Materials, *Computational Intelligence and Neuroscience*, Hindawi, 2022.
- [Val19] M. Valente and H. Silva and J. M. L. P. Caldeira and V. N. G. J. Soares and P. D. Gaspar, Detection of waste containers using computer vision, *Appl. Syst. Innov.*, 2, pp. 1-13, 2019. doi=10.3390/asi2010011.
- [Zio20] D. Ziouzos and D. Tsiktsiris and N. Baras and M. Dasygenis, A Distributed Architecture for Smart Recycling Using Machine Learning, *Futur. Internet*, 12(9), 141, 2020. doi=10.3390/fi12090141

Detection of Dangerous Situations Near Pedestrian Crossings using In-Car Camera

Mariusz Kubanek	Lukasz Karbowski	Janusz Bobulski
Czestochowa University of Technology	Czestochowa University of Technology	Czestochowa University of Technology
Dabrowskiego Street 69 42-201, Czestochowa, Poland	Dabrowskiego Street 69 42-201, Czestochowa, Poland	Dabrowskiego Street 69 42-201, Czestochowa, Poland
mariusz.kubanek@icis.pcz.pl	lukasz.karbowski@icis.pcz.pl	janusz.bobulski@icis.pcz.pl

ABSTRACT

The paper presents a method for detecting dangerous situations near pedestrian crossings using an in-car camera system. The approach utilizes deep learning-based object detection to identify pedestrians and vehicles, analyzing their behavior to identify potential hazards. The system incorporates vehicle sensor data for enhanced accuracy. Evaluation results show high accuracy in detecting dangerous situations. The proposed system can potentially enhance pedestrian and driver safety in urban transportation.

Keywords

Object detection, deep learning, autonomous systems.

1 INTRODUCTION

Pedestrian safety is a critical concern in urban transportation, with accidents often occurring at pedestrian crossings. Existing advanced driver assistance systems (ADAS) may not effectively detect dangerous situations near crossings. This paper proposes a method that utilizes a deep learning-based object detection algorithm using in-car cameras to identify pedestrians and vehicles near pedestrian crossings. The algorithm analyzes behavior and incorporates vehicle sensor data for enhanced accuracy. Related work, proposed method, dataset, experimental results, and conclusions are discussed in subsequent sections.

2 RELATED WORK ON PEDESTRIAN DETECTION AND RECOGNITION

"The concept of autonomous vehicles was proposed decades ago. With the development of hardware and algorithms, autonomous vehicles have become a reality [Bou00, Bob00]. The autonomous vehicle is an essential participant in intelligent transport systems, and it is capable of self-driving without human drivers' navigation [Sun00]. Human drivers' misoperation causes

many traffic accidents due to physical and mental conditions. Besides, some complex traffic scenarios and bad weather conditions can also lead to traffic tragedies. The application of autonomous vehicles could decrease the number of traffic accidents by making judgments more reasonably and driving more reliably [Alj00]. Before using autonomous vehicles, the current stage's fundamental objective is to ensure all traffic participants' safety [Zha00]. Pedestrians are very vulnerable in a traffic collision compared to the passengers inside the vehicle. Therefore, ensuring the safety of pedestrians is a critical step in advancing the application of autonomous vehicles [Wan00, Kar00].

In [Guo00], the authors propose a multi-scale feature fusion convolutional neural network (MFF-CNN) for pedestrian detection. The proposed approach is evaluated on three challenging pedestrian datasets: Caltech, INRIA, and ETH.

The authors of the work [Kon00] proposed a new approach for pedestrian detection based on Faster R-CNN, which overcomes the limitations of detecting small objects or objects with similar backgrounds. The proposed method combines contextual information with multi-level features to improve detection accuracy. Contextual information is used to help detect pedestrians from cluttered backgrounds, while multi-level features are more informative for detecting small-size pedestrians.

Two methods of object recognition were used in [Khe00]. This work focuses on new approaches in the embedded vision for object detection and tracking

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

in drone visual control. Two methods are used: 1) Classical image processing with improved Histogram Oriented Gradient (HOG) and Deformable Part Model (DPM) for real-time object (pedestrian) detection and distance estimation. 2) Deep learning-based object (pedestrian) detection for target position estimation and visual serving using improved HOG and PID controllers.

In [Pag00], the authors propose a new method for pedestrian detection using a combination of hand-crafted features and a modified pre-trained ResNet-18 network called Multi-layer Feature Fused-ResNet (MF2-ResNet).

3 PROPOSED METHOD

This section describes the proposed method for detecting potentially dangerous situations near pedestrian crossings using a camera-equipped vehicle. We utilize the popular You Only Look Once version 4 (YOLOv4, [Boc00]) object detection algorithm to identify pedestrians, vehicles, and other objects in the camera images.

Dataset

Our proposed method was evaluated on a diverse dataset of camera images captured by a vehicle near pedestrian crossings. The dataset includes images captured under various lighting, weather, and traffic conditions, and was manually annotated to include pedestrian crossings, objects such as pedestrians, vehicles, signs, and traffic lights, and different types of scenarios with and without traffic lights or signs, and various types of vehicles. Automatic object detection was used to generate training data with high accuracy, validated against the annotated ground truth.

YOLOv4 Object Detection

Our proposed method utilized the state-of-the-art YOLOv4 network for accurate and fast detection of pedestrians, vehicles, signs, and other objects near pedestrian crossings. YOLOv4 uses a single neural network with advanced techniques for improved accuracy, including spatial pyramid pooling and feature fusion. We fine-tuned the pre-trained YOLOv4 model with our annotated dataset to achieve high accuracy and speed in detecting objects in diverse lighting and weather conditions.

Detecting Dangerous Situations

Detecting Dangerous Situations: We combine object detection results with other information to detect potentially dangerous situations near pedestrian crossings. We analyze relevant objects like pedestrians, vehicles, signs, and traffic lights, and identify situations such as pedestrians crossing outside designated areas or vehicles approaching at high speed. We use rule-based and

machine learning-based approaches to improve accuracy, capturing typical patterns of dangerous situations and training a model for complex scenarios. Examples include detecting vehicles running red lights or pedestrians crossing outside designated areas, showcasing the effectiveness of our method.

Our method detected many potentially dangerous situations near pedestrian crossings, demonstrating its effectiveness in improving pedestrian safety (Figure 1).

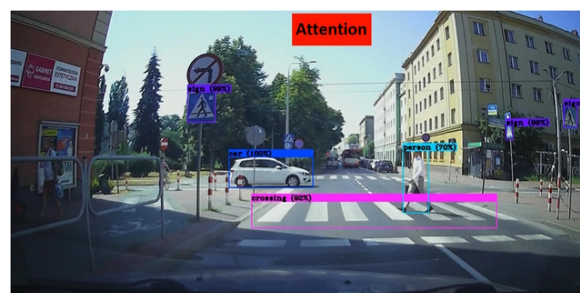


Figure 1: Sample frames from the dataset with dangerous situation detection

Combination of Rule-based and Machine Learning-based Approaches

We use rules based on domain knowledge and experience in pedestrian safety to capture typical and less common patterns of dangerous situations near pedestrian crossings. We complement this with machine learning techniques, training a deep neural network using YOLOv4 to detect relevant objects and classify them based on appearance and location. The neural network's output enhances the rule-based system, allowing it to detect complex and unusual situations. Techniques such as data augmentation, transfer learning, and ensembling are used to improve the performance of our machine learning-based approach. The combination of rule-based and machine learning-based approaches enables us to detect a wide range of dangerous situations near pedestrian crossings.

4 EXPERIMENTAL RESULTS AND ANALYSIS

We evaluate the performance of our method on a dataset of real-world traffic scenarios.

Object Detection Results

Table 1 shows the object detection results using our YOLOv4-based approach only for selected objects. We report the average precision (AP) and average recall (AR) for each class of objects. As can be seen from the table, our approach achieved high detection performance for all classes of objects, with an overall mAP of 0.92.

