ISSN 2464–4617 (print) ISSN 2464–4625 (online)

Computer Science Research Notes

CSRN 3401

(Eds.)

Vaclav Skala
University of West Bohemia, Czech Republic

32. International Conference in Central Europe on
Computer Graphics, Visualization and Computer Vision
WSCG 2024
Plzen, Czech Republic
June 3 - 6, 2024

Proceedings

WSCG 2024

Proceedings

ISSN 2464–4617 (print) ISSN 2464–4625 (online)

ISSN 2464–4617 (print) ISSN 2464–4625 (online)

Computer Science Research Notes

CSRN 3401

(Eds.)

Vaclav Skala
University of West Bohemia, Czech Republic

32. International Conference in Central Europe on
Computer Graphics, Visualization and Computer Vision
WSCG 2024
Plzen, Czech Republic
June 3 - 6, 2024

Proceedings

WSCG 2024

Proceedings

ISSN 2464–4617 (print) ISSN 2464–4625 (online)

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 3401

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
Univerzitní 8, 306 14 Pilsen, Czech Republic

ISSN 2464-4617 (Print) ***ISSN 2464-4625*** (online)

CSRN 3401

International Program Committee

WSCG 2024

Abdul Karim, S. (Malaysia)
Baranoski, G. (Canada)
Barton, M. (Spain)
Benes, B. (United States)
Benger, W. (Austria)
Birra, F. (Portugal)
Bouatouch, K. (France)
De Martino, J. (Brazil)
Drakopoulos, V. (Greece)
Eisemann, M. (Germany)
Feito, F. (Spain)
Fu, Z. (China)
Galo, M. (Brazil)
Gavrilova, M. (Canada)
Gdawiec, K. (Poland)
giannini, f. (Italy)
Gudukbay, U. (Turkey)
Gunther, T. (Germany)
Hast, A. (Sweden)
Hauenstein, J. (United States)
Hitschfeld, N. (Chile)
Hu, S. (China)
Chaudhuri, P. (India)
Juan, M. (Spain)
Kurt, M. (Turkey)
Lee, J. (United States)
Liu, S. (China)
Lobachev, O. (Germany)
Manoharan, P. (India)
Marco, C. (Brazil)
Montrucchio, B. (Italy)
Oliveira, J. (Portugal)
Pan, R. (China)
Pedrini, H. (Brazil)
Platis, N. (Greece)
Puig, A. (Spain)
Radouane, K. (France)
Renaud, c. (France)
Reshetov, A. (United States)
Ritter, M. (Austria)
Rodrigues, J. (Portugal)
Sabharwal, C. (United States)
Semwal, S. (United States)
Schmalstieg, D. (Germany)
Sirakov, N. (United States)
Sousa, A. (Portugal)
Thalmann, D. (Switzerland)
Tokuta, A. (United States)
Westermann, R. (Germany)
Wu, S. (Brazil)
Wunsche, B. (New Zealand)
Zwettler, G. (Austria)

CSRN 3401

Board of Reviewers

WSCG 2024

Abdul Karim, S. (Malaysia)
Adzhiev, V. (United Kingdom)
Aguirre-Lopez, M. (Mexico)
ARORA, R. (United States)
Baranoski, G. (Canada)
Barton, M. (Spain)
Benes, B. (United States)
Benger, W. (Austria)
Benziane, S. (Algeria)
Birra, F. (Portugal)
Bouatouch, K. (France)
Cabiddu, D. (Italy)
Capobianco, A. (France)
De Martino, J. (Brazil)
Drakopoulos, V. (Greece)
Eisemann, M. (Germany)
Fedorov, V. (Russia)
Feito, F. (Spain)
Fu, Z. (China)
Fuenfzig, C. (Germany)
Galo, M. (Brazil)
Gavrilova, M. (Canada)
Gdawiec, K. (Poland)
Gerrits, T. (Germany)
giannini, f. (Italy)
Goncalves, A. (Portugal)
Grajek, T. (Poland)
Gudukbay, U. (Turkey)
Gunther, T. (Germany)
Hast, A. (Sweden)
Hauenstein, J. (United States)
Heil, R. (Sweden)
Hitschfeld, N. (Chile)
Hu, S. (China)
Hu, C. (Taiwan)
Chaudhuri, P. (India)
Juan, M. (Spain)
Kaczmarek, A. (Poland)
Kerdvibulvech, C. (Thailand)
Klimaszewski, K. (Poland)
Kurasova, O. (Lithuania)
Kurt, M. (Turkey)
Lee, J. (United States)
Lefkovits, S. (Romania)
Lengauer, S. (Austria)
Liu, S. (China)
Lobachev, O. (Germany)
Magdalena-Benedicto, R. (Spain)
Manoharan, P. (India)
Marco, C. (Brazil)
Marques, R. (Spain)
Meyer, A. (France)
Miandji, E. (Sweden)
Miller, M. (Germany)
Montrucchio, B. (Italy)
Müller, S. (Germany)
Nawfal, S. (Iraq)
Nguyen, S. (Viet Nam)
Oliveira, J. (Portugal)
Pagnutti, G. (Italy)
Pan, R. (China)
Papaioannou, G. (Greece)
Pedrini, H. (Brazil)
Pereira, J. (Portugal)
Perez-Diaz, S. (Spain)
Pintus, R. (Italy)
Platis, N. (Greece)
Pombinho, P. (Portugal)
Puig, A. (Spain)

Radouane, K. (France)
Raidou, R. (Austria)
Ray, B. (India)
Renaud, c. (France)
RESHETOV, A. (United States)
Ritter, M. (Austria)
Rodrigues, N. (Portugal)
Rodrigues, J. (Portugal)
Romanengo, C. (Italy)
Sabharwal, C. (United States)
Sacco, M. (Italy)
Sardana, D. (United States)
Satpute, V. (India)
Semwal, S. (United States)
Seracini, M. (Italy)
Shendryk, V. (Ukraine)
Schmalstieg, D. (Germany)
Sirakov, N. (United States)
SLUZEK, A. (Poland)
Sousa, A. (Portugal)
Tandianus, B. (Singapore)
Thalmann, D. (Switzerland)
Tokuta, A. (United States)
Tourre, V. (France)
Tytkowski, K. (Poland)
Wang, L. (United States)
Westermann, R. (Germany)
Wu, S. (Brazil)
Wunsche, B. (New Zealand)
Yoshizawa, S. (Japan)
Zwettler, G. (Austria)

CSRN 3301

Computer Science Research Notes

WSCG 2023 Proceedings

Contents

KEYNOTE Page

A Synergy of Computer Graphics and Generative AI: Advancements and Challenges 1
Gavrilova,M.

FULL papers

The Impact of the Number of k-Means Clusters on 3D Point Cloud Registration 3
Ankomah,P., Vangorp,P.

SEL-CIE: Self-Supervised Learning Framework for CIE-XYZ Reconstruction from Non-Linear sRGB Images 13
Barzel,S., Lindenbaum,O., Averbuch,A.

Automated Bioacoustic Monitoring for South African Bird Species on Unlabeled Data 23
Doell,M., Kuehn,D., Suessle,V., Burnett,M.J., Downs,C.T., Weinmann,A., Hergenroether,E.

Beyond the Benchmark: Detecting Diverse Anomalies in Videos 33
Arad, Y. , Werman, M.

PixLabelCV - Labeling images for semantic segmentation fast, pixel-precise and offline 47
Schraml, D., Trambickii, K., Notni, G.

Show Me the GIFference! Using GIFs as Educational Tools 57
Amabili,L., Gröeller,M. E., Raidou,R. G.

Attention-Aware DAE for Automated Solar Coronal Loop Segmentation 67
Dhaubhadel,P.M., Lee,J.K., Tian,Q.

Combining bidirectional path tracing, DDGI, and ReSTIR to improve real-time rendering quality 77
Kretzschmar, V., Meyer, B., Hergenröhter, E.

Empathy Training using Virtual Environments 87
Jackson, R; Semwal, S.K.

Page

Texture-based Global Illumination for Physics-Based Light Propagation in Interactive Web Applications 97
Roth,A., Cords,H.

Improving Image Reconstruction using Incremental PCA-Embedded Convolutional Variational Auto-Encoder 107
Azizi,A.,Charalambous,P.,Chrysanthou,Y.

Reevaluation in Rule-Based Graph Transformation Modeling Systems 117
Gaide,M., Marcheix,D., Arnould,A., Skapin,X., Belhaouari,H., Jean,S.

PLOD: Point cloud level of detail for polygon mesh 129
Konks,E., Shcherbakov,A.

Towards Multi-Species Animal Re-Identification 137
Fruhner,M.,Tapken,H.

Texture reading optimization for 2 dimensional filter based graphics algorithms 147
Zubatov K., Shcherbakov A.

An automated Pipeline to bring NeRFs to the Industrial Metaverse 157
Schleise, S., Hahne, U., Lindinger, J.

Segmentation of discrete surfaces into plane segments based on a distance map 167
Drieu La Rochelle,L., Zrour,R., Largeteau-Skapin,G., Andres,E., Tankyevych,O., Cheze Le Rest,C.

Measuring the Influence of Alcohol Consumption on Presence in Virtual Reality 177
Genz,F., Dreer,F., Kroetz,F., DAmelio,M., Kranzlmueller, D.

Automatic data generation of incorrect image-text pairs for effective contrastive learning of CLIP model 187
Tagami,R., Kobayashi,H., Akizuki,S., Hashimoto,M.

Color Quality Comparison in Spectrally (Un)Correlated Random Field Models 197
Havlicek, M., Haindl, M.

3D Multi-Criteria Design Generation and Optimization of an Engine Mount for an Unmanned Air Vehicle Using a Conditional Variational Autoencoder 207
Eilermann, S., Petroll, C., Höfer, P., Niggemann, O.

Adapting Virtual Reality Training Applications by Dynamically Adjusting Visual Aspects 217
Genz,F., Kranzlmueller,D.

AI-based Density Recognition 227
Müller, Simone, Kolb Daniel, Müller Matthias, Kranzlmüller Dieter

Page

Cocoa beans moisture content prediction using Machine Learning Model, based on the color image features 237
Ako,J.E, NZi,C.E., Kpalma,K.

Deep learning-based classification of breast tumors using selected subregions of lesions in sonograms 247
Schmidt, C., Overhoff, HM

LVCluster: Bounded Clustering using Laguerre Voronoi Diagram 255
Ohi, AQ., Gavrilova, ML.

From Sources to Solutions: Enhancing Object Detection Models through Synthetic Data 263
Bartolovic,E., Höfer,T., Hage,C., Nischwitz,A.

ECNNXAI: Ensembled CNNs with eXplainable Artificial Intelligence for Colon Histopathology Image Classification 273
Qadri, J., Jothi, J. A. A.

Polychromatism of all light waves and a new approach to the interpretation of fluorescence mechanisms; Idea of the fluorescent color monitor 283
Justyna Niewiadomska-Kaplar

Impact of Calibration Matrices on 3D Monocular Object Detection: Filtering, Dataset Combination and Integration of Synthetic Data 291
Evain,A., Khemmar,R., Orzalesi,M., Ahmedali,S.

Spotlight Control for Real-Time Targeting 299
Heinemann,M., Kernbauer,T., Fleck,P., Arth,C.

Real-Size Experience for Virtual Try-On 307
Wu, F., Chen, W., Dellinger, A., Cheng, H.

Genetic Subdivision Curve and Surface Reconstruction 315
Komar, A., Riegler, M., Preiner, R., Augsdoerfer, U.

LatEd: A Geometric Latent Vector Editor 325
Komar, A., Kammerer, M., Barzegar Khalilsaraei, S., Augsdoerfer, U.

Brain Computer Interfacing with a Virtual Environment 331
Gamillscheg,F., Ruprecht,I., Settgast,V., Pietroszek,K., Augsdörfer,U.

A Comparative Study of Convex Combination and Inner Ordinate Methods For Scattered Data Interpolation Using Quartic Triangular Patch 339
Owen Tamin, Samsul Ariffin Abdul Karim, Mohammad Khatim Hasan

Lightweight single image dehazing utilizing relative depth information 349
Panagiotis Frasiolas, Asterios Reppas, Konstantinos Konstantoudakis, Dimitrios Zarpalas

Correlational Data Visualizations with Colored Bar Charts 359
Alwuqaysi,B., Abdul-rahman,A., Borgo,R.

POSTERS

Sharpness Measurement by Edge-related Frequency Components 371
Lee, Y.K., Ding,J.J.

The Disease of the Canine Eye - From Image to Diagnosis Using AI 377
Buri?, M., Ivasic-Kos, M., Martincic-Ipsic, S.

Study of Evolution in Virtual Worlds 381
Ben Amar, H., Masashi, O.

Analysis of natural lighting conditions for the digitization of artwork in an art gallery interior 391
Drofova, I., Adamek, A.

The color recognition methods for the active markers in the motion capture system, using various techniques, including ML 395
Mrzyglod J., Kowalski P.

Dynamic Many-Light Importance Sampling for Real-Time Ray Tracing 403
Freire,P.,Pereira,J.

A Novel Silhouette Extraction Method for Binary Images Based on the Wall-Follower Algorithm 407
Aguirre-López,M.A., Izaguirre-Prado, O.E., Soto-Villalobos,R., Hueyotl-Zahuantitla,F.

A Synergy of Computer Graphics and Generative AI: Advancements and Challenges

Marina Gavrilova

University of Calgary
2500 University Drive NW,
Calgary, AB, Canada T2N1N4

mgavrilo@ucalgary.ca

ABSTRACT

A traditional computer graphics domain has received an unprecedented boost from the newest developments in generative Artificial Intelligence (GenAI). It affects all areas: from image generation, to face recognition, to object detection, to aerial surveillance, to autonomous car vision systems. The newest deep learning architectures make it possible to generate new images from texts, to apply styles to portraits, to de-identify facial images, and to recognize human and objects in videos. This keynote will delve into some of the most exciting applications in medical AI diagnostics, human face recognition and aesthetics domains, while making a strong case for resulting image authenticity, bias mitigation, and trust.

Keywords

Generative AI, computer graphics, deep learning, biometrics, digital human, authenticity, bias, trust.

1. INTRODUCTION

The next generation Generative Artificial Intelligence (GenAI) algorithms have dramatically transformed not only research landscape, but the way society communicates and functions. At the same time, trust in AI-powered decision making, bias mitigation and responsible computing emerged as the key societal challenge of the 21st century [Gav23]. According to researchers, practitioners, industry experts, and the public, it is imperative to build in safety mechanisms in the ways AI is utilized in commercial and public domains [Anz22, Cha21]. Computer Graphics domain has witnessed significant transformation due to the development of advanced deep learning architectures. Text to image and reverse conversion allows for better storytelling, while style application to art works or advertisements inspire creative designs never possible before [Ska23].

2. MOTIVATION

The synergy between image processing and deep learning resulted in new advancements in medical diagnostics, patient-focused personalized healthcare delivery, and new drug developments. Another domain that benefitted from the newest deep learning architectures is information security. Biometric authentication systems based on convolution neural networks, transformer-type architectures, graph-based deep learning, and adversarial neural networks allow not only faster training with more accurate user identity authentication, but also allow for higher degree of privacy protection through situation awareness [Pau14] and facial de-identification [Gav21]. Those developments lead to the need for especially careful consideration of user's data privacy, ethical use of data, and bias free data collection. The ease of accessing abundant social medial data, which is frequently used for targeted advertisement, service delivery, political campaigns, and e-commerce, augments the need for protecting user's rights and enforcing ownership of data [Gav17]. At the same time, more and more researchers are starting to pay attention to the development of not only accurate and reliable decision-making systems, but to the urgent demand for ethical, responsible, and trustworthy computing [Col18, Lin20, Lyu21].

3. RECENT DEVELOPMENTS

This keynote starts with presenting state of the art in generative AI developments and the transformative changes in computer graphics, image processing, human aesthetics and biometric authentication domains. It proceeds with introducing concepts related to data representation, feature engineering, prediction and classification outcomes of AI models. It then introduces the concept of trustworthiness in decision-making as the ability of computer systems to perform a real-world task reliably, with decision being transparent, processes being explainable and algorithms being bias-free. Data privacy is discussed along with the issues of data authenticity, ownership, and ethics.

https://www.doi.org/10.24132/CSRN.3401.1

4. CONCLUSIONS

This invited keynote presents an overview of most recent advancements and challenges associated with rapid adoption of generative AI breakthrough in modern society. It elaborates on technological innovations that made broad adaptions of deep learning into consumer products and services possible, describes key research in the domains of fair, trustworthy and explainable decision-making systems powered by generative AI, and provides examples from the domains of cybersecurity, biometric and computer graphics. The keynote is based on the innovative research conducted at the Biometric Technologies Laboratory of the University of Calgary, Canada and supported by the Research Excellence Chair in Trustworthy and Explainable AI awarded to Prof. Marina Gavrilova.

5. ACKNOWLEDGMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) Discovery Grant, by the Innovation for Defense Excellence and Security Network (IDEaS) and by Research Excellence Chair Program at the University of Calgary.

6. BIOGRAPHY

Marina L. Gavrilova is a Full Professor, an Order of the University of Calgary Inductee and a Head of the Biometric Technologies and the SPARCS Laboratories in the Faculty of Science. Her publications include over 300 refereed articles, edited special issues, books and book chapters in the areas of machine learning, information fusion, knowledge discovery and cybersecurity. She serves as a Founding Editor-in-Chief of Transactions on Computational Science Journal, Springer and an Editor-in-Chief of the International Journal of Digital Human, Inderscience. As a globally renown award-winning researcher and educator, Dr. Gavrilova has given over 50 keynotes, invited lectures and tutorials at major scientific gatherings worldwide. Dr. Gavrilova is a passionate advocate of equity, diversity and inclusion in academia, industry, and society.

7. REFERENCES

[Anz22] F. Anzum, A.Z. Asha and M. Gavrilova, Biases, Fairness, and Implications of Using AI in Social Media Data Mining, International Conference on Cyberworlds, IEEE, pp 251-254, October 2022

[Cha21] R. Chatila and F. Rossi, How to Address Artificial Intelligence Fairness, World Economic Forum, Davos, Switzerland, January 2021

[Col18] J. Collmann and S.A. Matei, Ethical Reasoning in Big Data, Springer, 2018

[Gav17] M. Gavrilova, F. Ahmed, S. Azam, P. Paul, M. Sultana, W. Rahman, F. T. Zohra, Emerging Trends in Security System Design using the Concept of Social Behavioral Biometrics, in Information Fusion for Cyber-Security Analytics, Studies on Computational Intelligence, Ch 10, vol. 691, pp. 229-252, Springer, 2017

[Gav21] M. Gavrilova, F. Ahmed, H. Bari, R. Liu, T. Liu, Y. Maret, B. Sieu, and T. Sudhakar, Multi-modal Motion Capture Based Biometric Systems for Emergency Response and Patient Rehabilitation, in Research Anthology on Rehabilitation Practices and Therapy, Ch. 7, pp. 653-678, IGI Global, 2021

[Gav23] M. Gavrilova, Responsible Artificial Intelligence and Bias Mitigation in Deep Learning Systems, 2023 27th International Conference Information Visualisation, pp. 329-333, IEEE, 2023

[Lin20] P. Linardatos, V. Papastefanopoulos, S. Kotsiantis Explainable AI: A Review of Machine Learning Interpretability Methods. Entropy, vol 23, iss. 1, (18), pp 1-45, 2020

[Lyu21] D. Lyu, F. Yang, H. Kwon, W. Dong, L. Yilmaz, B. Liu, TDM: Trustworthy Decision-Making via Interpretability Enhancement, IEEE Transactions on Emerging Topics in Computational Intelligence, pp. 1-12, 2021

[Pau14] P. Paul, M. Gavrilova, S. Klimenko, Situation Awareness of Cancellable Biometric System, The Visual Computer, Vol 30, Iss. 9, pp. 1059-1067, 2014

[Ska23] V. Skala, Multispectral Image Generation from RGB Based on WSL Color Representation: Wavelength, Saturation, and Lightness, Computers, Vol 12, No 9, 182, 2023

The Impact of the Number of *k*-Means Clusters on 3D Point Cloud Registration

Peter Ankomah
Manchester Metropolitan University
Department of Computing And Maths
All Saints Building
United Kingdom, Manchester
p.ankomah@mmu.ac.uk

Peter Vangorp
Utrecht University
Department of Information and
Computing Sciences
Utrecht, The Netherlands
p.vangorp@uu.nl

ABSTRACT

Point cloud registration plays a crucial role in many applications, from robotics and autonomous navigation to medical imaging and 3D scene reconstruction. While the Iterative Closest Point (ICP) algorithm is a well-known shape registration choice, its efficiency and accuracy can be affected by the vast search space for point correspondences. *k*-means clustering emerges as a promising solution for partitioning the search space into smaller clusters to reduce the computational complexity and increase the performance of the matching. However, the number and size of these clusters and how they affect the registration remains a critical and yet not fully explored factor. This paper delves into the relationship between the number of *k*-means clusters and point cloud registration accuracy. To determine the effect of the number of *k*-means clusters on registration accuracy and efficiency and to understand any emerging pattern, *k*-meansICP is developed to use the *k*-means algorithm to cluster the correspondence search space. Two sets of 3D molecular shapes with differing complexities are matched using initial rotation angles 15, 30, and 60 degrees with 2 to 10 *k*-means clusters. The results are then compared with the original ICP algorithm.

Keywords

3D Point Cloud, 3D Shape Registration. Iterative Closest Point, *k*-Means Clustering

1 INTRODUCTION

Point cloud registration is one of the most important yet challenging areas of computer vision [Pot06, Goj20, MA23]. This field is related to tasks in several domains such as molecular biology for characterising biological interactions [Ovr23], medical image processing [Wu18] for medical image visualisations, robotics [Lyu24, Kri15] for 3D object recognition through reconstruction, and human pose estimation and tracking [Che11], and for optimising cutting tool positioning in the context of Computer Numerical Control (CNC) machining [Bo19].

The iterative closest point algorithm (ICP) [Bes92] is the most popular point cloud registration algorithm [Lyu24, Li22] because of its simplicity and modular nature, allowing enhancements to different aspects of the registration process as required. ICP aims to align two point clouds by finding a transformation that minimises the distance between the point clouds. The algorithm repeatedly alternates between finding the closest point in the target point cloud for each point in the source point cloud and computing a transformation that minimises the distance between the corresponding points [Zha22].

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 Limitations of existing solutions

ICP requires a good initial transformation [Lyu24] to converge to a good solution. The algorithm can perform poorly with a large initial transformation resulting in a less optimal convergence [Wu18]. The presence of outliers (non-uniform or non-conforming points), noise produced from point cloud acquisition devices [Du15], and partial overlaps, such as registering subsets of a point cloud, can also decrease the efficiency of the registration process driven by false matches [Pom13]. The nature of ICP as an optimisation algorithm is to find approximate solutions to the registration problem by converging to a locally optimal solution. This makes the quality of the registration results heavily dependent on the initial transformation. An ICP registration process with less optimal initial transformation can lead to incorrect correspondence, and smaller transformation convergence regions making the registration process unstable [Sal24]. In some instances, ICP may not converge at all.

1.2 Outline of this work

This paper evaluates how the number of clusters in a *k*-means clustered point cloud, and indirectly the number of points in each cluster, affects the registration accuracy. An experiment is set up using two 3D molecular structures from the protein data bank [Ber00], 2HAX and 2JKF. These structures represent different forms of the common cold virus's protein molecule. The experiment registers each structure to a copy of itself by partitioning the search space from 2 to 10 clusters, each time setting an initial rotation of 15, 30, or 60 degrees along one axis. Partitioned clusters reduce the point correspondence search space. This reduction then decreases the computational complexity of the registration process.

Section 3 details an implementation of ICP [Bes92] using *k*-means clustering to reduce the search space. Section 4 details the experimental setup and design as well as the results, analysis, and implications. Section 5 discusses the findings from the experiment and the performance of the algorithms. Section 6 presents the conclusion of this paper with the reiteration of the problems with ICP [Bes92], how a *k*-means based implementation improves the matching, further limitations of this approach, and our ongoing and future work.

2 RELATED WORK

2.1 The ICP Algorithm

The original ICP algorithm [Bes92] repeatedly computes the transformation that aligns two point clouds. The algorithm does a correspondence search by finding the closest point in the target point cloud for each point in the source point cloud. A transformation computed from these correspondences is then applied to the position of the source point cloud to bring it in alignment with the target point cloud. The algorithms do the next correspondence search from this transformed source point cloud until the convergence criteria are reached, which is a set number of iterations or a set mean squared error threshold.

2.2 Improvements to the ICP Algorithm

Hundreds of variations of ICP have been developed demonstrating the need for improved versions of the algorithm to handle the vast and diverse datasets [Pom13, Rus01]. The modular nature of ICP allows improvements to be made at different stages including the closest point search, more elaborate distance metrics than the Euclidean distance between points, and weighting of matching point pairs as a way of rejecting less likely matches [Wan17]. One recent approach to using ICP to match protein structures focused on enhancing the registration quality using metadata knowledge of the point cloud [Ank20]. This included extracting rotation-invariant features such as the *k* nearest neighbours of each point, and other available point labelling information to aid in the correspondence search process.

Algorithm 1 Iterative Closest Point algorithm

```
function ITERATIVE CLOSEST POINT($P$, $X$)
    $P_0 \leftarrow P$
    for iteration $i := 0$ to $i_{\max}$ do
        closest points $Y_i$
            $\leftarrow$ CLOSEST POINT SEARCH($P_i$, $X$)
        transformation $M_i$, MSE $d_i$
            $\leftarrow$ REGISTRATION($P_0$, $Y_i$)
        $P_{i+1} \leftarrow$ TRANSFORM($M_i$, $P_0$)
        if change in MSE $d_{i-1} - d_i <$ threshold then
            terminate the for loop
        end if
    end for
    return $P_{i+1}, M_i, d_i$
end function
```

2.3 Use of *k*-Means Clustering

The *k*-means clustering algorithm [Kri99] is an iterative optimisation algorithm to cluster any point cloud into a chosen number of *k* clusters. The algorithm is domain-independent making it an ideal choice for use in specific tasks in pattern recognition and machine learning.

The algorithm accepts the desired number *k* of partitions and then clusters the data points into *k* clusters, assigning each point to the closest cluster. This is done by computing the squared Euclidean distance between each data point and the centroid of each cluster, and then assigning each point to the cluster with the closest centroid [Iko23]. There are several variants of the *k*-means clustering algorithm based on modifications to characteristics such as the distance measure, ways to reduce the number of distance calculations, and the reduction of the dimension of the data set [Iko23]

PF-ICP [Sal24] was proposed as an improvement to the traditional ICP [Bes92] algorithm. PF-ICP uses *k*-means clustering to help correctly estimate the transformation during registration. The research utilises the $k-means$ clustering method to guide the initialisation of the transformation. The $k-means$ algorithm was employed to segment the data point cloud into clusters, and the initial transformation was obtained by doing a correspondence match of the centroids of the generated clusters. This process was found to reduce the computational complexity and increase registration speed. The transformations computed from matching the centroids of the clusters were then used as the initial transformation for the point cloud registration. In PF-ICP [Sal24], $k-means$ is primarily used for clustering and estimating the initial transformation, whilst $k-means$ partitions the search space into clusters to reduce the computational complexity of the registration process by reducing the correspondence search space.

3 THE *k*-MEANSICP ALGORITHM

Our algorithm is based on the original ICP algorithm [Bes92] (Algorithm 1). The ICP algorithm finds the correspondence for each point by computing the distance from each point in the model point cloud to each point in the data point cloud. This process is computationally expensive. Several measures have been proposed to decrease the computational cost of the distance calculation such as by using *k*d-tree [Bes92], and partitioning the search space using metadata of the points [Ank20]. [Kan02] proposed an efficient *k*-means clustering algorithm that leverages Voronoi diagrams to significantly reduce time complexity, resulting in substantial accelerations compared to standard *k*-means clustering methods.

Our algorithm improves computational cost by partitioning the correspondence search space using the *k*-means clustering algorithm [Kri99]. This means that each point does two sets of distance calculations: The first is to pick the closest cluster by finding the closest centroid to the point. The second is to find the distance from the point to each point belonging to the chosen cluster. This implies every point does point-to-cluster and then point-to-point distance calculations. The number of clusters (k) is a user-configurable constant that can be increased or decreased based on the performance of the registration. Although there are two sets of distance calculations involved in our algorithm, the computational complexity is reduced.

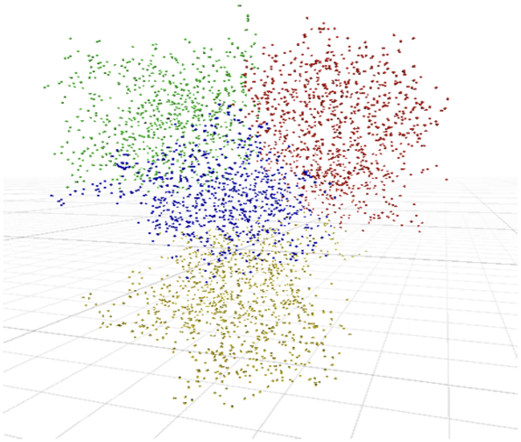


Figure 1: 3D point cloud clustered into 4 colour coded regions

Assume that the point cloud P_i is clustered into k clusters with centroids X_c, where $c \in 1 \dots k$. For each point $p \in P_i$ the algorithm first finds point p's closest cluster C based on the distances between p and the cluster centroids. The algorithm then finds the closest point to p among the points $x \in C$ by performing a naive search comparing the distance d of every point x and finding the minimum distance d_{min}. The closest point y is appended to the list of closest points Y_i as the correspondence for p.