Object Class	Average Precision (AP)	Average Recall (AR)
Pedestrian	0.94	0.93
Vertical Sign	0.88	0.87
Horizontal Sign	0.91	0.90
Traffic Light	0.96	0.95
Cyclist	0.90	0.89
Overall mAP	0.92	0.91

Table 1: Object detection results using our YOLOv4-based approach

From the object detection results presented in Table 1, we can see that our YOLOv4-based approach achieved high detection performance for all classes of objects. The highest detection performance was achieved for traffic lights, with an AP of 0.96 and an AR of 0.95. The lowest detection performance was achieved for vertical signs, with an AP of 0.88 and an AR of 0.87.

Dangerous Situation Detection Results

Table 2 shows the results of our proposed method for detecting dangerous situations near pedestrian crossings: ST1 - pedestrian crossing road, ST2 - pedestrian on the road, ST3 - cyclist on the road, and ST4 - vehicle approaching the crossing. We report our method's precision, recall, F1-score, and accuracy for each type of dangerous situation. As can be seen from the table, our approach achieved high performance in detecting dangerous situations, with an overall accuracy of 0.88.

Situation Type	Precision	Recall	F1-score	Accuracy
ST1	0.84	0.91	0.87	0.91
ST2	0.89	0.88	0.87	0.86
ST3	0.91	0.89	0.90	0.89
ST4	0.82	0.84	0.83	0.84

Table 2: Dangerous situation detection results using our proposed method

Regarding detecting dangerous situations, as shown in Table 2, our proposed method achieved high performance for all dangerous situations. The highest accuracy was achieved for detecting pedestrians crossing the road, with an accuracy of 0.91. The lowest accuracy was achieved for detecting vehicles approaching the crossing, with an accuracy of 0.84.

Different detection scenarios

We also analyzed the detection of objects in different weather conditions and at different types of intersections. We assumed a sunny, cloudy, and rainy day for the weather conditions. Additionally, we estimated the strengthening of the objects for the YOLOv4 model by asking about the causes for various intersection scenarios: T-intersection, Four-way intersection, Roundabout,

and Pedestrian crossing. Table 3 shows the analysis results of object detection analysis for different weather. Table 4 shows the performance of the YOLOv4 object detection model on the pedestrian detection task for different intersection scenarios.

Condition	Object Detection Accuracy	Dangerous Situation Situation Accuracy
Sunny	0.92	0.90
Cloudy	0.89	0.88
Rainy	0.86	0.85

Table 3: The results of the analysis object detection for different weather

Intersection scenario	AP [%]
T-intersection	98.2
Four-way intersection	96.7
Roundabout	94.5
Pedestrian crossing	90.8

Table 4: The performance of the YOLOv4 object detection model on the pedestrian detection task for different intersection scenarios

Analysis of Results

Table 1 presents the results of the object detection task using the YOLOv4 model on the pedestrian detection dataset. The average precision (AP) and recall were calculated for each object class, including pedestrians, vertical signs, horizontal signs, traffic lights, and bicycles. The overall mAP was 92.5%, indicating the high accuracy of the YOLOv4 model in detecting objects relevant to pedestrian safety.

Table 2 presents the dangerous situation detection task results using the proposed method on the pedestrian detection dataset. The precision, recall, and F1 score were calculated for each detected dangerous situation, including jaywalking, cars blocking the crosswalk, and pedestrians crossing outside.

The proposed method achieved high performance in detecting dangerous situations, with an overall F1 score of 0.89. The highest F1 score was achieved for detecting jaywalking, which is the most frequent dangerous situation in the dataset.

The performance of the YOLOv4 object detection model on the pedestrian detection task was evaluated for different lighting and weather conditions. Table 3 presents the AP and recall for each object class under different conditions.

The YOLOv4 model showed consistent and high performance across lighting and weather conditions, with an overall mAP of 92.1%. However, the AP for pedestrians was slightly lower in low-light conditions, indicating the need for further improvement in detecting pedestrians in challenging lighting conditions.

The performance of the proposed method in detecting dangerous situations was evaluated for different intersection scenarios, including T-intersections, crosswalks with pedestrian signals, and crosswalks without pedestrian signals. Table 4 presents the precision, recall, and F1 score for each detected dangerous situation in each scenario.

The proposed method showed high performance in detecting dangerous situations across all scenarios, with an overall F1 score of 0.89. The highest F1 score was achieved for detecting jaywalking, the most frequent dangerous situation in all scenarios.

5 CONCLUSION AND FUTURE WORK

This paper proposes a novel approach for detecting dangerous situations at road intersections using rule-based and machine-learning techniques. Our method accurately detected dangerous situations, such as pedestrian crossing violations and red-light running. We also demonstrated the effectiveness of using the YOLOv4 object detection model for pedestrian detection tasks in different intersection scenarios.

Our results indicate that our approach has the potential to improve intersection safety by providing real-time warnings to drivers and pedestrians. However, our study has several limitations that can be addressed in future work. One limitation is that our dataset is limited to a specific geographic region and does not represent a wide range of traffic scenarios. Future work could collect data from different locations and use transfer learning techniques to address this limitation to improve the model's performance.

Another limitation is that our approach currently relies on fixed rules for identifying dangerous situations, which may not be optimal for all scenarios. Future work could explore using reinforcement learning techniques to learn optimal rules for different intersection scenarios.

6 ACKNOWLEDGMENTS

The project financed under the program of the Polish Minister of Science and Higher Education under the name "Regional Initiative of Excellence" in the years 2019 - 2023 project number 020/RID/2018/19 the amount of financing PLN 12,000,000.

7 REFERENCES

- [Alj00] Aljeri. N., Boukerche. A. A probabilistic neural network-based roadside unit prediction scheme for autonomous driving. In Proceedings of the IEEE International Conference on Communications, pages 1-6, 2019.
- [Bob00] Bobulski, J., Kubanek, M., Kulawik, J., Szymoniak, S. Data Container for Autonomous Cars, Applied Human Factors and Ergonomics International, Human Systems Engineering and Design (IHSED2021): Future Trends and Applications, AHFE International, Volume 21, Issue 21, pp 2, 2021.
- [Boc00] Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. Yolov4: Optimal speed and accuracy of object detection. Computer Vision and Pattern Recognition, Image and Video Processing, arXiv preprint arXiv:2004.10934, pp. 17, 2020.
- [Bou00] Boukerche, A., Robson, E. Vehicular cloud computing: Architectures, applications, and mobility. Computer Networks, 135:171-189, 2018.
- [Guo00] Guo, A., Yin, B., Zhang, J., and Yao, J. Pedestrian detection via multi-scale feature fusion convolutional neural network. 2017 Chinese Automation Congress (CAC), pp. 1364–1368, 2017.
- [Kar00] Karbowski, L., Kubanek, M. Using Edge Processing with Artificial Intelligence in Monitoring the Pedestrian Crossing, 14-th International Conference on Parallel Processing and Applied Mathematics (PPAM 2022), 2022.
- [Khe00] Khemmar, R., Gouveai, M., B. Decoux, B., Ertaud, J.Y. Real-Time Pedestrian and Object Detection, and Tracking-based Deep Learning. Application to Drone Visual Tracking, WSCG 2019 - CSRN 2902 - Short and Poster papers, 2019.
- [Kon00] Kong, W., Li, N., Li, T.H., and Li, G. Deep Pedestrian Detection Using Contextual Information and Multi-level Features. In book: MultiMedia Modeling, MultiMedia Modeling, MMM 2018, LNCS, Vol 10704, 2018.
- [Pag00] Panigrahi, S., and Raju, U. S. N. Pedestrian Detection Based on Hand-crafted Features and Multi-layer Feature Fused-ResNet Model. International Journal on Artificial Intelligence Tools, Vol. 30, No. 05, 2150028, 2021.
- [Sun00] Sun, P., AlJerri, N., Boukerche, A. Dacon: A novel traffic prediction and data-highway-assisted content delivery protocol for intelligent vehicular networks. IEEE Transactions on Sustainable Computing, 5(4):501-513, 2020.
- [Wan00] Wang, H., Wang, B., Liu, B., Meng, X., Yang, G. Pedestrian recognition and tracking using 3d lidar for an autonomous vehicle. Robotics and Autonomous Systems, 88:71-78, 2017.
- [Zha00] Zhang, J., Lin, L., Zhu, J., Li, Y., Chen, Y., Hu, Y., Hoi, S.C.H. Attribute-Aware Pedestrian Detection in a Crowd. IEEE Trans. Multim. 23: 3085-3097, 2021.