Algorithm 2 *k*-meansICP closest point search

```
function CLOSEST POINT SEARCH($P_i$, $X$)
        ▷ $X$ is clustered into $k$ clusters with centroids $X_c$
    closest points $Y_i \leftarrow$ empty list
    for all points $p$ in $P_i$ do
        cluster $C \leftarrow$ CLOSEST CLUSTER($p$, $X_c$)
        closest point $y \leftarrow$ empty
        closest point distance $d_{min} \leftarrow \infty$
        for all points $x$ in $C$ do
            distance $d \leftarrow$ DISTANCE($p$, $x$)
            if $d < d_{min}$ then
                closest point $y \leftarrow x$
                closest point distance $d_{min} \leftarrow d$
            end if
        end for
        append closest point $y$ to closest points $Y_i$
    end for
    return $Y_i$
end function
```

3.1 Computational complexity

k-meansICP leverages the clustering information to reduce the search space. It achieves this by first finding the nearest cluster for each data point (p) and then searching only within that cluster (C) for the closest neighbour (x). This approach reduces complexity by a factor depending on the number of clusters and their uniformity.

k-means clustering tends to create clusters of equal volume. The effectiveness of *k*-meansICP depends on the distribution of points within the clusters. Evenly distributed points lead to equal-sized clusters in terms of cardinality. An evenly distributed point cloud with cardinality N would be partitioned into k clusters of cardinality N/k. For each of the N points we first find the nearest of the k cluster centroids using $O(k)$ distance calculations, and then then nearest of the on average N/k points in that cluster using $O(N/k)$ distance calculations. Hence the computational complexity of *k*-meansICP becomes $O(N(k+N/k))$. In theory the optimal choice of k would be $k=\sqrt{N}$ to reduce the total computational complexity to $O(N^{1.5})$, which would be a significant reduction from $O(N^2)$ for the original ICP algorithm. More generally, for non-uniform point clouds the computational complexity of *k*-meansICP depends on the actual cardinalities of the clusters and on the probability that each cluster is selected as the closest to the point p.

However, the accuracy of the matches might be affected, particularly for points near cluster boundaries. This is because k-meansICP limits the search space to the chosen cluster C. While C is the cluster with the closest centroid to p, it may occur that the actual closest point is not in C but in a different cluster. In such cases, a suboptimal closest point will be chosen from C. The iterative nature of the ICP algorithm makes it robust to small amounts of such suboptimal correspondences. More suboptimal correspondences may delay or even prevent convergence. To keep the amount of suboptimal correspondences small we keep k relatively small, in practice $k = 2$ to 10. This is typically less than optimal for the theoretical computational complexity, but we have found that it is a better trade-off in terms of convergence in practice.

4 EXPERIMENT

k-meansICP was implemented in Microsoft C#. Unity3D was used to visualise the registration simulation and generate the analysis data.

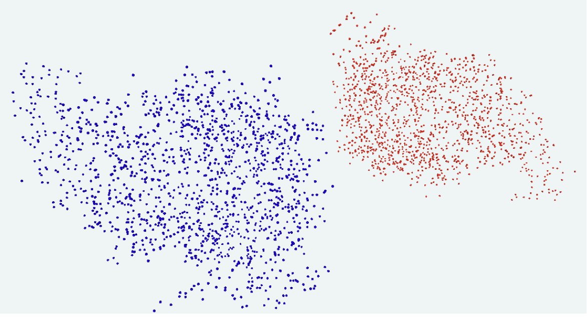


Figure 2: 3D point cloud for molecules 2JKG (left) and 2JKF (right)

4.1 Experimental Design

This experiment investigates the performance of the k-meansICP algorithm for protein structure matching. The focus of the experiment was on how the number of k-means clusters [Kri99] can affect registration accuracy and convergence.

4.2 Data Acquisition and Pre-Processing

For this study, two protein structures from the Protein Data Bank (PDB) [Ber00] serve as the source for protein structure data. The PDB IDs are 2JKF and 2JKG (provided for reproducibility) consisting of 1420 and 1452 atoms respectively. Each protein structure shape will be matched to a rotated version of itself. Each structure was imported into Unity using the CellUnity [Geh15] package to facilitate the visualisation and manipulation of protein structures. The package automatically generated a 3D representation based on the coordinates of each atom in the PDB file. Atoms were visualised as spheres, allowing for spatial analysis of the protein molecule.

4.3 Experimental Setup

The experiment evaluated the performance of the ICP and k-meansICP algorithms under varying initial rotation angles and different numbers of clusters for the k-meansICP algorithm. Specifically, the source and target copy of a protein structure were aligned by their centres of mass, followed by giving one copy a starting rotation along one axis at predefined angles (15°, 30°, and 60°). Each test case was run with 100 iterations for k-meansICP and original ICP. However, in the case of k-meansICP, each test case was also run for $k = 2$ to 10 clusters.

4.4 Rationale

Large initial rotations pose a challenge for ICP algorithms. The introduction of an initial rotation angle was to add some complexity to the registration process.

The original ICP algorithm served as the baseline for comparison, isolating the effects of the proposed improvements within the k-meansICP. Integrating k-meansICP with other ICP variants would potentially introduce confounding factors from their inherent improvements, hindering the accurate measurement of our specific contributions related to the number of clusters and the initial rotation angles.

4.5 Rotational angles and number of clusters

While both algorithms consider translation during the matching process, this experiment primarily focused on rotational alignment and cluster sizes. The pre-alignment of structures at their centres of mass minimizes the impact of the translation vector and is a common pre-processing step in ICP.

5 ANALYSIS AND EVALUATION

To assess and compare the performance of k-meansICP with each number of clusters, as well as the original ICP, two types of graphs are presented. The convergence graphs show the mean squared alignment error at each iteration. This data is generated at the end of each iteration and it presents an analysis of the gradual progress of the registration process towards convergence. It can also show the iteration count at convergence, as well as how smooth the transition is towards convergence.

The match quality graphs show the cumulative histogram of the number of points that have a root mean squared alignment error less than a value on the horizontal axis. This data represents the final quality of the match when the registration process is completed. By looking at the cumulative number of points at a particular root mean square error, we can analyse the effectual quality of the match. A good match quality would have a steep rise to nearly 100% at small alignment errors. The vertical line that sometimes appears at the right edge of a match quality graph is an artefact of Matplotlib [Hun07] that indicates the maximum alignment error, where the cumulative histogram reaches 100% of points. Note that the horizontal axis is scaled to fit the range of alignment errors. If the match is almost perfect and the maximum alignment error is very small, e.g., 1e-8, then the match quality curve has an irregular shape caused by floating point precision and does not represent significant alignment errors.

5.1 ICP

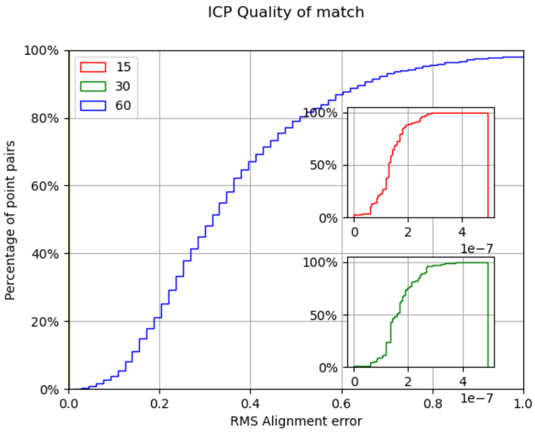


Figure 3: Match quality results for molecule 2JKF using ICP at initial rotation angles 15°, 30°, and 60°

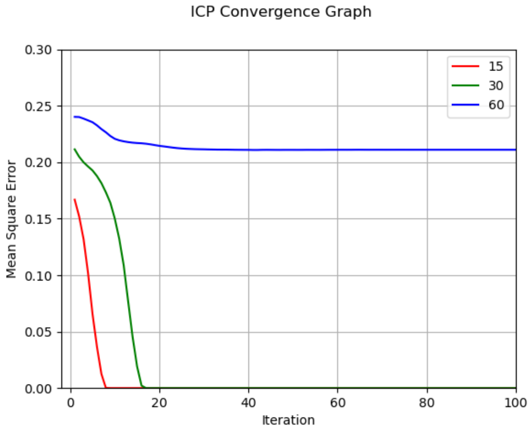


Figure 4: Convergence results for molecule 2JKF using ICP at initial rotation angles 15°, 30°, and 60°

From the matching results for ICP for molecule 2JKF (Figure 3), we realize a good match for initial rotation angles 15 and 30 degrees and a comparatively worse one for angle 60 degrees. This is typical of the ICP algorithm's behaviour of worse convergence with a poor initialisation and thus requiring a good initial transformation. The convergence graph (Figure 4) confirms that performance with the 15 and 30-degree initial rotation angles converging faster at approximately 10 iterations and 18 iterations respectively, whereas the angle 60 degrees has reached a plateau after approximately 30 iterations with a substantial remaining alignment error.

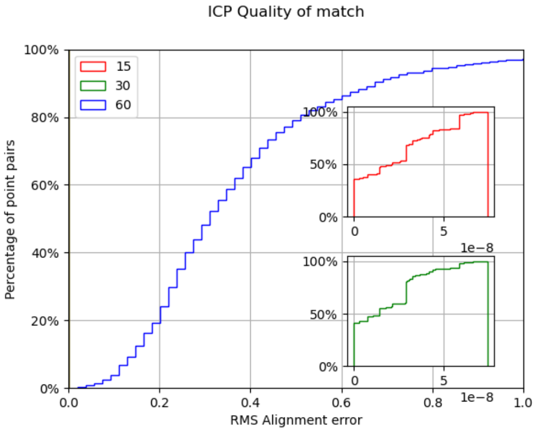


Figure 5: Match quality results for molecule 2JKG using ICP at initial rotation angles 15°, 30°, and 60°

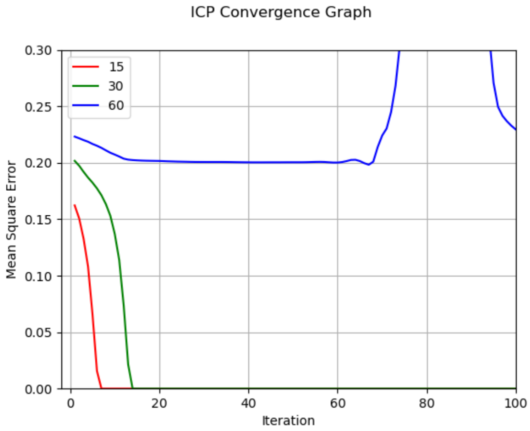


Figure 6: Convergence results for molecule 2JKG using ICP at initial rotation angles 15°, 30°, and 60°

Similarly, for molecule 2JKG, we realize a good match for initial rotation angles of 15 and 30 degrees and a worse match for the angle of 60 degrees (Figure 5). The convergence graph (Figure 6) shows similar convergence results for 15 and 30 degrees. For an initial rotation angle of 60 degrees, the convergence is much worse compared to 2JKF, even showing intermittent divergence around iterations 70–100.

5.2 *k*-MeansICP

It can be observed that the match quality of *k*-meansICP with 2 clusters (Figure 7) was comparable to ICP (3) but only for the 60 degrees initial rotation. ICP performed much better at converging at a lower mean squared

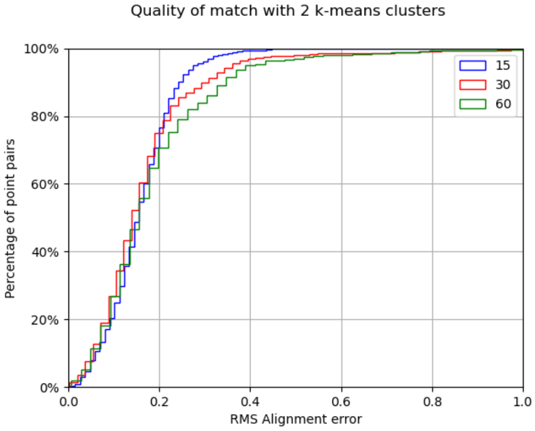


Figure 7: Match quality results for k-meansICP with $k = 2$ clusters

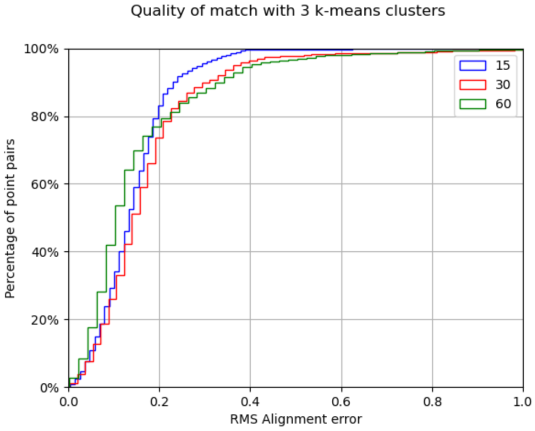


Figure 8: Match quality results for k-meansICP with $k = 3$ clusters

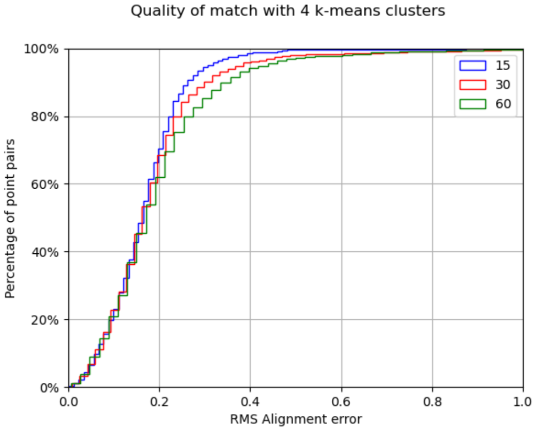


Figure 9: Match quality results for k-meansICP with $k = 4$ clusters

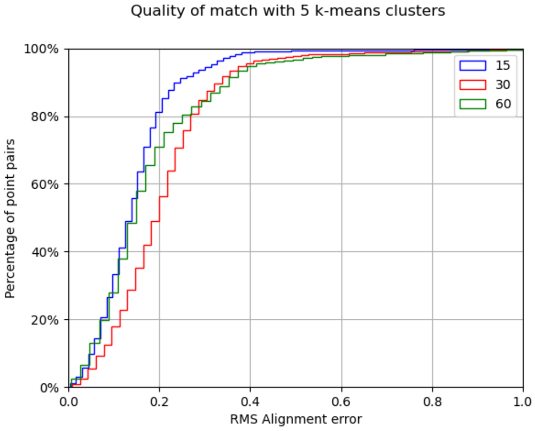


Figure 10: Match quality results for k-meansICP with $k = 5$ clusters

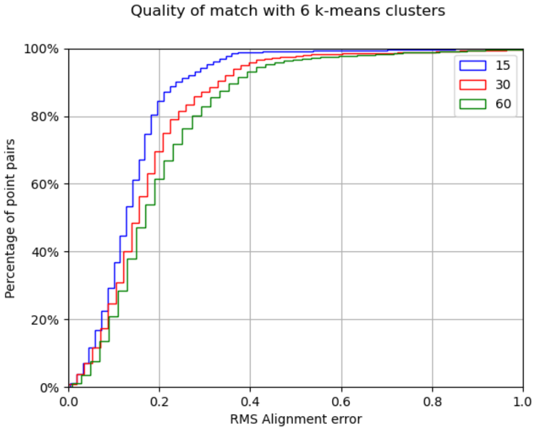


Figure 11: Match quality results for k-meansICP with $k = 6$ clusters

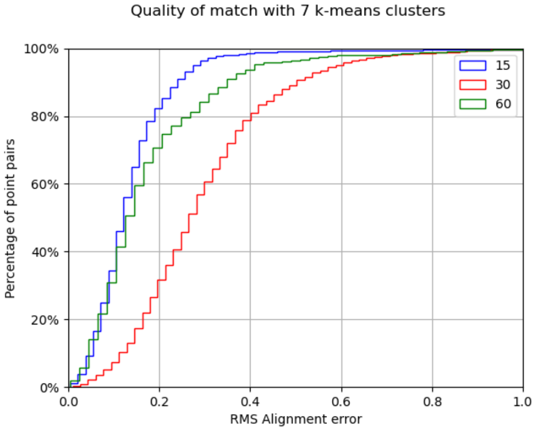


Figure 12: Match quality results for k-meansICP with $k = 7$ clusters

error for all initial rotation angles as compared to k-meansICP (Figure 16). This trend is seen across all the other match quality (Figures 7–15) and convergence graphs (Figures 16–24) for increasing number of clusters, showing lower final point pair alignment errors. This further illustrates the robust nature of the ICP algorithm for registering the same shapes with no noise. Although k-meansICP makes use of k-means clustering to reduce the search space and make the registration faster, this improvement did not correlate with improved registration results at lower convergence errors. k-means clustering [Kri99] is used to accelerate the closest point