Photogrammetry workflow for obtaining low-polygon 3D models using free software.

Ricardo Pardo Romero
Institute of New Imaging Technologies
Universitat Jaume I
Spain, Castello de la Plana
ripardo@uji.es

Inmaculada Remolar Quintana
Institute of New Imaging Technologies
Universitat Jaume I
Spain, Castello de la Plana
remolar@uji.es

ABSTRACT

This paper proposes a workflow for inexperienced designers to create low-poly 3D models using free software. It addresses the problem of the complexity generated by photogrammetry. The solution aims to enable independent developers to create realistic assets at cheap cost. It eliminates the need for experienced 3D artists or expensive commercial solutions.

Keywords

Photogrammetry, Low-poly objects, free-software, geometric simplification, realism

1 INTRODUCTION

Highly realistic 3D objects often have a high polygon count. Simplification methods require manual work or expensive tools, creating challenges for independent video game studios. Interactive technologies like extended reality are used in Industry 4.0 and education, but end-users lack expertise in adapting virtual environments. Photogrammetry [2] is an Image Based modelling (IBM) technique that uses photographs as the fundamental medium to extract accurate measurements and information about the physical properties of objects and their surroundings. It offers a cost-effective solution, allowing non-experts to scan real objects and obtain digital representations quickly. LiDAR, another remote sensing technology, produces precise 3D maps but is affected by weather conditions. According to [4], while both of these methods capture locations, photogrammetry requires less expertise to provides photo-realistic results. The article proposes a solution for adapting photogrammetric objects using free software, benefiting independent studios and inexperienced creators.

Summarizing, the main contributions made in this paper are the following:

- A workflow is given to adapt point clouds obtained by photogrammetry to assets suitable for video games, with high realism but low geometric complexity.
- A method is given to optimise the scanned geometric mesh, independently of the applied photogrammetry method, using only free software.
- A simple pipeline is offered to perform the whole process of complexity reduction without losing realism, so that anyone, even without experience as a 3D artist, can follow it.
- To evaluate the level of performance, visual quality and automation, the results obtained with the proposed pipeline will be compared with other methods.

The remainder of the paper is organised as follows. Section 2 briefly reviews important methods explored in recent years. Section 3 outlines the design of the presented pipeline. In Section 4, the results obtained in the work are compared and discussed. Finally, Section 5 presents the conclusions and future work.

2 RECENT SOLUTIONS

Photogrammetry is a popular and user-friendly Image-Based Modeling (IBM) method for creating 3D models [6]. It involves capturing images from different perspectives and using software to recreate the object. However, real-time visualization is challenging due to the high polygon count. Structure from motion (SFM) [5] is one of the most popular photogrammetry methods. Laser scanning, particularly LiDAR, provides accurate depth information and is suitable for larger ob-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

jects and spaces. LiDAR is commonly used for mapping terrain and architectural purposes. While laser scanning is expensive, it is starting to be incorporated into mobile devices. Photogrammetry has gained traction in the gaming industry, reducing production costs and enhancing asset creation. Companies like, Smaller studios may face challenges due to limited resources, but free software and phone cameras can still yield impressive results. Despite the availability of free software, there is a lack of specific papers presenting a pipeline workflow. This article aims to fill this gap by providing steps to obtain accurate 3D representations of real-life objects using free software.

3 PROPOSED SOLUTION

The proposed solution aims to provide an easy workflow for obtaining a low-poly 3D model of a real object, specifically targeting inexperienced designers and indie game studios. The workflow utilizes free software to promote accessibility and affordability. The object chosen for this work is a complex piece of dry trunk, as shown in Figure 1, which exhibits non-homogeneous shapes and indentations.



Figure 1: Photograph of the real object to be digitally represented.

The workflow is summarized in Figure 2. It is primarily based on photogrammetry, starting with capturing pictures of the object using any device, in our case, the iPad Pro 2022. These images are then imported into *Meshroom* to generate a polygonal mesh. However, the resulting mesh is highly complex and needs simplification to be suitable for various applications. *Blender* and *Instant meshes* are used for this simplification process. To maintain realism in the simplified mesh, texture maps are created using *Blender*. Additional maps are generated using *Materialize* to enhance the realism and create a Physically Based Rendering (PBR) material.

3.1 Capturing the object

To ensure a faithful 3D model from photogrammetry, consider the following factors: Objects should be opaque, avoiding transparent or translucent materials [3]. Optimal lighting conditions are crucial for capturing accurate textures and proportions. Cloudy weather

is recommended for outdoor photography, while flat lighting is preferred indoors. Capture a minimum of 20 photos from various angles around the object, with at least 50% overlap and neighboring images [7]. In our experiment (Figure 3), 42 surrounding photos were taken. More photos improve the accuracy of the result.

3.2 Generating the point cloud and mesh

Once the images have been taken, it is time to generate the geometric mesh from them. For this, *MeshRoom* is the free option that has been selected for its simplicity. This 3D reconstruction software [1] is easy to use and allows the entire photogrammetric pipeline to be executed. Designers simply input the images obtained (seen in Figure 3) and the software generates a 3D model and textured mesh, using a node-based workflow.

At the end of this process, in the project folder *MeshRoom*, an .fbx 3D model will appear, with the geometric mesh information and the texture distribution unwrapped (Figure 5). The 3D model obtained usually has a large number of polygons. In this example there are 1,392,431 polygons, (Figure 4). This amount has to be reduced with the proposed pipeline to obtain the low-poly representation.

3.3 Optimizing the mesh

The next step involves decimating the geometry to obtain different levels of detail. Before this process, it is recommended to clean up unnecessary parts of the mesh captured during the photogrammetry process. *Blender* can be used to delete these parts. The modifiers in *Blender* are not suitable for this process due to their limitations. Instead, a third-party software called *Instant Meshes* is recommended for geometric simplification. It allows setting the desired number of polygons and predefined edge flow. In this case, the target polygon count is 6,892 seen in Figure 6.

Once the simplified mesh is obtained, a high-poly version of it is generated in *Blender* for texture baking, using the "Multi-Resolution" and the "ShrinkWrap" modifiers to project the vertices of the decimated mesh onto the SFM model's surface. Now we have two levels of detail.

Finally, *Blender*'s "Face orientation" option can be used to identify inverted normals and fix them, either flipping or recalculating them.

Mapping coordinates

The SFM mesh already have textures, but we want to put them on our new mesh. So new mapping coordinates need to be obtained for the simplified mesh. *Blender* offers the "UnWrap" modifier, and selecting

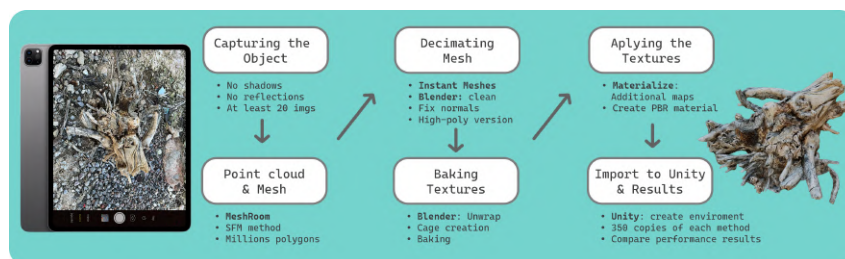


Figure 2: Overview of the presented work.

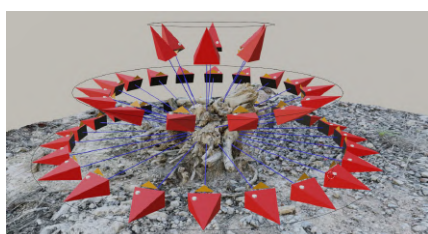


Figure 3: The cameras surrounding the object, represented by red pyramids.