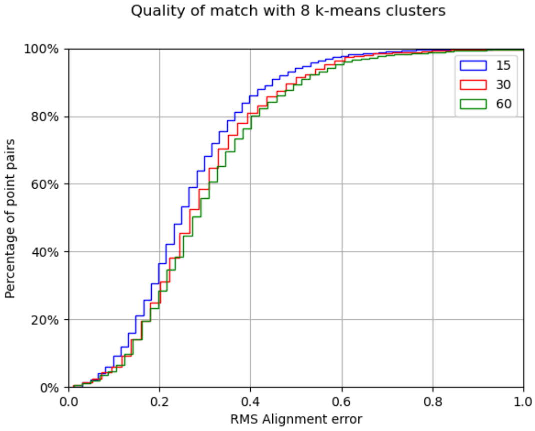


Figure 13: Match quality results for k-meansICP with $k = 8$ clusters

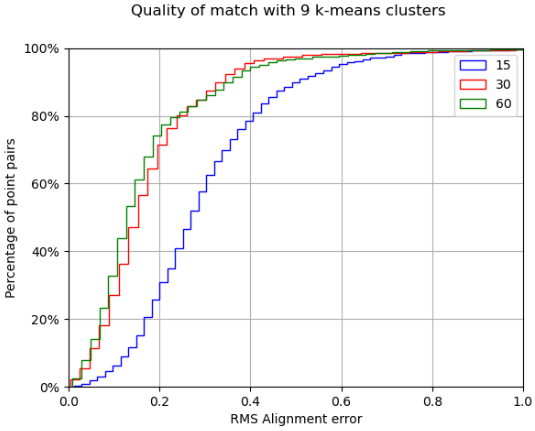


Figure 14: Match quality results for k-meansICP with $k = 9$ clusters

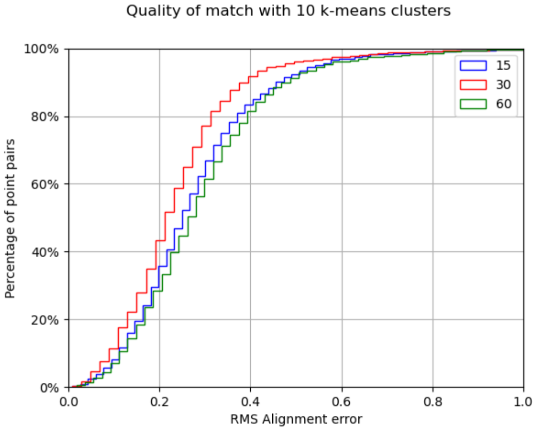


Figure 15: Match quality results for k-meansICP with $k = 10$ clusters

search but can result in less suitable matches leading to less optimal performance.

We also observed that the number of clusters had minimal impact on match quality and convergence.

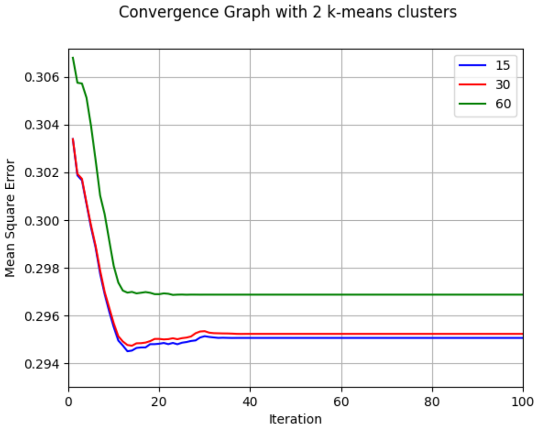


Figure 16: Convergence results for k-meansICP with $k = 2$ clusters

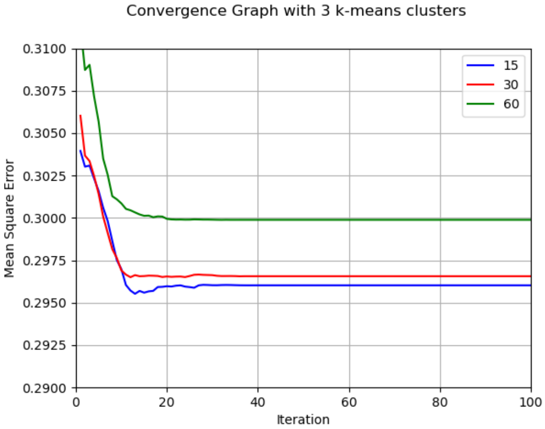


Figure 17: Convergence results for k-meansICP with $k = 3$ clusters

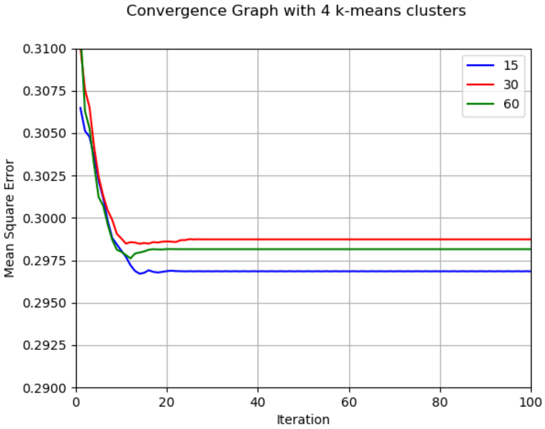


Figure 18: Convergence results for k-meansICP with $k = 4$ clusters

6 CONCLUSIONS AND FUTURE WORK

The original ICP algorithm [Bes92] is the most common algorithm used for point cloud registration, however, it has limitations such as the requirement for a good initial transformation, the computational com-

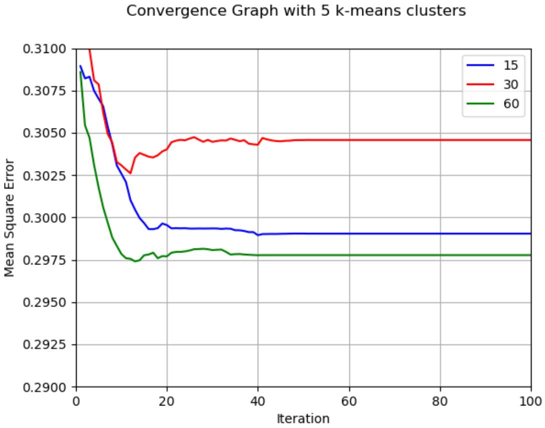


Figure 19: Convergence results for *k*-meansICP with $k = 5$ clusters

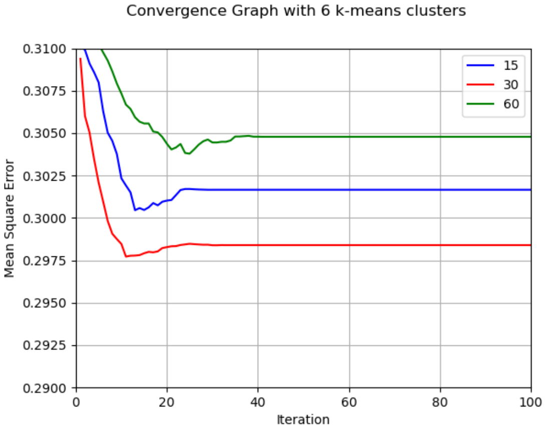


Figure 20: Convergence results for *k*-meansICP with $k = 6$ clusters

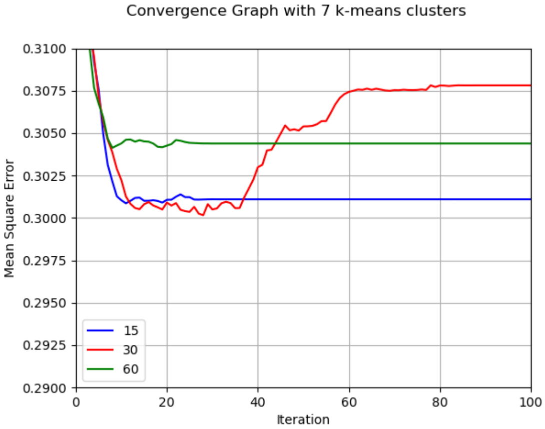


Figure 21: Convergence results for *k*-meansICP with $k = 7$ clusters

plexity of the correspondence search process, and poor performance in the presence of noise and outliers.

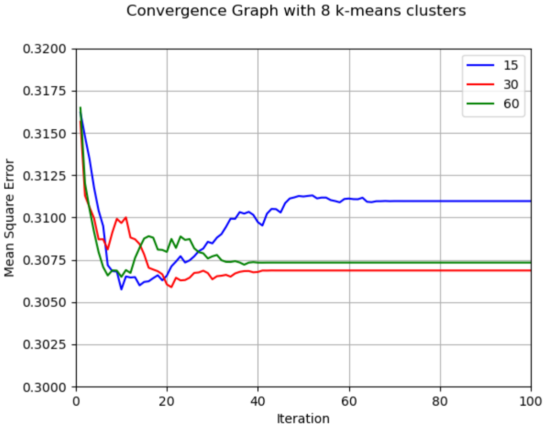


Figure 22: Convergence results for *k*-meansICP with $k = 8$ clusters

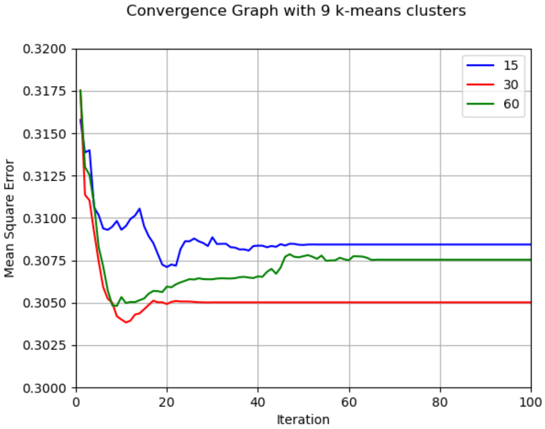


Figure 23: Convergence results for *k*-meansICP with $k = 9$ clusters

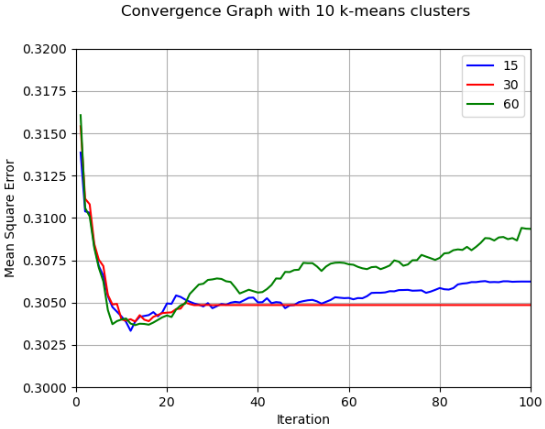


Figure 24: Convergence results for *k*-meansICP with $k = 10$ clusters

This paper presented the *k*-meansICP algorithm to improve the performance of ICP by reducing the correspondence search space. The approach partitions the data point cloud using *k*-means clustering [Kri99] resulting in a decrease in the computational complexity. This work also investigated the impact of the number of *k*-means clusters on point cloud registration. The results did not show a clear correlation between the number of *k*-means clusters and the match quality and convergence using the *k*-meansICP algorithm.

Results indicate a lower quality match and slower convergence compared to ICP. However, the impact of the number of *k*-means clusters on 3D point cloud registration can be investigated further to consider variables such as the symmetry of the point cloud, and implementation options such as using *k*-meansICP for an initial coarse alignment before a fine alignment using ICP.

7 REFERENCES

[Ank20] Ankomah, P., Vangorp, P., Behera, A., and Liu, Y. Tagged-ICP: An Iterative Closest Point Algorithm with Metadata Knowledge for Improved Matching of 3D Protein Structures. In Proc. Irish Machine Vision and Image Processing conference, 2020.

[Ber00] Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., and Bourne, P. The Protein Data Bank. Nucleic Acids Research 28, no. 1, pp. 235–242, 2000.

[Bes92] Besl, P.J. and McKay, N.D. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14, no. 2, pp. 239–256, 1992.

[Bo19] Bo, P. and Barton, M. On initialization of milling paths for 5-axis flank CNC machining of free-form surfaces with general milling tools. Computer Aided Geometric Design 71, pp. 30–42, 2019.

[Che11] Chen, J., Xiaojun, W., Wang, M., and Fuqin, D. Human Body Shape and Motion Tracking by Hierarchical Weighted ICP. In Proc. International Symposium on Visual Computing, 2011.

[Du15] Du, S., Liu, J., Zhang, C., Zhu, J., and Li, K. Probability iterative closest point algorithm for m-D point set registration with noise. Neurocomputing 157, pp. 187–198, 2015.

[Geh15] Gehrer, D. CellUnity - an Interactive Tool for Illustrative Visualization of Molecular Reactions. In Proc. Central European Seminar on Computer Graphics, 2015.

[Goj20] Gojcic, Z., Zhou, C., Wegner, J.D., Guibas, L.J., and Birdal, T. Learning multiview 3D point cloud registration. In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1759–1769, 2020.

[Hun07] Hunter, J.D. Matplotlib: A 2D graphics environment. Computing in Science & Engineering 9, no. 3, pp. 90–95, 2007.