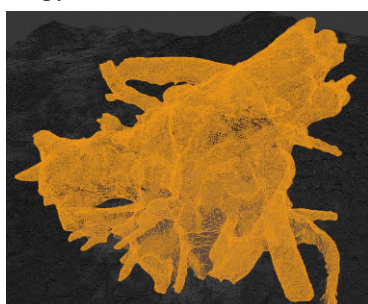


Figure 4: Geometric mesh extracted using the SFM method (*MeshRoom*). It is composed by 1,392,250 polygons.

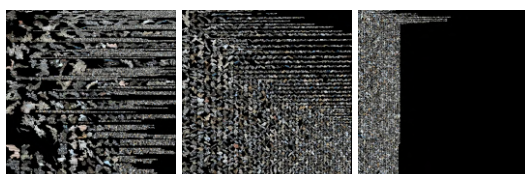


Figure 5: Unwrap distribution extracted using the SFM method (*MeshRoom*): 3 images are generated that contain all the unwrapped texture.

the "Smart UV projection" option can be a suitable solution on most cases, but depending on the mesh's complexity it is recommended to manually place marker seams to ensure accurate mapping. This process generates a 2D image where the texture of the simplified mesh is unwrapped.

Baking

To create texture maps for a 3D model, texture baking is a crucial process. It allows the recovery of lost details from the decimation process with minimal effort. Each baked map should be manually saved to avoid overwriting previous versions. To start, assign a default material

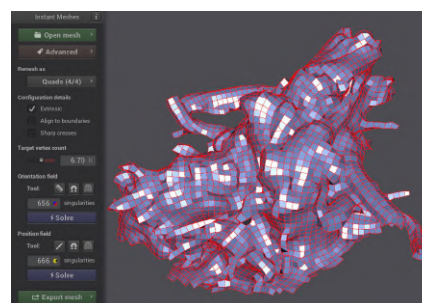


Figure 6: Configuration to obtain a decimated mesh in Instant Meshes.

with a 2D image map to the polygonal mesh. The content of this map is unimportant as it will be replaced. Configure the Render Properties to use the "Cycles" engine and access the "Bake" tab to begin the texture baking process. To extract the maps from a SFM 3D model in Blender, enable the "Selected to active" option and utilize the cage feature to determine texture projection points on the simplified mesh. Different texture maps are generated during the baking process. The *Diffuse* map contains color information, the *Normal* map analyzes high-poly mesh details, the *Ambient Occlusion* map adds depth through darkening areas with limited light exposure, and the *Height* map stores height information for vertex displacement. For enhanced realism, additional maps like *Roughness*, *Metallic*, and *Edge* can be generated using software like Materialize. To create a cage for baking and extracting textures, follow these steps. Duplicate the low-poly mesh and inflate it using the "Sculpt" section's "Inflate" tool. Ensure that the inflated cage completely covers the original mesh. Add the inflated cage in the "Bake" section and adjust the "Max Ray Distance" property for desired results. With the settings prepared, proceed with the baking process. For the *Diffuse* map, select only the "Color Contribution" and perform the bake. The *Height* map can only be generated using the "Multires" option and may not be applicable to all objects. Bake the remaining maps without changing any options. After baking in Blender, apply the generated maps to the textured object. The *Roughness* map obtained from Materialize will improve the visuals. However, the *Metallic* and *Edge* maps are unnecessary for this particular object.

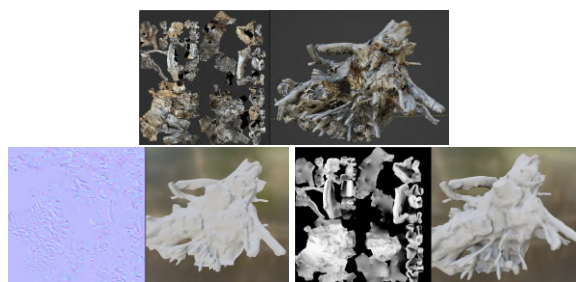


Figure 7: The *Diffuse*, *Normal* and *Ambient Occlusion* maps.

Once the maps are generated, link them to the applied material and evaluate the results. Additional adjustments can be made using Materialize to refine the *Roughness* map. For a better visual representation, consider adding the ground to the scene. This will enhance the overall presentation of the model.



Figure 8: Comparison between the original picture (left) and the render made in Blender 3D (right).

4 EXPERIMENTAL RESULTS AND DISCUSSION

Figure 9 shows a comparison of the visual results, highlighting the realism of each method. LiDAR does not provide good visual results for complex shapes. SFM representation has a very good visual quality. Our Low-Poly method has 99.5% fewer polygons than the SFM model, yet the visual quality remains comparable.

The LiDAR pipeline utilized an iPad app called Scanner 3D App, which produced a textured model. Considering the results, users may prefer one method over another based on their specific needs. LiDAR performs better with simpler objects and larger environments. SFM method has a lot of polygons but can be utilized to improve performance and maintain visual quality when combined with the presented pipeline.

5 CONCLUSION

With the pipeline proposed, obtaining photorealistic low-poly models is accessible to almost any developer. The contributions enable cost-effective creation of assets for interactive apps/games, with future plans to develop *Blender* plug-ins to automate part of the process.

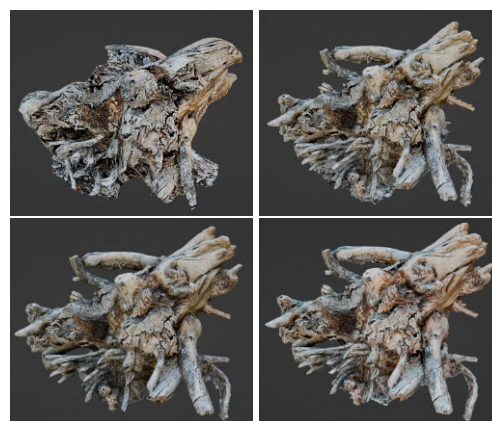


Figure 9: LiDAR (First), Imaged Based SFM (Second), High-Poly (Third), Low-Poly (Fourth).

Acknowledgements

Research supported by the e-DIPLOMA, project number 101061424, funded by the European Union. Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.



6 REFERENCES

- [1] Photogrammetric computer vision framework. <https://alicevision.org/>. Last accessed January 2023.
- [2] J. S. Aber, I. Marzloff, and J. B. Ries. *Small-Format Aerial Photography*. Elsevier Science, 2010.
- [3] Sébastien Lachambre, Sébastien Lagarde, Cyril Jover, et al. Photogrammetry workflow. *Rapport Technique, Unity*, 2017.
- [4] Thinal Raj, Fazida Hanim Hashim, Aqilah Baseri Huddin, Mohd Faisal Ibrahim, and Aini Hussain. A survey on lidar scanning mechanisms. *Electronics*, 9(5), 2020.
- [5] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [6] Nataska Statham. Use of photogrammetry in video games: A historical overview. *Games and Culture*, 15(3):289–307, 2020.
- [7] Krzysztof Woloszyk, Pawel Michal Bielski, Yordan Garbatov, and Tomasz Mikulski. Photogrammetry image-based approach for imperfect structure modelling and fe analysis. *Ocean Engineering*, 223:108665, 2021.

Polychromatism of all light waves: new approach to the analysis of the physical and perceptive color aspects

Justyna Niewiadomska-Kaplar

tab edizioni

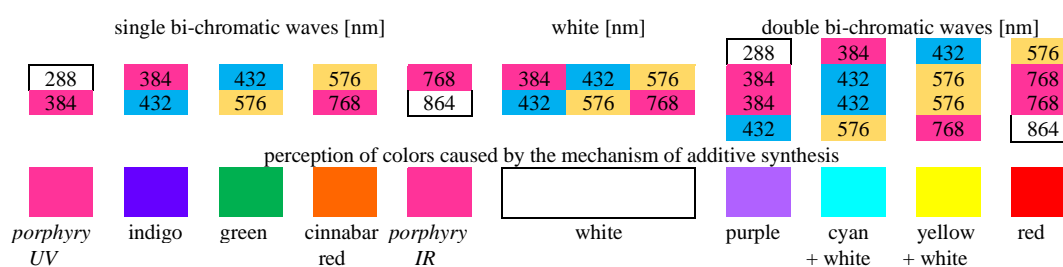
Viale Manzoni 24c

00185, Rome, Italy

micromacro@ymail.com

ABSTRACT

Research on light vision mechanisms in biosystems and on the mechanisms of formation of deficits in color discrimination [Nie20a] reveals that not only white light is polychromatic but all light waves are. The spectrum of white light is composed of aggregations of only 4 monochromatic waves: *magenta UV*¹ 384 nm, cyan 432 nm, yellow 576 nm and *magenta IR* 768 nm, grouped in 5 **bi-chromatic** waves: cinnabar red (*magenta IR* + yellow), green (yellow + cyan), indigo (cyan + *magenta UV*) and also two *semi-bright* bi-chromatic waves - *porphyry IR* (semi-infrared wave composed of the *magenta IR* 768 nm wave and the colorless infrared wave 864 nm) and *porphyry UV* (semi-ultraviolet wave composed of the *magenta UV* 384 nm wave and the colorless ultraviolet wave 288 nm). The light waves thus composed create the light sensations due to the mechanism of additive synthesis.



The method allows a new approach to interpret the composition of the bright waves, the phenomenon of decomposition of colours and additive synthesis that constitutes the principle of colour production in computers. The new elaborate models of colour physics also constitute the basis by interpretation of the mechanisms of vision of colours.

Keywords

Human photoreceptors (cones) are sensitive to: cyan 432 nm, yellow 576 nm and cyan + yellow = green 504nm, Bright waves consist of aggregation in couples of heterogeneous 4 monochromatic waves: 384 nm, 432nm, 576 nm and 768nm.

1. INTRODUCTION

This research is based on two beliefs regarding the collecting and processing of data on acoustic waves (distinction of the height of sound) and electromagnetic waves of the visible (distinction of colors):

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

1/ The human brain is a "digital" and non -analogical tool, therefore it produces information on the acoustics and electromagnetic waves through the codification of the information and not through the measurement of all acoustic waves of the audible and all electromagnetic waves of the visible;

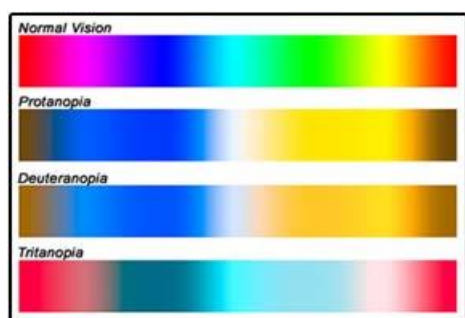
2/ The instruments of measurement of the wave motion such as cochlea or photoreceptors must be considered as biosensors who work as precise and objective instruments of measurement of these parameters of the wave motion for which they have

¹ The denomination of the waves that have wavelengths 384 and 768 nm as *magenta UV* or *IR* and of the couple of the waves that have wavelengths 384/288 nm and 768/864 nm as *porphyry UV* or *IR* is proposed from the author.

evolved in the space of millions of years, which therefore in their context of sensitivity they are equal at the measurement tools created artificially.

The starting point of the reasoning on the polychromatism of all light radiation has become analysis of the polychromatic composition of the light waves perceived such as red, green and blue revealed by the deficit in the distinction of colors. For this reason, the following table is proposed that illustrates some of these deficits and the citation of how they are explained in literature.

“Protanopia and deuteranopia are various forms of red-green colorblindness, the most common form of dichromia (protanopia is a deficiency in red cells, deuteranopia a deficiency in green; the results are similar because red and green are close in wavelength). Tritanopia is commonly called blue-yellow colorblindness and is less common, with full colorblindness (monochromia) being very rare.” [AVI22a]



Author proposes the following questions to readers:

- Why cyan is seen as white by the Protanopes and Deuteranopes?
- Why both green and yellow are seen as yellow by the protanopes and deuteranopes?
- Why is yellow seen as white by the Tritanopes?
- Why both cyan and green are seen cyan by the Tritanopes?

The answer to these questions proposed by the author reveals the polychromatism of the bright waves that we perceive as red, green and blue.

2. STATE OF ART

The proposed research verifies the following three postulates of the theories on the vision of color and on the nature of the bright radiation in which it is stated that:

- 1/ White light only is polychromatic: the visible light waves are countless monochromatic waves between 380-760 nm; [Enc08a]
- 2/ Human photoreceptors responsible for the distinction of colors (cones) are sensitive to red, green and indigo (RGB); [Enc08a]
- 3/ Deficit of the discrimination between the colors are due to non-phototransduction by the cones of one of the RGB signals. [Cro01a]

3. DESCRIPTION OF THEORY AND EXPERIMENTS

3.1. The fluctuation of the speed of light hypothesized, selects only 4 monochromatic waves as part of the visible: 764 nm, 576 nm, 432 nm and 384 nm

Based on the considerations on the fluctuation of the speed of the propagation of the wave motion [Nie18a] it is deduced that the spectrum of the visible is discontinuous because the frequencies of the visible are only four.

432 THz (768 nm) <i>magenta IR</i>	576 THz (576 nm) <i>yellow</i>	768 THz (432 nm) <i>cyan</i>	864 THz (384 nm) <i>magenta UV</i>
--	--------------------------------------	------------------------------------	--

These four frequencies are interpreted by the brain like magenta, yellow and cyan and with these three chromatic sensations (plus black and white) the brain builds all colors through the following mechanisms:

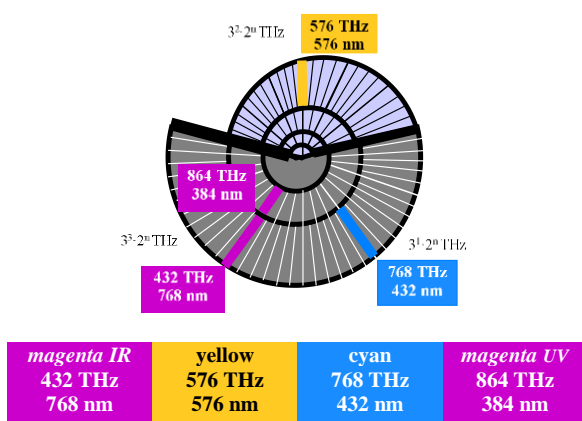
- homologation of frequency multiples,
- additive synthesis,
- on-off phototransduction system of the cones signals that produces 6 chromatic and 2 achromatic information.

3.2. The luminous frequency multiple (384 - 768 nm) is considered homologous, as is the case for the perception of height of the sounds

The four bright waves are perceived as three fundamental colors: *cyan* 432 nm, *yellow* 576 nm and *magenta UV* 384 nm and *magenta IR* 768 nm: 768 nm being the multiple of 384 nm is perceived as the same color.

In the perceptive systems that concern the distinction of the parameters of the wave motion, the multiples of frequency are considered homologous. At the perceptive system of the sound the multiples of frequency (the octaves) also calling them with the same name. For example, sounds with 32, 64, 128, 256, 512 Hz are all called Do. By analogy in this research, the bright wave of 384 nm, like the luminous wave of double length, equal 768nm, is distinguished as the *magenta*.

The logarithmic spiral in which the volutes divide the values of the frequency multiples placed on the radial axes is the model proposed in this research work to represent the increase in energy generated by the frequency of the wave motion. (Note the resemblance of structure of the cochlea.) [Nie18a]



3.3. The luminous wave consists of two heterogeneous monochromatic waves

According to this thesis with the four monochromatic waves - cyan 432 nm, yellow 576 nm, magenta UV 384 nm and magenta IR 768 nm - three bi-chromatic bright waves are formed: cinnabar red, green and indigo. In addition, the perceptual system is able to recognize the magenta components in the infrared and ultraviolet waves, present at the two extremes of the visible. To be part of the rose of the visible waves, two half-visible waves also enter for this reason: porphyry IR (semi-infrared wave consisting of waves with lengths 768 and 864 nm) and porphyry UV (semi-ultraviolet wave composed of waves with lengths 384 and 288 nm).