[Iko23] Ikotun, A.M., Ezugwu, A.E., Abualigah, L., Abuhaija, B., and Heming, J. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences 622, pp. 178–210, 2023.

[Kan02] Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., and Wu, A.Y. An efficient k-means clustering algorithm: Analysis and implementation. IEEE transactions on pattern analysis and machine intelligence 24, no. 7, pp. 881–892, 2002.

[Kri99] Krishna, K. and Narasimha Murty, M. Genetic K-means algorithm. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 29, no. 3, pp. 433–439, 1999.

[Kri15] Kriegel, S., Rink, C., BodenmÃ¼ller, T., and Suppa, M. Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects. Journal of Real-Time Image Processing 10, no. 4, pp. 611–631, 2015.

[Li22] Li, J., Hu, Q., Zhang, Y., and Ai, M. Robust symmetric iterative closest point. ISPRS Journal of Photogrammetry and Remote Sensing 185, pp. 219–231, 2022.

[Lyu24] Lyu, M., Yang, J., Qi, Z., Xu, R., and Liu, J. Rigid pairwise 3D point cloud registration: A survey. Pattern Recognition 151, p. 110408, 2024.

[MA23] Monji-Azad, S., Hesser, J., and Löw, N. A review of non-rigid transformations and learning-based 3D point cloud registration methods. ISPRS Journal of Photogrammetry and Remote Sensing 196, pp. 58–72, 2023.

[Ovr23] Ovrebo, O., Ojansivu, M., Kartasalo, K., Barriga, H.M.G., Ranefall, P., Holme, M.N., and Stevens, M.M. RegiSTORM: channel registration for multi-color stochastic optical reconstruction microscopy. BMC Bioinformatics 24, no. 1, p. 237, 2023.

[Pom13] Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. Comparing ICP variants on real-world data sets: Open-source library and experimental protocol. Autonomous Robots 34, no. 3, pp. 133–148, 2013.

[Pot06] Pottmann, H., Huang, Q.X., Yang, Y.L., and Hu, S.M. Geometry and convergence analysis of algorithms for registration of 3D shapes. International Journal of Computer Vision 67, pp. 277–296, 2006.

[Rus01] Rusinkiewicz, S. and Levoy, M. Efficient variants of the ICP algorithm. In Proc. 3-D Digital Imaging and Modeling, pp. 145–152, 2001.

[Sal24] Saleh, A.R. and Momeni, H.R. An improved iterative closest point algorithm based on the particle filter and K-means clustering for fine model matching. The Visual Computer , 2024.

[Wan17] Wang, F. and Zhao, Z. A survey of iterative closest point algorithm. In 2017 Chinese Automation Congress (CAC), pp. 4395–4399. IEEE, Jinan, 2017.

[Wu18] Wu, M.L., Chien, J.C., Wu, C.T., and Lee, J.D. An Augmented Reality System Using Improved-Iterative Closest Point Algorithm for On-Patient Medical Image Visualization. Sensors 18, no. 8, p. 2505, 2018.

[Zha22] Zhang, J., Yao, Y., and Deng, B. Fast and Robust Iterative Closest Point. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, no. 7, pp. 3450–3466, 2022.

SEL-CIE: Self-Supervised Learning Framework for CIE-XYZ Reconstruction from Non-Linear sRGB Images

Shir Barzel
Tel Aviv University
sbarzel@gmail.com

Moshe Salhov
Playtika LTD
moshesa@playtika.com

Ofir Lindenbaum
Bar Ilan University
ofirlin@gmail.com

Amir Averbuch
Tel Aviv University
amir1@tauex.tau.ac.il

ABSTRACT

Modern cameras typically offer two types of image states: a minimally processed linear raw RGB image representing the raw sensor data, and a highly-processed non-linear image state, such as the sRGB state. The CIE-XYZ color space is a device-independent linear space used as part of the camera pipeline and can be helpful for computer vision tasks, such as image deblurring, dehazing, and color recognition tasks in medical applications, where color accuracy is important. However, images are usually saved in non-linear states, and achieving CIE-XYZ color images using conventional methods is not always possible. To tackle this issue, classical methodologies have been developed that focus on reversing the acquisition pipeline. More recently, supervised learning has been employed, using paired CIE-XYZ and sRGB representations of identical images. However, obtaining a large-scale dataset of CIE-XYZ and sRGB pairs can be challenging. To overcome this limitation and mitigate the reliance on large amounts of paired data, self-supervised learning (SSL) can be utilized as a substitute for relying solely on paired data. This paper proposes a framework for using SSL methods alongside paired data to reconstruct CIE-XYZ images and re-render sRGB images, outperforming existing approaches. The proposed framework is applied to the sRGB2XYZ dataset.

Keywords

CIE-XYZ Color Space, sRGB, Image Reconstruction, Self-Supervised Learning (SSL), Raw Image, Macbeth ColorChecker

1 INTRODUCTION

In the realm of digital photography, a customary process involves the transformation of a sensor RAW image captured by a digital camera into the standardized sRGB format utilizing an in-camera Image Signal Processor (ISP) [16]. Traditional ISPs are optimized primarily to generate visually appealing, compressed RGB images that cater to human perception. The pervasive availability of such RGB images on the internet has contributed significantly to the recent advancements in machine learning-based computer vision technologies. The image processor in a digital camera applies various adjustments to the captured sensor image [23]. In the initial stage, linear operations like white balance and color adaptation transform the sensor-specific raw RGB image into a standardized color space, such as CIE-XYZ [24]. This creates a scene-referred image that directly correlates with the original captured scene. Subsequently, the "photo-finishing" stage involves applying non-linear adjustments and local operators to enhance the visual aesthetics of the photograph. This may include selectively manipulating colors to improve skin tones or increasing local contrast for a more striking appearance. Finally, the processed image is converted to the desired output color space.

The increasing prevalence of digital imaging has propelled the development of modern cameras that provide users access to either one of the two distinct image states: minimally processed linear raw RGB data and highly processed non-linear images, such as those in the sRGB state. These two image states serve different purposes, with the former representing the raw sensor data and the latter addressing the visual perception of users. With its linear relationship to scene radiance, the raw-RGB image state offers advantages for low-level computer vision tasks such as deblurring, dehazing, denoising, and image enhancement [25, 13, 26]. However, the sensor-specific nature of color filter arrays in the raw RGB format leads to significant variations in captured values between different sensors, often necessitating sensor or camera-specific tailor-made algorithms. The display-referred image state, typically in the sRGB color space, is widely used for display purposes but can vary significantly in value due to proprietary photo-finishing applied by different cameras. This leads to differences in sRGB values between images captured of the same scene using different camera models or settings.

Using a device-independent linear color space in computer vision applications, such as CIE-XYZ, has proven valuable for multiple tasks [4, 13]. The reconstruction of color images in the CIE-XYZ color space from non-linear images is crucial for achieving accurate color representations. However, conventional methods that rely

https://www.doi.org/10.24132/CSRN.3401.3

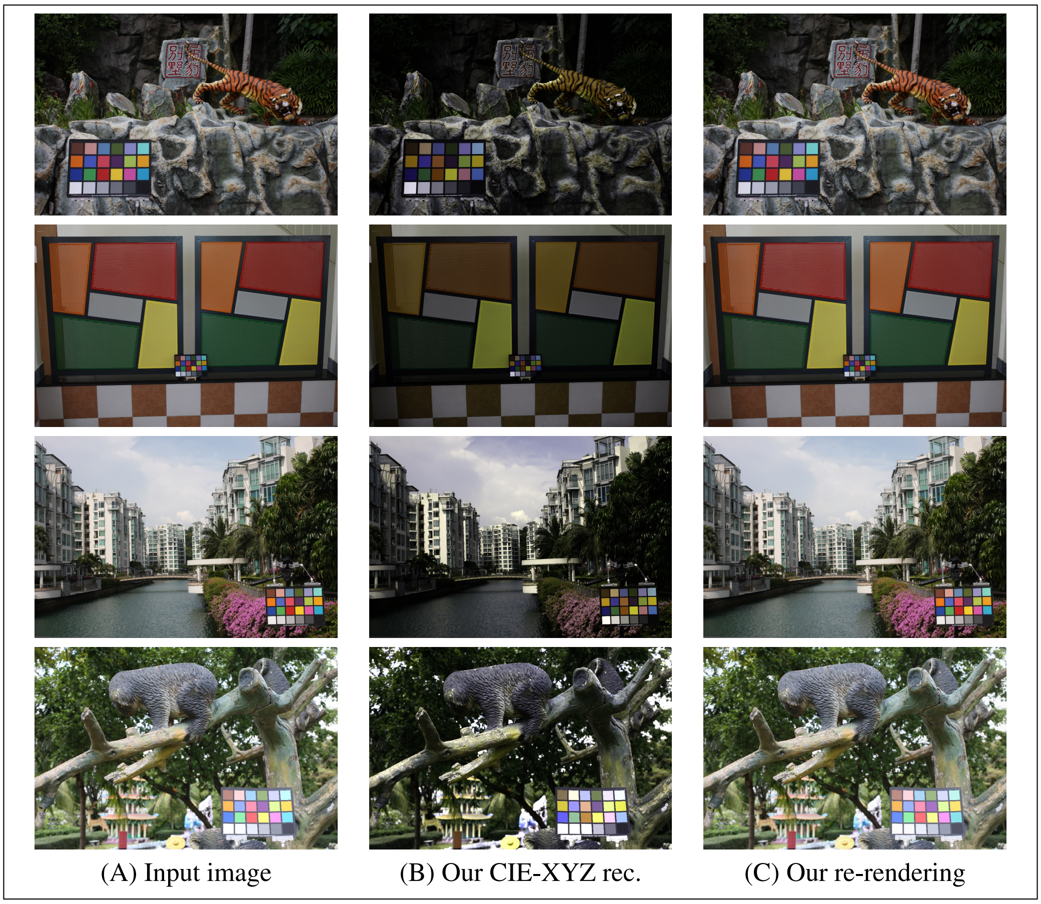

(A) Input image (B) Our CIE-XYZ rec. (C) Our re-rendering

Figure 1: Visual comparisons for CIE-XYZ reconstruction and re-rendering. (A) The input sRGB image. (B) CIE-XYZ reconstruction using the proposed method. (C) Our re-rendered output was generated from the reconstructed CIE XYZ image. CIE XYZ images are scaled by a factor of two to aid visualization. The input images are sourced from the NUS dataset [1].

solely on paired CIE-XYZ and sRGB representations face challenges due to the limited availability of large-scale paired image datasets. Acquiring such datasets is often time-consuming, expensive, and not easily scalable, hindering the development of robust color image reconstruction techniques. To address the challenges of acquiring labeled datasets at scale (like CIE-XYZ and sRGB pairs), SSL has recently gained significant attention as an alternative paradigm [27, 37, 38, 39]. Unlike traditional supervised learning, which relies on externally provided labels, SSL leverages the intrinsic properties of the input data to generate surrogate labels. By exploiting the inherent information in the data, SSL offers a promising avenue to overcome the reliance on paired data and enhance the performance of color image reconstruction.

Most SSL approaches in computer vision focus on image segmentation and classification. Using such SSL techniques has substantially improved performance outcomes in various domains, including medical image analysis and object recognition. For example, SimCLR [14] proposed a simple yet effective framework for SSL of visual representations. The SimCLR method used a contrastive learning approach to learn representations that capture the similarity between different views of the same image. SimCLR demonstrated state-of-the-art performance on several benchmark image classification datasets, including CIFAR-10, CIFAR-100 [33], and ImageNet [10]. Utilizing a pre-trained model followed by fine-tuning offers the advantage of requiring less data than training from scratch. However, in certain applications, a suitable pre-trained model may not be available. In such cases, an alternative approach that is both practical and yields comparable results while avoiding the need for large amounts of annotated data required by pre-trained models becomes desirable. This is where SSL comes into play. SSL operates without needing external labels, instead leveraging inherent information in the input data. In this study, we aim to adapt the SSL method to reconstruct CIE-XYZ from non-linear RGB images.

This paper proposes an SSL-based method for reconstructing CIE-XYZ images from non-linear RGB inputs. Our framework aims to mitigate the challenges associated with limited paired data availability. We draw inspiration from successful SSL techniques developed for conventional computer vision tasks. These techniques have been applied in image segmentation and classification tasks, as demonstrated in foundational works such as [34]. The SimCLR [14] framework, in particular, demonstrated remarkable performance in learning visual representations by capturing the similarity between different views of the same image."

Our methodology employs SSL and includes a pretask that focuses on the color boards present in the images. We were inspired by the work of [7], who introduced a domain knowledge-guided SSL technique for change detection in remote sensing images, and we adopted a similar approach. The authors in [7] utilized prior knowledge of remote sensing indices to direct the learning process and improve change detection capabilities. Similarly, we use prior knowledge of color boards within the images to guide the learning process and enhance the quality of reconstruction.

By leveraging the predetermined colors of patches on the color boards, we develop a self-supervised training paradigm that enables the reconstruction of color patches in the CIE-XYZ color space. This concept can be seen in Fig. 1, where we present visual comparisons showcasing CIE-XYZ reconstruction and re-rendering processes. Incorporating color boards provides an inherent label source for the network during the self-supervised training, eliminating the need for paired CIE-XYZ and sRGB images. To evaluate the effectiveness of our proposed framework, we employ the sRGB2XYZ dataset created by [4]. This dataset, derived from the MIT-Adobe FiveK dataset, offers pairs of sRGB and camera CIE-XYZ images obtained through a camera pipeline. Additionally, we compare our framework's performance against state-of-the-art methods to showcase its color accuracy and reconstruction quality superiority.

In summary, our contributions are:

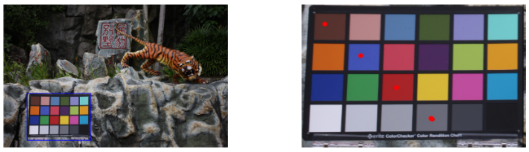
- We present a new framework that leverages SSL to advance the reconstruction of color images in the CIE-XYZ color space from non-linear RGB inputs. By mitigating the reliance on paired data and drawing inspiration from SSL techniques, our algorithm offers an innovative approach to enhance color image reconstruction.
- We benchmark our method and demonstrate its superiority over existing approaches, highlighting its potential impact on various computer vision applications requiring precise color representations.
- We further demonstrate that a pre-trained classification network (such as ResNet) can be used to improve performance in CIE-XYZ reconstruction. We, therefore, incorporate such a backbone into our model.

2 RELATED WORK

Methodologies for de-rendering sRGB images can be classified into two categories: those that incorporate specialized metadata during the capture process and blind methods that do not rely on additional information. Early digital cameras lacked access to the sensor's raw-RGB image, leading to radiometric calibration methods [17, 18, 19] focusing on linearizing the sRGB data rather than accurately recovering raw-RGB values. These methods employ simplistic models, such as a primary 1D response function per color channel, to establish a linear relationship between the digital values and scene radiance. As for the blind methods within the other category, they too can be further divided into two types. The initial type consists of methods attempting to model the parametric relationship that maps from sRGB to some linear state [20, 13]. The second type comprises machine learning methods that aim to learn the color transformation using pairs of sRGB images and their corresponding images used for reconstruction; this includes those in the CIE-XYZ color space [4] and raw-RGB format [22].