The author proposes the model of the double propeller shape for bi-chromatic bright wave. In the example form of the double propeller of the bi-chromatic cinnabar red wave, consisting of two monochromatic waves: magenta IR 768 nm and yellow 576 nm.



3.4. Additive synthesis generates the chromatic content of heterogeneous bi-chromatic waves and their aggregations

The bi-chromatic heterogeneous waves do not distinguish the primary constitutive elements - that is, the magenta, the yellow and the cyan - because the perception of the colors of light is based on the

mechanisms of additive synthesis, which constitutes (according to the writer) the only reason of chromogenesis. In fact, through the additive synthesis of only four bright waves all the shades of colors are obtained as the following tables describe.

a/ The synthesis of 2 single monochromatic waves creates cinnabar red, green and indigo.

768/576 nm
magenta IR/yellow
= cinnabar red

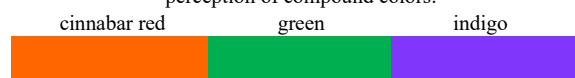
576/432 nm
yellow/cyan
= green

432/384 nm
cyan/magenta UV
= indigo




Monochromatic content of the bi-chromatic waves



Additive synthesis of monochromatic components
= perception of compound colors:

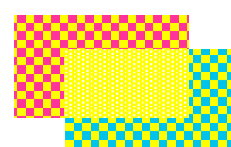


b/ The synthesis of 3 different monochromatic bright waves creates white.

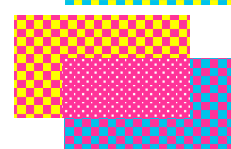
Three monochromatic heterogeneous content		Perception of compound color
		
= white		

c/ The synthesis of 2 different bi-chromatic bright waves creates following colors: magenta + white, yellow + white, cyan + white, red and violet.

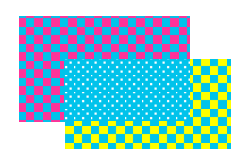
In the overlapping of a cinnabar red wave with a green bi-chromatic wave, the yellow majority monochromatic component is perceived mixed with white.



In the overlapping of cinnabar red bi-chromatic wave with a bi-chromatic wave indigo, the majority monochromatic component magenta mixed with white is perceived.



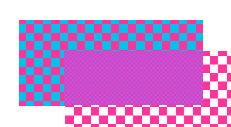
In the overlapping of indigo bi-chromatic wave with a bi-chromatic wave green, the majority monochromatic component cyan mixed with white is perceived.



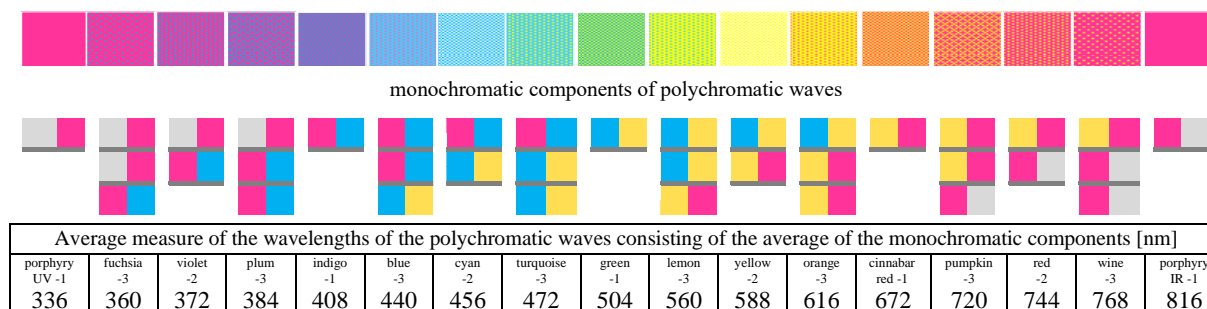
In the overlapping of a cinnabar red wave with a porphyry IR bi-chromatic wave, the color commonly called as red is perceived.



In the overlapping of a indigo wave with a porphyry UV bi-chromatic wave, the color commonly called as violet is perceived.

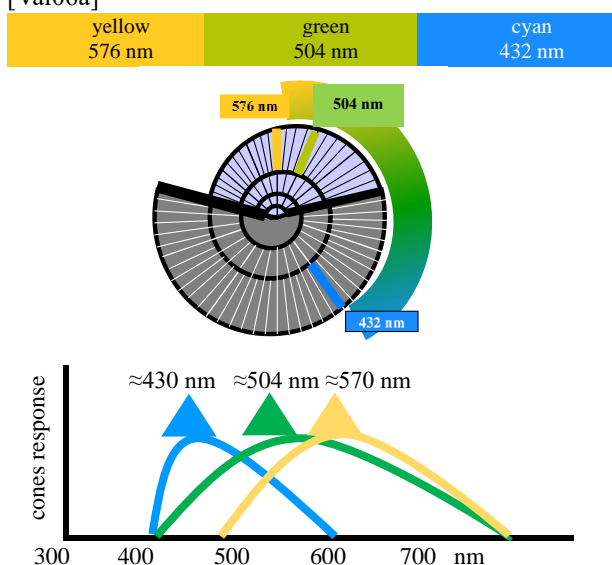


The following table illustrates the aggregations of the bi-chromatic waves: *porphyry IR* and *UV*, cinnabar red, green, indigo, their monochromatic content and the perception of the main colors of the spectrum through the described mechanism of additive synthesis and the average measure of the lengths of the polychromatic waves.



3.5. The peak of the sensitivity of the cones S, M and L corresponds to the cyan, green (cyan + yellow) and yellow respectively

The measurement of the chromatic sensitivity of the three photoreceptors of the human eyes was carried out from which the maximum sensitivity to the following lengths of the light waves is: about 550-580 nm, 500-540 nm and 420-450 nm. These wavelengths correspond to the colors: yellow, green and cyan. [Val06a]



3.6. ON - OFF process of the phototransduction of light signals

The power of the human brain allows to produce a huge amount of information using the minimum amount of data. In fact, according to the considerations that have been reached during this research, the human photoreceptors (cones and rods) detect only three sensations of color: white, cyan and yellow.

The rods detect the presence of light, translated from the visual areas of the brain in the sensation of white.

In the presence of the information of the rods on the presence of light, the three cones detect only two information: the presence of the monochromatic waves 432 nm and 576 nm, information translated by the brain respectively in sensations of colors: cyan and yellow.

- the cone S absorbs the monochromatic wave of 432 nm (cyan),
- the cone L absorbs the monochromatic wave of 576 nm (yellow),
- The cone M absorbs a green bi-chromatic wave, consisting of a 432 nm cyan monochromatic wave and a 576 nm yellow monochromatic wave.

cones:		
S	M	L
















Note that the cones measurement system is binary. It consists in reading only two cyan or yellow stimuli: individually (cones S and L) and together (cones M).

Thanks to the on-off mechanism of phototransduction of the bright signals and the additive synthesis mechanism, the four types of photoreceptors: rods, cones S, L and M are able to provide eight chromatic sensations:

- the rods with ON signal indicate the presence of light translated into sensation of white color and OFF signal indicate the absence of light translated in a sensation of black color;
- The S cones with ON signal indicate the presence of the monochromatic wave 432 nm = cyan and with OFF signal indicate the absence of cyan light translated into the sensation of the complementary color containing yellow and magenta and synthesized as cinnabar red;
- The L cones with ON signal indicate the presence of the monochromatic wave 576 nm = yellow and with OFF signal indicate the absence of yellow light translated into the sensation of the complementary color containing cyan and magenta, equal to indigo;

- The M cones with on the ON signal indicate the presence of two monochromatic waves 432 nm = **cyan** and 576 = **yellow** which through the mechanism of synthesis additive give the feeling of **green**; the M cones with OFF signal inform of the absence of **cyan** + **yellow** waves, message translated into the feeling of complementary color containing only **magenta**.