In addition to the de-rendering methods that adopt a generic approach, such as those presented in [13, 28], which often exhibit limited accuracy due to their inability to model camera-specific operations, there are other works that employ different neural network solutions. For instance, [35] is a relevant example. These methods assume a standard set of Image Signal Processor (ISP) operations, which hinders their ability to account for individual cameras' unique characteristics. In contrast, our combination of unsupervised and self-supervised training enables broader generalization, enhancing reconstruction accuracy. Moreover, while machine learning-based image reconstruction methods can take advantage of the abundance of images available on the internet for training, the absence of the desired image pairs required for supervised training significantly limits their effectiveness and generalization. Certain studies have incorporated prior knowledge into the learning process in scenarios where the availability of labeled data is limited. Integrating domain-specific information in the training process enhances the reliability of the resulting model [31].

In recent years, there has been an increasing interest in incorporating prior knowledge into the training process of machine learning models. As a result, data-driven and knowledge-guided methods have emerged, showing promising improvements in model performance. An innovative approach, proposed by [29], involves supervising neural networks by defining constraints in the



(a) (b)

Figure 2: The methodology employed for executing self-supervised training using the color boards allows the creation of a pre-task. This pre-task involves retaining and matching the colors of patches located on the color boards with corresponding colors in the CIE-XYZ color space. Sub-figure (a) displays an example image from the NUS dataset [1], where the color board is detected based on the metadata. Additionally, (b) illustrates a visualization of the extra information in the metadata that can be used to sample each color board's color patches during the pre-task.

output space rather than relying on explicit input-output pairs as training data. These constraints are derived from prior domain knowledge, such as known laws of physics. The authors demonstrate the efficacy of this method on various real-world and simulated computer vision tasks, highlighting its potential to improve the performance of neural networks in practical applications.

Compared to existing approaches in color image reconstruction, our framework offers a distinctive aspect by utilizing guided SSL with color boards, providing an innovative and practical way to enhance CIE-XYZ image reconstruction without the need for extensive paired data. Through evaluations on the sRGB2XYZ dataset [4], our proposed framework outperforms existing methods, showcasing its potential impact on various computer vision applications requiring precise color representations. Our work contributes to advancing the field of color image reconstruction and demonstrates the viability of SSL in this context.

3 PROPOSED METHOD - MULTI-PHASE TRAINING FRAMEWORK FOR CIE-XYZ IMAGE RECONSTRUCTION

The proposed approach follows a multi-phase training scheme for CIE-XYZ image reconstruction, comprising three key training phases with weight adaptation and transfer (see Fig.3). First, we use supervised training with pairs of sRGB images and their corresponding CIE-XYZ images to minimize discrepancies between the reconstructed and original images. Then, we perform a self-supervised step based on datasets with sRGB images containing color boards and predetermined colors for color patch regions, enriching the network's learning process. Finally, we perform additional supervised training on a dataset containing sRGB and CIE-XYZ image pairs, refining learned representations and incorporating supervised data for improved performance.

The supervised training phase requires a dataset comprising sRGB images paired with corresponding linear images in the CIE-XYZ color space. To achieve this, we use the sRGB2XYZ dataset [4] derived from the MIT-Adobe FiveK dataset [5]. Creating this dataset involves taking raw-RGB images from the MIT-Adobe FiveK dataset and processing them twice, resulting in both sRGB and CIE-XYZ versions of each image. The authors used the camera pipeline outlined in [6] to convert raw-RGB images into the CIE-XYZ color space. This pipeline allowed them to access the CIE-XYZ values by processing the sensor raw-RGB images. The process includes using the color space transformation (CST) matrices provided with the raw-RGB images. The dataset includes 1200 pairs of sRGB and camera CIE-XYZ images. The second type of dataset necessary for the self-supervised part must include sRGB images that contain a color board within the image. Here, we used the dataset presented by [1]. The dataset comprises images from 9 commercial cameras, where over 200 images were captured for each camera. The images were taken in natural settings, both indoor and outdoor, with a color board presented within the image. A text file containing the coordinates of the color boards and their corresponding color patches is provided for each image. Combining these two types of datasets allows the network to learn from both supervised and self-supervised data, contributing to a more robust and comprehensive training reconstruction process.

3.1 Phase I - Training With Paired Images

In the first phase, the network undergoes supervised training using the sRGB2XYZ dataset [4]. This dataset contains pairs of sRGB images and their corresponding linear images in the CIE-XYZ color space. The goal is to reconstruct CIE-XYZ images that accurately represent the color information in the input sRGB images. During training, the network's weights are adjusted to minimize the error between the reconstructed CIE-XYZ images and their corresponding ground truth CIE-XYZ images from the dataset. The supervised training phase is crucial for establishing an anchor for the main task of CIE-XYZ image reconstruction. By learning from the paired sRGB and CIE-XYZ images, the network learns the color relationships and representations required to transform sRGB data into the CIE-XYZ color space accurately.

The loss function (Eq. 1) used in the supervised training phase:

$$L_s = \lambda\,|\hat{x}_{xyz} - x^*_{xyz}| + |\hat{x}_{srgb} - x^*_{srgb}|, \qquad (1)$$

is derived from [4], which aims to minimize the mean absolute error (MAE) between the predicted CIE-XYZ

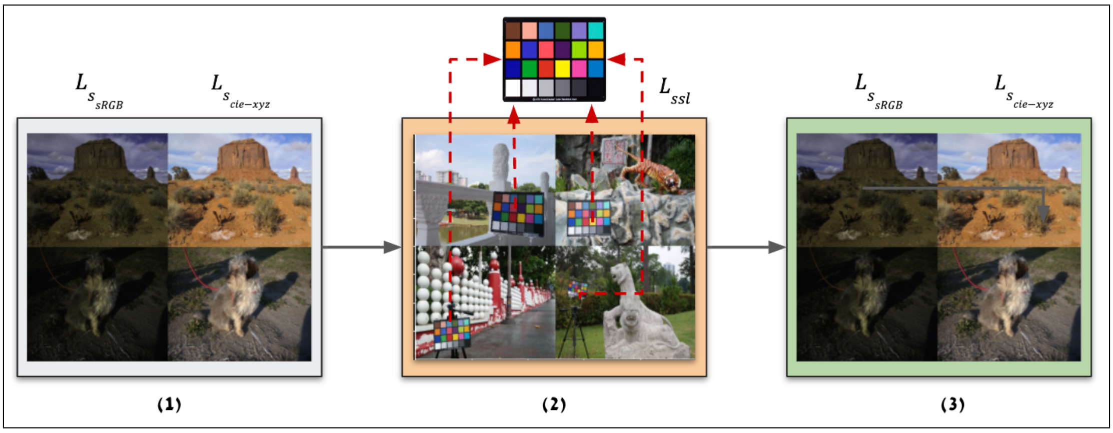


Figure 3: The training framework consists of three main phases: (1) Supervised training using the sRGB2XYZ dataset [4] to reconstruct CIE-XYZ images with the loss $L_{s_{srgb}}$ for minimizing sRGB image reconstruction error and $L_{s_{cie-xyz}}$ for minimizing CIE-XYZ image reconstruction error, (2) SSL using the NUS dataset [1] and color board patches with known CIE-XYZ colors to augment the network, with the loss L_{ssl} for minimizing Delta E 76 discrepancy between reconstructed color board patches and their corresponding ground truth CIE-XYZ values, and (3) An additional phase of supervised training on the sRGB2XYZ dataset for further enhancement and incorporation of supervised data with the same loss functions $L_{s_{srgb}}$ and $L_{s_{cie-xyz}}$.

image $\hat{x}_{xyz}$ and its corresponding ground truth x^*_{xyz}, as well as the predicted sRGB image $\hat{x}_{srgb}$ and its corresponding ground truth x^*_{srgb}. The loss enhances color image reconstruction by encouraging the model to capture the meaningful color relationships between the images and produce accurate representations. Additionally, the choice of L1 loss (MAE) over L2 (MSE) is preferred in this context because it can handle outliers and produce more visually pleasing results for color representations. The value of λ is a weighting factor calibrated by [4], and we have adopted their value of 1.5 for our implementation.

3.2 Phase II - Refinement With Color Boards

In the second phase, the network undergoes SSL using the NUS dataset [1]. The dataset contains sRGB images that have color boards positioned within the image. Each color board contains color patches with known colors. The network is now tasked with leveraging this known information to enhance its understanding of the CIE-XYZ color space transformation. We propose utilizing images containing color boards positioned within them. By doing so, we draw inspiration from the inherent knowledge present in the input data, explicitly referring to the known colors of the color patches on the color boards. These predetermined colors act as natural labels that guide the training process, eliminating the need for paired CIE-XYZ and sRGB images and enabling the network to learn the mapping between sRGB and CIE-XYZ color spaces more effectively. This idea is somewhat analogous to [2], who conducted a preliminary task involving the prediction of image rotations. Here, we use the fact that the colors of the patches on the color boards are predetermined and already known (an example can be seen in Fig.2).

By utilizing this fact, we established a pre-task wherein the color patches on the color boards were required to correspond to the relevant colors within the CIE-XYZ color space after the CIE-XYZ image reconstruction.

Considering the inherent constraint within the architecture of the proposed neural network, wherein local processing on the image is eliminated, it becomes evident that the network structure requires a transformation, which in this context refers to a matrix multiplication applied to the entire image. Consequently, if a transformation applies to the color board, it applies to all pixels in the image. The SSL phase is designed to augment the network with additional information using a dataset that may not necessarily comprise image pairs. Instead of relying on external annotations or ground truth labels, the color patches on the color boards act as inherent labels. The network's weights are adjusted to minimize the discrepancy between the actual color values of the color board patches in the CIE-XYZ color space and the color patches reconstructed by the network. By pre-training on this self-supervised task, the network can learn to associate the color board patches with their corresponding CIE-XYZ colors, thereby gaining valuable knowledge about the color space and improving its ability to reconstruct accurate CIE-XYZ images in the subsequent phases.

The loss function employed in the self-supervised training phase is based on the Delta E 76 formula [21]. The primary objective of this loss function is to minimize the Delta E 76 between the reconstructed colors of the color board patches and their corresponding ground truth CIE-XYZ color values. The rationale behind this loss function is to identify a pre-task that does not necessitate pairs of CIE-XYZ and sRGB images, thus allowing for the generalization and enrichment of the training data. The patch color is determined by sampling the colors inside the patch and representing them as a matrix C_s. The matrix C_s has a shape of $n \times 3$, where n is the number of pixels within the mask corresponding to the patch, and each row represents the color value of a pixel in the CIE-XYZ color space. To obtain a single representative color for the patch, the 75th percentile value of C_s is taken as $\mathscr{F}_{q_{75}}$ and used to obtain C_{sq}, defined as:

$$C_{sq} = \mathscr{F}_{q_{75}}(C_{sq}), \tag{2}$$

which is the reconstructed patch color in the CIE-XYZ color space. To utilize the Delta E method, colors must be converted to the CIELAB color space [32]. In this study, the reconstructed patch color C_{sq} is transformed into C_{rpc} using the conversion method outlined in [8]. The conversion function is denoted as $\mathscr{F}_{xyz->lab}$ and is used to convert from CIE-XYZ to CIELAB, as shown below:

$$C_{rpc} = \mathscr{F}_{xyz->lab}(C_{sq}). \tag{3}$$

The Delta E method computes the color difference between the reconstructed patch color and the ground truth color of the i-th patch in the LAB color space C_{gtpc}, which is provided as a property of the color checker, the color checker is a color board that complies with international standards and is widely used in camera calibration and color correction [36]. The Delta E for the i-th patch is calculated using the following equation:

$$\begin{aligned} \Delta E_i = {} & \sqrt{(C^L_{rpc} - C^L_{gtpc})^2} \\ & + \sqrt{(C^a_{rpc} - C^a_{gtpc})^2} + \sqrt{(C^b_{rpc} - C^b_{gtpc})^2}. \end{aligned} \tag{4}$$

Finally, the self-supervised loss function part is denoted as:

$$L_{ssl} = \mathscr{F}_m(\{\Delta E\}_{i=0}^{i=n}), \tag{5}$$

where $\mathscr{F}_m$ is the mean operator, and n is the number of color patches in the color board.

During the self-supervised training, the loss function combines two components - the self-supervised loss described in Equation 5 and the sRGB part of the supervised loss.

$$L_{sslt} = \delta|\hat{x}_{srgb} - x^*_{srgb}| + L_{ssl}. \tag{6}$$

Combining the above two terms balances the two aspects of the network's learning process. While the self-supervised loss facilitates the network in acquiring valuable insights into the transformation between the sRGB and CIE-XYZ color spaces, the sRGB part of the supervised loss ensures that the network accurately reconstructs the original sRGB image.

To balance the self-supervised learning and the supervised reconstruction tasks, we introduced a trainable parameter, denoted as δ, into our loss function. This addition empowers us to dynamically calibrate the interplay between self-supervised and supervised losses, thus enabling the network to progressively refine its focus on these dual objectives during the learning process. The δ parameter, a singular scalar, effectively controls the relative significance of the two types of loss elements. By incorporating δ as a trainable parameter, the network can find the best balance between grasping color space insights and achieving accurate sRGB image reconstruction.

3.3 Phase III - Final Supervised Refinement

In the final phase, the network undergoes another round of supervised training using the sRGB2XYZ dataset [4]. Similar to Phase I, this phase involves training the network on pairs of sRGB images and their corresponding linear images in the CIE-XYZ color space. The purpose of this final supervised training phase is twofold. Firstly, it aims to enhance the information learned in the previous self-supervised phase by fine-tuning the representation learned by the network. Secondly, it addresses any errors or biases that might have occurred during the self-supervised training process.

By using a multi-phase training approach, the network can benefit from both supervised and semi-supervised learning. The supervised training provides a solid foundation for the main task of CIE-XYZ image reconstruction, while the SSL with color boards augments the network's understanding of the CIE-XYZ color space. Combining these phases enables the network to produce more accurate and reliable CIE-XYZ image reconstructions.

4 EXPERIMENTAL RESULTS

Our approach is evaluated on the benchmark proposed by [4], utilizing the test set of the sRGB2XYZ dataset. The benchmark serves to validate the efficacy of the proposed framework in the mapping of camera-generated sRGB images to CIE-XYZ and the processing of CIE-XYZ images back to sRGB.

4.1 Implementation Details

Our neural network architecture is based on the network described in [4], which aims to emulate the camera imaging pipeline. The architecture comprises two sub-networks that model global and local processing parts. This architecture is employed across all three training phases, with the weights transferred from one phase to another. Pre-training is a prevalent approach in computer vision, wherein the backbone of object detection and segmentation models is often initialized using supervised ImageNet pre-training. Our research explored another innovation, which involves using a pre-trained backbone before the local processing CNN, as depicted in Fig.4. Specifically, we utilized a pre-trained ResNet50 based on the architecture proposed in [9] that was trained on the ImageNet dataset [10]. This choice of using ResNet50 as the pre-trained backbone was motivated by its proven effectiveness in a wide range of computer vision tasks, demonstrated by its outstanding performance in various benchmarks. Moreover, ResNet50's depth and skip connections facilitate feature extraction at multiple scales, which is particularly beneficial for tasks like image reconstruction.

The split of training phases into supervised and self-supervised components is based on the availability and characteristics of the datasets and provides several benefits. The first and third components, which are supervised, rely on the sRGB2XYZ dataset [4], consisting of 971 pairs of sRGB images and their corresponding linear CIE-XYZ images. This dataset was suitable for the supervised training as it provided ground truth pairs, enabling the network to learn the color transformation accurately. On the other hand, the self-supervised part of the training utilized a dataset introduced by [1] containing sRGB images with color boards. We integrated this dataset into our training process, enabling the network to leverage inherent labels from the color board patches and learn more about the CIE-XYZ color space transformation without relying solely on paired data. The split was chosen to ensure that both supervised and self-supervised learning components were optimally trained with appropriate datasets for their respective tasks.

During the supervised training, the network was trained with randomly extracted patches of size 256×256 from the training set, with a mini-batch size of 4. Additionally, scaling and reflection augmentations were employed on the extracted patches to enhance the training process. The stages of the framework were trained for 300 epochs each, employing the Adam optimizer [15] with $\beta_1 = 0.9$ as the gradient decay factor and $\beta_2 = 0.999$ as the squared gradient decay factor. A learning rate of 10^{-4} was utilized, with a decay factor of 0.5 every 75 epochs to enable convergence to a lower minimum. Additionally, to prevent overfitting, we incorporated an $L2$ regularization into our loss function in Eq.1 with a regularization weight of $\lambda_{reg} = 10^{-3}$. The choice of the parameters aligns with the parameters used in [4], which have been demonstrated to be effective for similar tasks.

The proposed method imposes an inherent constraint on the neural network's architecture, specifically requiring global processing across the entire image instead of local operations. This constraint ensures that the network can perform transformations encompassing the whole image, allowing it to be compatible with the proposed framework for CIE-XYZ image reconstruction. Moreover, this global processing capability is essential for successfully utilizing the color boards-based pre-train task.

4.2 Evaluation Metrics

Our framework's capability to "unprocess" sRGB images to CIE-XYZ and reconstruct them from CIE-XYZ back to sRGB is verified and demonstrated. To evaluate our framework's mapping to sRGB, we conduct experiments using our reconstructed CIE-XYZ results and ground-truth CIE-XYZ images as the starting points. For evaluation purposes, we compare our approach with the supervised training method in [4] and the standard CIE-XYZ mapping in [11] and [12], which uses a simple 2.2 gamma tone curve. Additionally, we compare our results with the unprocessing technique (UPI) in [13], which provides a proxy for the major procedures of the camera pipeline. We compare our results with UPI obtained at the CIE-XYZ stage to ensure a fair evaluation.

Following the proposed benchmark in [4], Table 1 shows peak-signal-to-noise ratio (PSNR) results averaged over the 244 unseen testing images from the sRGB2XYZ dataset. The terms Q1, Q2, and Q3 represent the first (lowest), second (median), and third quartile, correspondingly, of the PSNR (Peak Signal-to-Noise Ratio) values achieved by each approach.

Table 1 illustrates that our proposed method (SEL-CIE) has yielded superior results across all evaluated metrics. Furthermore, incorporating a pre-trained ResNet50 backbone (SEL-CIE-RB) has further improved the performance in the case of sRGB to CIE-XYZ transformation.

In our comprehensive evaluation, we employed the Structural Similarity Index (SSIM) [30] as a robust metric to gauge the degree of similarity between the reconstructed CIE-XYZ images and their corresponding ground truth images extracted from the sRGB2XYZ dataset [4]. SSIM takes into account various image attributes such as luminance, contrast, and structural features, providing a holistic assessment of image similarity. A higher SSIM index signifies a more significant resemblance between the images.

Table 2, situated below, provides a summarized view of the average SSIM values computed across the testing images for each of the different models used in our evaluation. It serves as a valuable reference point for understanding the performance of these models in preserving image fidelity.

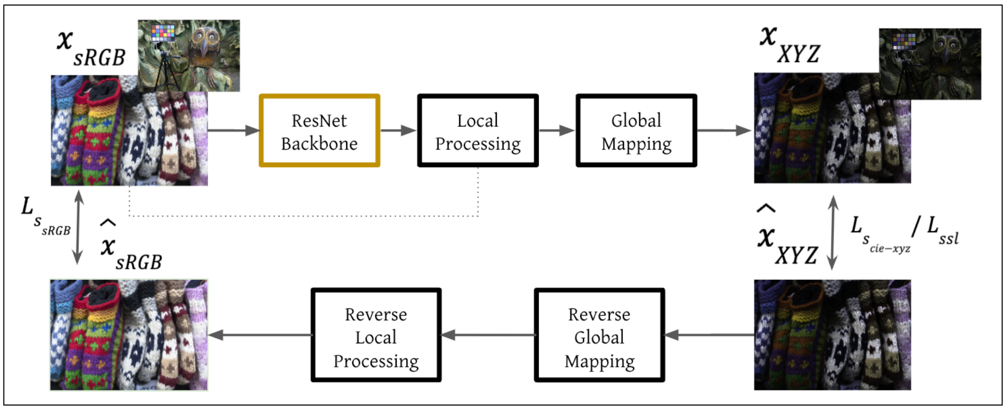


Figure 4: The CIE-XYZ image pipeline network from [4] with the ResNet pre-trained ImageNet backbone. The neural network architecture used in this study aims to emulate the camera imaging pipeline. It comprises two sub-networks that model both the global and local processing parts. The network's backbone, responsible for feature extraction and representation learning, is initialized using a pre-trained ResNet50 based on the architecture proposed in [9]. This ResNet50 backbone was initially trained on the ImageNet dataset [10], a large-scale dataset with various annotated images. The pre-trained backbone is integrated into the network before the local processing convolutional neural network (CNN), allowing it to benefit from the learned features during pre-training. The pre-trained weights of the backbone are then fine-tuned during subsequent training phases, where the network learns to perform the specific task at hand, resulting in an efficient and effective image-processing pipeline.

Method	sRGB → XYZ				Rec. XYZ → sRGB				GT XYZ → sRGB			
	Avg.	Q1	Q2	Q3	Avg.	Q1	Q2	Q3	Avg.	Q1	Q2	Q3
Standard [11, 12]	21.84	16.88	20.91	25.24	-	-	-	-	22.22	19.19	21.79	24.37
Unprocessing [13]	22.19	19.31	22.12	24.75	37.72	37.78	40.56	41.88	18.04	15.67	17.79	20.02
Afifi et al. [4]	29.66	23.77	29.57	34.71	43.82	41.43	43.94	46.58	27.44	23.57	28.32	30.88
SEL-CIE	30.38	24.51	30.46	35.16	46.43	42.49	46.04	50.54	27.87	23.86	28.8	31.49
SEL-CIE-RB	32.11	27.49	32.02	36.49	44.51	41.64	44.72	47.79	27.94	24.11	29.04	31.55

Table 1: PSNR comparison across various methods: first, sRGB to CIE-XYZ reconstruction using ground truth CIE-XYZ; second, sRGB image reconstruction from reconstructed CIE-XYZ with ground truth being the original sRGB image; and finally, PSNR comparison between the network's reconstruction (input: ground truth CIE-XYZ, output: reconstructed sRGB) and the corresponding ground truth sRGB image. The proposed SEL-CIE method surpasses existing methods in all metrics. Additionally, integrating a pre-trained ResNet50 backbone further improves performance (SEL-CIE-RB).

Method	Average SSIM
SEL-CIE-RB	0.9408
SEL-CIE	0.9363
Afifi et al. [4]	0.9338

Table 2: Comparison of Average Structural Similarity Index (SSIM) Results for Image Reconstruction Methods

These results highlight the superiority of the SEL-CIE-RB model in preserving both structural and perceptual similarity compared to previous models.

5 CONCLUSION AND FUTURE WORK

This paper has presented a framework that harnesses the power of self-supervised learning (SSL) to enhance the reconstruction capabilities of CIE-XYZ images from their corresponding non-linear sRGB images. By reducing our dependency on paired data and leveraging insights derived from SSL techniques, our framework showcased its ability to reduce errors in CIE-XYZ image reconstruction and sRGB image re-rendering. This accomplishment holds potential for various computer vision applications that demand the use of linear color representations. Furthermore, our approach has excelled compared to existing methods, emphasizing its capability to advance the field of image processing. This suggests that our framework could be valuable in diverse domains, from medical imaging to color-sensitive computer vision tasks. In addition to the SSL techniques, we have integrated a pre-trained ResNet50 backbone into our framework, resulting in an even more refined transformation process from sRGB to CIE-XYZ. This integration underscores the versatility of our approach and its potential to enhance color image reconstruction further.

Future research can explore ways to enhance our framework's generalization capabilities. While it has shown promise, there may be scenarios and datasets where its performance could be further improved. Investigating techniques to adapt the SSL model to varying imaging conditions and scene complexities is important. Additionally, collecting a dataset with color boards from a diverse range of camera types, including smartphone cameras, can contribute significantly to the model's generalization. This broader dataset can help the model adapt to different camera manipulations and sensor characteristics, further improving its robustness. As an example of exploring ways to enhance generalization capabilities, one potential scenario for future research is collaborating with companies in the medical field that develop camera-based products aimed at achieving color normalization and standardization. This approach could be an exciting exploration avenue to train our framework with their data, which may include images with color boards, enabling the algorithm to work independently on different cameras, regardless of the specific processing performed in each camera. Moreover, there exists an opportunity to explore the practical implementation of our suggested framework within the field of medical imaging, particularly in contexts necessitating color standardization and normalization across different cameras. Medical scenarios often demand exceptional precision, particularly when dealing with color accuracy. Tasks such as analyzing wound tissues or interpreting color variations in diagnostic tests critically hinge upon obtaining precise color information. Adapting the usage of our framework to address these specific requirements in medical imaging could yield substantial benefits. This research direction can pave the way for improved diagnostic accuracy and heightened reliability in medical assessments by addressing the unique challenges related to standardization and normalization within the medical domain.

6 REFERENCES

[1] Cheng, Dongliang; Prasad, Dilip K; Brown, Michael S. *Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. JOSA A*, 31(5), 1049–1058, 2014. Publisher: Optica Publishing Group.

[2] Gidaris, Spyros; Singh, Praveer; Komodakis, Nikos. *Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728*, 2018.

[3] McCamy, Calvin S; Marcus, Harold; Davidson, James G; et al. *A color-rendition chart. J. App. Photog. Eng*, 2(3), 95–99, 1976.

[4] Afifi, Mahmoud; Abdelhamed, Abdelrahman; Abuolaim, Abdullah; Punnappurath, Abhijith; Brown, Michael S. *Cie xyz net: Unprocessing images for low-level computer vision tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9), 4688–4700, 2021. Publisher: IEEE.

[5] Bychkovsky, Vladimir; Paris, Sylvain; Chan, Eric; Durand, Frédo. *Learning photographic global tonal adjustment with a database of input/output image pairs*. In: *CVPR 2011*, 97–104, 2011. Organization: IEEE.

[6] Abdelhamed, Abdelrahman; Lin, Stephen; Brown, Michael S. *A high-quality denoising dataset for smartphone cameras*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1692–1700, 2018.

[7] Yan, Li; Yang, Jianbing; Wang, Jian. *Domain Knowledge-Guided Self-Supervised Change Detection for Remote Sensing Images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2023. Publisher: IEEE.

[8] Pub, CIE. *Technical Report 15: 2004: Colorimetry. Vienna: CIE Central Bureau*, 2004.

[9] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian. *Deep residual learning for image recognition*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778, 2016.

[10] Russakovsky, Olga; Deng, Jia; Su, Hao; et al. *Imagenet large scale visual recognition challenge. International journal of computer vision*, 115, 211–252, 2015. Publisher: Springer.

[11] Anderson, Matthew; Motta, Ricardo; Chandrasekar, Srinivasan; Stokes, Michael. *Proposal for a Standard Default Color Space for the Internet-sRGB*. In: *Color Imaging Conference*, 6, 1996.

[12] Ebner, Marc. *Color constancy*. Volume: 7, 2007. Publisher: John Wiley & Sons.

[13] Brooks, Tim; Mildenhall, Ben; Xue, Tianfan; Chen, Jiawen; Sharlet, Dillon; Barron, Jonathan T. *Unprocessing images for learned raw denoising*. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11036–11045, 2019.

[14] Chen, Ting; Kornblith, Simon; Norouzi, Mohammad; Hinton, Geoffrey. *A simple framework for contrastive learning of visual representations*. In: *International conference on machine learning*, 1597–1607, 2020. Organization: PMLR.

[15] Kingma, Diederik P; Ba, Jimmy. *Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980*, 2014.

[16] Karaimer, Hakki Can; Brown, Michael S. *A software platform for manipulating the camera imaging pipeline*. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 429–444, 2016. Organization: Springer.

[17] Debevec, Paul E; Malik, Jitendra. *Recovering high dynamic range radiance maps from photographs*. In: *ACM SIGGRAPH 2008 classes*, 1–10, 2008. Publisher: Association for Computing Machinery.

[18] Grossberg, Michael D; Nayar, Shree K. *Determining the camera response from images: What is knowable?*. *IEEE Transactions on pattern analysis and machine intelligence*, 25(11), 1455–1467, 2003. Publisher: IEEE.

[19] Mitsunaga, Tomoo; Nayar, Shree K. *Radiometric self calibration*. In: *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, 1, 374–380, 1999. Organization: IEEE.

[20] Nguyen, Rang MH; Brown, Michael S. *RAW image reconstruction using a self-contained sRGB-JPEG image with only 64 KB overhead*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1655–1663, 2016.

[21] Green, Phil *Colorimetry and colour difference*, In: *Fundamentals and Applications of Colour Engineering, pages 27–52, 2023* Wiley Online Library

[22] Nam, Seonghyeon; Punnappurath, Abhijith; Brubaker, Marcus A; Brown, Michael S. *Learning srgb-to-raw-rgb*

[23] Brown, Michael S; Kim, SJ. *Understanding the in-camera image processing pipeline for computer vision*. In: *IEEE International Conference on Computer Vision (ICCV)-Tutorial*, 3, 1–354, 2019.

[24] Kerr, Douglas A. *The CIE XYZ and xyY color spaces*. *Colorimetry*, 1(1), 1–16, 2010.

[25] Tai, Yu-Wing; Chen, Xiaogang; Kim, Sunyeong; Kim, Seon Joo; Li, Feng; Yang, Jie; Yu, Jingyi; Matsushita, Yasuyuki; Brown, Michael S. *Non-linear camera response functions and image deblurring: Theoretical analysis and practice*. *IEEE transactions on pattern analysis and machine intelligence*, 35(10), 2498–2512, 2013. Publisher: IEEE.