In this way each cone detects one color and its complementary tint, composing together the white.

Cones: S				M				L			
monochromatic components of waves											
ON		OFF		ON		OFF		ON		OFF	
											
perception				perception				perception			
											

Consequently, the bi-chromatic waves (indigo-1, green-1 and cinnabar red-1) and their binary aggregations (viola-2, cyan-2, yellow-2, red-2, magenta-2) are coded in the following way by the ON-OFF system:

a/ Single bi-chromatic wave.

green-1 1 bi-chromatic wave:	cones response:											
	S			M			L					
	ON	OFF		ON	OFF		ON	OFF				

cinnabar red-1 1 bi-chromatic wave:	cones response:											
	S			M			L					
	ON	OFF		ON	OFF		ON	OFF				

indigo-1 1 bi-chromatic wave:	cones response:											
	S			M			L					
	ON	OFF		ON	OFF		ON	OFF				

porphyry-1 1 bi-chromatic wave:	cones response:											
	S			M			L					
	ON	OFF		ON	OFF		ON	OFF				

b/ Binary bi-chromatic waves.

magenta-2 2 bi-chromatic wave:	cones response:											
	S			M			L					
	ON	OFF		ON	OFF		ON	OFF				

yellow-2 2 bi-chromatic wave:	cones response:											
	S			M			L					
	ON	OFF		ON	OFF		ON	OFF				

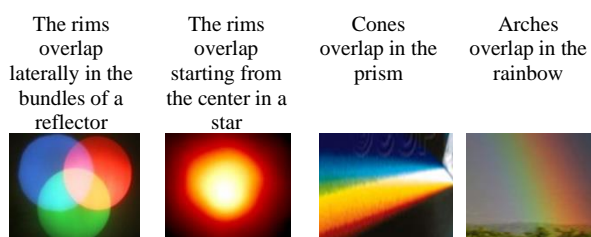
cyan-2 2 bi-chromatic wave:	cones response:											
	S			M			L					
	ON	OFF		ON	OFF		ON	OFF				

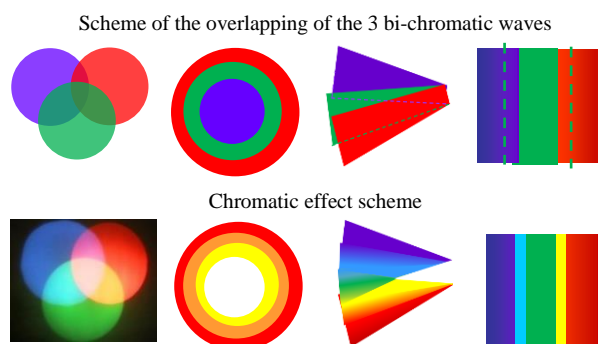
red-2 2 bi-chromatic wave:	cones response:											
	S			M			L					
	ON	OFF		ON	OFF		ON	OFF				

violet-2 2 bi-chromatic wave:	cones response:											
	S			M			L					
	ON	OFF		ON	OFF		ON	OFF				

The bi-chromatic composition of the bright waves and the on-off system of the phototransduction of the chromatic signals outlined in this book is confirmed by the classic example of additive synthesis. In fact, the overlap of **red** and **green** lights produces **yellow** + white, the overlap of **red** and **indigo** lights produces **magenta** + white, the overlap of **indigo** and **green** lights produces **cyan** + white and the overlap of **indigo**, **red** and **green** lights produces white. The traditional approach does not explain the reasons why these color changes occur.

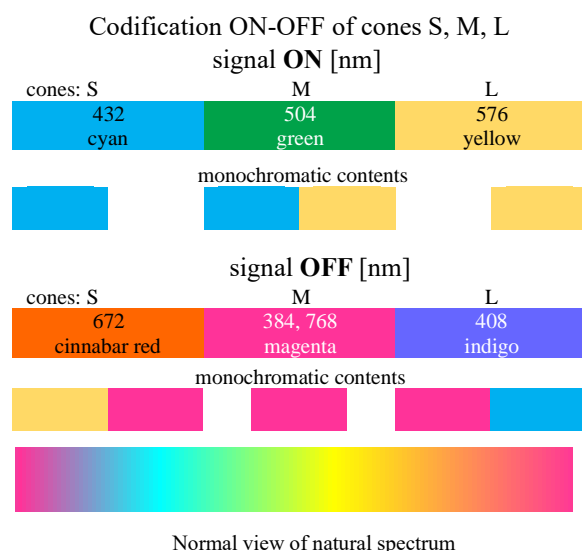
The colors of the stars and decomposition of light in the prism or in the rainbow takes place with the same mechanism of the additive synthesis generated by the overlap of the bi-chromatic waves **cinnabar red/green**, **green/indigo** and **cinnabar red/indigo**.





3.7. Deficit of color vision due to the dysfunction of the process of phototransduction of light signals

Each color we see produces generic effect on the rods (presence of light) and specific on-off responses from the three cones. The following table illustrates how the colors of the natural spectrum are coded in system shown below. (The natural spectrum differs from the spectrum produced by the RGB system from the different way of generating the purple colors which in the first case are produced by the overlapping of the *indigo-1* and *porphyry UV-1* waves and in the second case by the overlapping of the *indigo-1* and *red-2* waves.)



The following tables present:

a/ The monochromatic content of the bi-chromatic waves: single and binary of the natural spectrum and their perception;

P.UV -1	V -2	I -1	C -2	V -1	Y -2	C.R. -1	R -2	P.IR -1
Monochromatic content of the first bi-chromatic wave								
Monochromatic content of the second bi-chromatic wave								

b/ Contribution of the three cones in the formation of the chromatic map through the ON-OFF system;

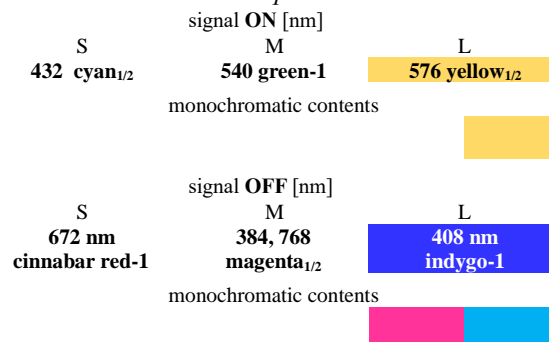
cones	P.UV -1	V -2	I -1	C -2	V -1	Y -2	C.R. -1	R -2	P.IR -1
Monochromatic content of the first bi-chromatic wave									
Decoding ON-OFF of the first bi-chromatic wave by cones:									
S									
M									
L									
Monochromatic content of the second bi-chromatic wave									
Decoding ON-OFF of the second bi-chromatic wave by cones:									
S									
M									
L									

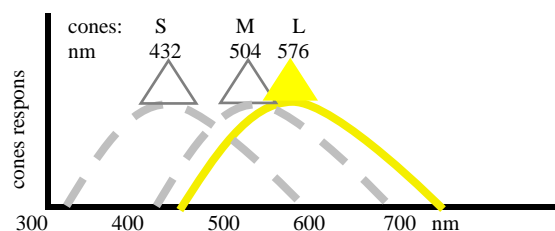
c/ Actually chromatic map generated by the additive synthesis of white.

cones	P.UV -1	V -2	I -1	C -2	V -1	Y -2	C.R. -1	R -2	P.IR -1
Monochromatic content of the first bi-chromatic wave									
Decoding ON-OFF of the first bi-chromatic wave by cones:									
S									
M									
L									
Monochromatic content of the second bi-chromatic wave									
Decoding ON-OFF of the second bi-chromatic wave by cones:									
S									
M									
L									
Percentage of the white present in the colors perceived									
	60,0	33,3	0,0	66,7	0,0	66,7	0,0	33,3	60,0
Percentage of the color present in the tint perceived									
	P.UV	V	I	C	V	Y	C.R.	R	P.IR
	40,0	66,7	100,0	33,3	100,0	33,3	100,0	66,7	40,0

The tables below illustrate the deformation of the vision of the spectrum in the case of the absence of the signals from the cones M and S, where therefore the phototransduction of the ON-OFF signals only works by the cones L. This dysfunction will be called *L-monopsia* here. (In literature this dysfunction is called *Protanopia*).

L-monopsia





The following tables present:

a/ The monochromatic content of the bi-chromatic waves: single and binary of the natural spectrum and their perception and the contribution of cones L only in the formation of the chromatic map through the ON-OFF system for the decoding of the monochromatic content of the composed colors displayed up (*L-monopsia*);

cones	P.UV -1	V -2	I -1	C -2	V -1	Y -2	C.R. -1	R -2	P.I.R. -1
	Monochromatic content of the first bi-chromatic wave								
	Decoding ON-OFF of the first bi-chromatic wave by cones:								
L									
	Monochromatic content of the second bi-chromatic wave								
	Decoding ON-OFF of the second bi-chromatic wave by cones:								
L									

c/ Actually chromatic map generated by the additive synthesis of white in the case of *L-monopsia*;

cones	P.UV -1	V -2	I -1	C -2	V -1	Y -2	C.R. -1	R -2	P.I.R. -1
	Monochromatic content of the first bi-chromatic wave								
	Decoding ON-OFF of the first bi-chromatic wave by cones:								
L									
	Monochromatic content of the second bi-chromatic wave								
	Decoding ON-OFF of the second bi-chromatic wave by cones:								
L									
	Percentage of the white present in the colors perceived								
	0,0	0,0	0,0	100,0	0,0	0,0	0,0	100,0	0,0
	Percentage of the color present in the tint perceived								
	I	I	I	C	Y	Y	Y	R	I
	100,0	100,0	100,0	0,0	100,0	100,0	100,0	0,0	100,0

The vision with the contribution of the only L cones with (*L-monopsia*) therefore deforms as follows the perception of the colors of the natural spectrum:

Normal vision



View of the natural spectrum without contribution of the cones S and M (*L-monopsia*); The contribution of the only L cones.



- Cyan is perceived white,
- Green, yellow and orange are perceived yellow,

- Red is perceived white.

This dysfunction, which according to the writer must be attributed to the functioning of only cones L is identified in literature as the dysfunction just of the cones L, responsible for the vision of red (protanopia).

The table below is compared to the graph of the deformation of the vision of the colors of the spectrum in RGB by people with protanopia (presumed dysfunction of the cones L) obtained experimentally, with the graph of the deformation of the vision of the colors of the spectrum in RGB by the *L-monopsia* (dysfunction of the cones S and M) obtained with the analytical method of this work.

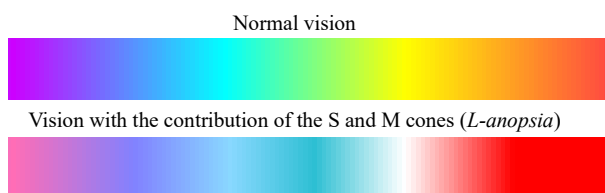
a/ Vision of the spectrum in RGB by people with protanopia. [AVI22a]	
b/ Vision of the spectrum in RGB by people with <i>L-monopsia</i> .	

Note a substantial similarity between the two draws. In the same way with this method the spectrum is obtained in the case of the dysfunction of the cones L, where the measurement of light and phototransduction is made only by the cones S and M (*L-anopsia*).

cones	P.UV -1	V -2	I -1	C -2	V -1	Y -2	C.R. -1	R -2	P.I.R. -1
	Monochromatic content of the first bi-chromatic wave								
	Decoding On-Off of the first bi-chromatic wave by cones:								
S									
M									
	Monochromatic content of the second bi-chromatic wave								
	Decoding On-Off of the second bi-chromatic wave by cones:								
S									
M									

Actually chromatic map generated by the additive synthesis of white in the case of *L-anopsia*;

cones	P.UV -1	V -2	I -1	C -2	V -1	Y -2	C.R. -1	R -2	P.I.R. -1
	Monochromatic content of the first bi-chromatic wave								
	Decoding On-Off of the first bi-chromatic wave by cones:								
S									
M									
	Monochromatic content of the second bi-chromatic wave								
	Decoding On-Off of the second bi-chromatic wave by cones:								
S									
M									
	Percentage of the white present in the colors perceived								
	0,0	60,0	0,0	60,0	0,0	100,0	0,0	0,0	0,0
	Percentage of the color present in the tint perceived								
	C.R.	P.UV	I	C	T	Y	R	R	R
	100,0	40,0	100,0	40,0	100,0	0,0	100,0	0,0	100,0



Yellow is perceived white, green is perceived turquoise (green/cyan), violet is perceived pink, the orange colors are not distinguished from the reds, the orange/yellow colors are perceived pink.

According to the conclusions of this research, this dysfunction is erroneously identified in literature such as the dysfunction of the S cones, responsible for the vision of blue (Tritanopia).

The table below is compared to the graph of the deformation of the vision of the colors of the spectrum in RGB by the Tritanopes (alleged dysfunction of the cones S) with the graph of the deformation of the vision of the colors of the spectrum in RGB by the *L-anopes* (dysfunction of the cones L) obtained with the analytical method of this work.

a/ Vision of the spectrum in RGB by people with tritanopia [Wik22a]	
b/ Vision of the spectrum in RGB by people with L-anopes.	

The similarity between the two graphic is evident. Note the absence of yellow and the presence of incongruent blue to the traditional interpretation of this dysfunction.

CONCLUSION

All these reasoning was necessary to answer our initial questions. Here are the answers formulated according to this research:

Question 1. The cyan is seen white by the Protanopes (*L-Monopes* second definition of this research) because:

- The cyan of the spectrum is made up of indigo and green, composed respectively by magenta UV/cyan and cyan/yellow monochromatic waves;

=	+	=	+

- When work only L cones the cyan-2 is perceived as an indigo-1 + yellow-1/2, then magenta-1/2 + cyan-1/2 + yellow-1/2 together form white.

=	+	=	+
= white			

Question 3. In the same way, the yellow composed of the spectrum (yellow-2) is perceived white in the case of the function only of the cones S and M.

yellow binary =2 bi- chromatic waves	mono- chromatic contents	lecture of S cones	lecture of M cones
=	+	=	+
		= white	= white

Question 2 and 4. For Protanopes (*L-Monopes*) green and yellow are seen as yellow because they are able to perceive only the yellow monochromatic component from bi-chromatic waves that make up these colors. In the same way for the Tritanopes (*L-Opsia*) cyan and green are seen cyan. They only perceive the cyan monochromatic component from the bi-chromatic waves that make up these colors.

Analysis of dysfunctions in the vision of colors can allow us to hypothesize the polychromatism of all light waves, not only of white.

A more precise knowledge of the composition and parameters of the bright waves could give a new impulse in the search for simplification and improvement of the performance of the production of colors in computers.

REFERENCES

- [Nie20a] Niewiadomska-Kaplar J., Meccanismi della visione del colore e discromatopsie, Tabedizioni, 2020.
- [AVI22a] Aviation, Why do some air forces not allow color-blind pilots?, 2022
<https://aviation.stackexchange.com/questions/16098/why-do-some-air-forces-not-allow-color-blind-pilots>
- [Nie18a] Niewiadomska-Kaplar J., Fluttuazione della velocità del moto ondulatorio, Aracne, (2018).
- [Enc08a] Encyclopedia Britannica, Health & Medicine, Anatomy & Physiology, The visible spectrum, 2008
<https://www.britannica.com/science/color/The-visible-spectrum>
- [Enc08a] Encyclopedia Britannica, Health & Medicine, Anatomy & Physiology, Color vision, 2008
<https://www.britannica.com/science/color-vision>
- [Cro01a] Cronin-Golomb A., Color Vision, Object Recognition, and Spatial Localization in Aging and Alzheimer's Disease, Functional Neurobiology of Aging, 2001.
- [Val06a] Valberg A., Light vision color, John Wiley & Sons, Hoboken, 2006.
- [Wik22a] Wikipedia, Ślepota barw, 2022
https://pl.wikipedia.org/wiki/%C5%9Alepota_barw