[26] Zamir, Syed Waqas; Arora, Aditya; Khan, Salman; Hayat, Munawar; Khan, Fahad Shahbaz; Yang, Ming-Hsuan; Shao, Ling. *Cycleisp: Real image restoration via improved data synthesis*. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2696–2705, 2020.

[27] Gui, Jie; Chen, Tuo; Cao, Qiong; Sun, Zhenan; Luo, Hao; Tao, Dacheng. *A survey of self-supervised learning from multiple perspectives: Algorithms, theory, applications and future trends*. *arXiv preprint arXiv:2301.05712*, 2023.

[28] Koskinen, Samu; Yang, Dan; Kämäräinen, Joni-Kristian. *Reverse imaging pipeline for raw RGB image augmentation*. In: *2019 IEEE International Conference on Image Processing (ICIP)*, 2896–2900, 2019. Organization: IEEE.

[29] Stewart, Russell; Ermon, Stefano. *Label-free supervision of neural networks with physics and domain knowledge*. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 31, 2017.

[30] Wang, Zhou. *Image quality assessment: from error visibility to structural similarity*. *IEEE transactions on image processing*, 13(4), 600–612, 2004. Publisher: IEEE.

[31] Von Rueden, Laura; Mayer, Sebastian; Beckh, Katharina; Georgiev, Bogdan; Giesselbach, Sven; Heese, Raoul; Kirsch, Birgit; Pfrommer, Julius; Pick, Annika; Ramamurthy, Rajkumar; et al. *Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems*. *IEEE Transactions on Knowledge and Data Engineering*, 35(1), 614–633, 2021. Publisher: IEEE.

[32] Schanda, Janos, editor. *Colorimetry: Understanding the CIE System*. John Wiley and Sons, 2007.

[33] Krizhevsky, Alex and Hinton, Geoffrey and others. *Learning multiple layers of features from tiny images*. Toronto, ON, Canada, 2009

[34] Liu, Xiao and Zhang, Fanjin and Hou, Zhenyu and Mian, Li and Wang, Zhaoyu and Zhang, Jing and Tang, Jie *Self-supervised learning: Generative or contrastive* IEEE, 2021

[35] Tang, Yahui and Chang, Kan and Huang, Mengyuan and Li, Baoxin *BMISP: Bidirectional mapping of image signal processing pipeline* Elsevier, Signal Processing, 2023

[36] Gong, Rui and Wang, Qing and Shao, Xiaopeng and Liu, Jietao *A color calibration method between different digital cameras* Elsevier, Optik, 127, 2016

[37] Svirsky, Jonathan and Lindenbaum Ofir "Interpretable Deep Clustering." International Conference on Machine Learning (ICML). (2024).

[38] Eisenberg, Ran and Svirsky, Jonathan and Lindenbaum, Ofir Self Supervised Correlation-based Permutations for Multi-View Clustering. *ArXiv Preprint ArXiv:2402.16383*. (2024)

[39] Rozner, Amit and Battash, Barak and Wolf, Lior and Lindenbaum, Ofir Domain-Generalizable Multiple-Domain Clustering. *Transactions On Machine Learning Research*. (2023)

Automated Bioacoustic Monitoring for South African Bird Species on Unlabeled Data

Michael Doell[1,*] Dominik Kuehn[1,*] Vanessa Suessle[1,2,3*]
Matthew J. Burnett[2] Colleen T. Downs[2] Andreas Weinmann[3] Elke Hergenroether[1]

* joint first authors
[1]Department of Computer Science, University of Applied Sciences Darmstadt, Schoefferstrasse 3, Darmstadt, Germany
[2]Centre for Functional Biodiversity, School of Life Sciences, University of KwaZulu-Natal, P/Bag X01, Scottsville, Pietermaritzburg, 3209, South Africa
[3]Algorithms for Computer Vision, Imaging and Data Analysis Group, University of Applied Sciences Darmstadt, Schoefferstrasse 3, Darmstadt, Germany
@ michael.doell@extern.h-da.de, dominik.kuehn@extern.h-da.de, vanessa.suessle@h-da.de

ABSTRACT

Analyses for biodiversity monitoring based on passive acoustic monitoring (PAM) recordings is time-consuming and challenged by the presence of background noise in recordings. Existing models for sound event detection (SED) worked only on certain avian species and the development of further models required labeled data. The developed framework automatically extracted labeled data from available platforms for selected avian species. The labeled data were embedded into recordings, including environmental sounds and noise, and were used to train convolutional recurrent neural network (CRNN) models. The models were evaluated on unprocessed real world data recorded in urban KwaZulu-Natal habitats. The *Adapted SED-CRNN* model reached a F1 score of 0.73, demonstrating its efficiency under noisy, real-world conditions. The proposed approach to automatically extract labeled data for chosen avian species enables an easy adaption of PAM to other species and habitats for future conservation projects.

Keywords

Bioacoustic Monitoring, Species Classification, Spectrograms, CNNs, Bidirectional GRU, Ecology, Wildlife Conservation

1 INTRODUCTION

The monitoring of wildlife is essential for wildlife conservation and biodiversity management. While camera traps were the tool of choice for monitoring many species [1, 2], the practicality of relying solely on visual data diminished for certain species because of factors such as body size and behavior [3, 4]. Passive acoustic monitoring (PAM) emerged as an alternative to collect large datasets [5], especially for avian species. The recorded soundscapes gave vital information on the ecosystems, its population and biodiversity [6, 7, 8]. Sound meters enabled a constant and unobtrusive recording of ambient soundscapes, collecting large amounts of data [9]. The manual analyses of such data created a bottleneck for research and the presence of noise complicated the analyses [10, 11]. The classification of avian species using acoustic data is further complicated by factors such as inter-species similarity, overlapping calls and diverse characteristics of intra-species calls.

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.

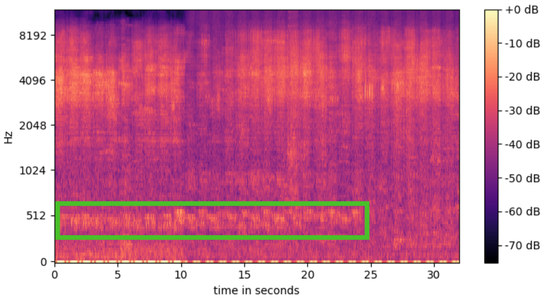


Figure 1: Example of a red-eyed dove Mel-spectrogram with a high level of background noise.

Avian species might have different prominence in the soundscape depending on the intensity and frequency of their calls [6, 8]. The automated analyses of data from camera traps, using artificial intelligence (AI) based methods, has been intensively researched over the past years [12].

Transfer of such methods for audio recordings is still in its infancy compared with camera traps [9]. The conversion of audio recordings into spectrograms facilitated the application of AI-based computer vision methods [13, 14]. With an "image" as input, detection and classification models could be applied to the converted audio recording. But the lack of comprehensive labeled datasets caused, that existing models often exhibit species specificity, limited geographical coverage or required re-training and programming expertise to be applied to new case studies [14, 9, 4]. To the best of our knowledge, available ready-to-use models have not been developed to process large datasets or they required a certain amount of manual labeling.

In this work we developed a framework that can automatically detect the presence of certain avian species without the requirement of prior manual labeling for unprocessed data recorded in noisy environments. This framework exhibits a high degree of generality for a simple extension to other species in the future.

2 RELATED WORK

PAM has been applied to a variety of species, including marine fauna, birds, insects, terrestrial mammals and amphibians [15, 16], leading to a growing research field based on audio data. The analyses of such data presented numerous challenges, depending upon the species, their habitats, and the various types of sound pollution associated with them. Insights derived from this research are valuable for informing economic decision-making processes. For instance, wildlife occurrence assessments can be crucial in areas surrounding wind turbines to ensure the safety of endangered species like birds and bats [17, 18]. Numerous studies have focused on bats due to their reliance on sound for orientation and the informative nature of their calls [16]. Bat calls were detected by converting audio recordings to Mel-frequency cepstral coefficients (MFCCs) and processing them with convolutional neural networks (CNNs) [19]. The identification of bird calls has increasingly attracted research interest as well, prompting the initiation of public challenges like *BirdClef* [20, 21], which provided essential data to accelerate the expansion in avian bioacoustics research. The datasets were focused on species of the northern hemisphere. Existing deep learning models like *Nighthawk*, specialized for short-duration calls, were retrained on the Merlin Sound ID framework with datasets manually labeled for in-flight vocalizations of pre-selected nocturnal migratory species [22]. While existing applications like *BirdNET* [23] offered user-friendly interfaces for bird call identification, they were not designed to handle large datasets and frequently relied on location-specific species recognition. Further research is necessary for species in other geographic locations. Especially for rare species, that are often of special interest, the data availability is sparse [15].

One approach in the field of audio analyses involved treating audio data as a computer vision problem. Audio signals were converted into Mel-spectrograms, which then could be processed using CNNs [24]. The CNN models, which were pretrained on ImageNet, were fine-tuned using labeled audio files, which significantly improved bird call classification and demonstrated their ability to handle the complexity of bioacoustic data [25, 26]. However, generating sufficient labeled data for such analyses is labor and challenging [27]. Data augmentation is an option to efficiently generate labeled datasets. Approaches to achieve labeled images for anomaly detection in an industrial context [28] resonated with our approach to obtain training data by overlaying labeled bird sounds onto background noise, circumventing the scarcity of labeled data.

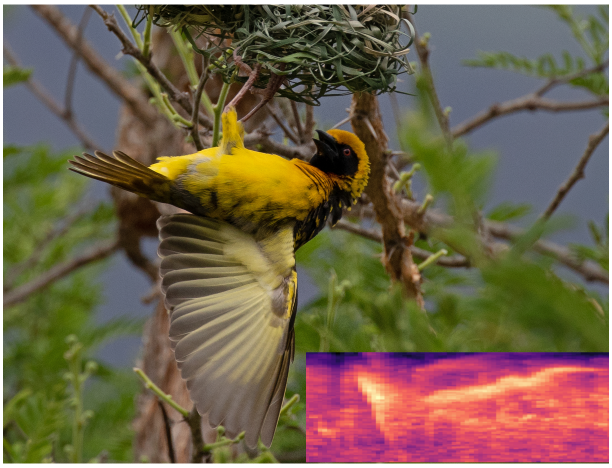


Figure 2: Village weaver while nest building and a spectrogram of its call.

The study [29] that compared models that capture temporal dependencies with those that do not, made findings that hybrid models, incorporating explicit temporal layers, significantly outperformed ImageNet-based models in the classification of bird acoustic data. This comparison underscored that the presence of temporal layers, including mechanisms like long short-term memory (LSTM), gated recurrent units (GRU) in recurrent neural networks (RNN) led to more accurate outcomes.

3 DATASET AND ITS GENERATION

We generated a dataset by automatically combining unlabeled and labeled data from existing sources to train our models. The fundamental dataset consisted of unlabeled audio recordings in which we embedded snippets of labeled data records.

3.1 Data Sources

We utilized 27 unlabeled audio recordings originating from a PAM program at different locations in suburban areas in the KwaZulu-Natal (KZN) Province, South Africa. The dataset was recorded with *Wildlife Acoustics - Song Meter Mini 2 AA*[1] at all locations. Each

---

[1] https://www.wildlifeacoustics.com/products/song-meter-mini-2-aa

recorded file was saved as a *.wav* file with a length of about one hour. On the contrary, the labeled training data were exported from the *Xeno-Canto*[2] database. This is a publicly accessible collection of bird vocalizations from various global regions contributed by ornithologically interested volunteers. Other than the KZN dataset, the samples were recorded by different devices. Each recording on *Xeno-Canto* is labeled with the heard species and a rating on the sound quality from *A* (indicating the highest quality) to *E* (denoting the lowest quality).

To enhance the robustness and generalization capability of our model, we included labeled data across all quality ratings for the training dataset. Recordings for six chosen species were selected (Table 1).

Class	**Latin name**	**No. of Files**
Brown-hooded Kingfisher	*Halcyon albiventris*	70
Dark-capped Bulbul	*Pycnonotus tricolor*	346
Hadada Ibis	*Bostrychia hagedash*	177
Olive Thrush	*Turdus olivaceus*	49
Red-eyed Dove	*Streptopelia semi-torquata*	129
Village Weaver	*Ploceus cucullatus*	134

Table 1: Labeled data records exported from *Xeno-Canto* per selected bird species.

3.2 Combining Labeled and Unlabeled Data

For the generation of the training dataset, files from the unlabeled KZN dataset were used as background and the labeled samples extracted from the *Xeno-Canto* were added to the background file. More precisely, randomly selected labeled samples from the *Xeno-Canto* subset (Table 1) for each of the species of interest were inserted at random locations in the KZN background audio file creating overlaps (Figure 3).

The generated dataset was automatically labeled based on where the samples were embedded into the background file. Audio files can be labeled with different approaches. The labels mark the start and end point of a call based on time. We chose one-dimensional labels for this implementation, where each call was defined by its start and end point. Using the one-dimensional approach increased the range of possibilities to use labeled data from existing platforms, like *Macaulay*[3] and *Xeno-Canto*, for future applications and species. The start and end points of the embedded overlapping samples defined the label for that occurrence of the species. For each KZN background file, the labels were provided as a two-dimensional list, where every occurrence of a species was saved with the start and end point (Figure 3). This process was carried out for a total of 27 KZN files. An equal distribution of files per species were inserted, to cope with the problem of unbalanced datasets. The *fill density* parameter regulated the amount of *Xeno-Canto* audio files that were integrated into a KZN audio file and therewith the relation between labeled bird calls and ambient noise. An exception is the *max fill density* setting: here all available labeled data for the relevant species were embedded in the KZN background files, resulting in a high density of labeled *Xeno-Canto* embeddings, but imbalanced distribution of species. Inspired by the natural behaviour of multiple birds chirping simultaneously, multiple em-

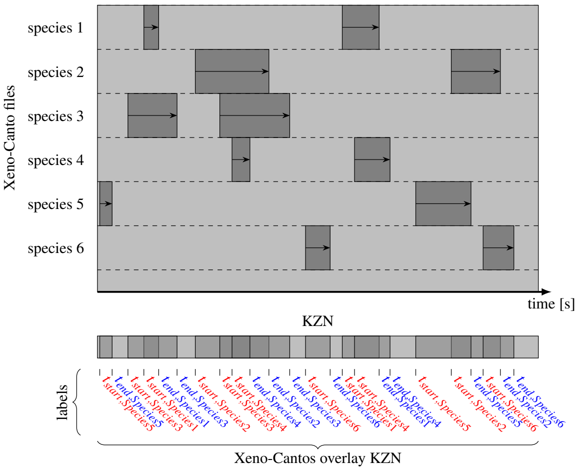


Figure 3: Passive acoustic monitoring data record (KZN) enriched with randomly sampled labeled data snippets: Selecting *k* items for each species and overlaying them at randomly selected timesteps in the KZN background record. Bottom: Darker shades of grey imply a higher overlapping.

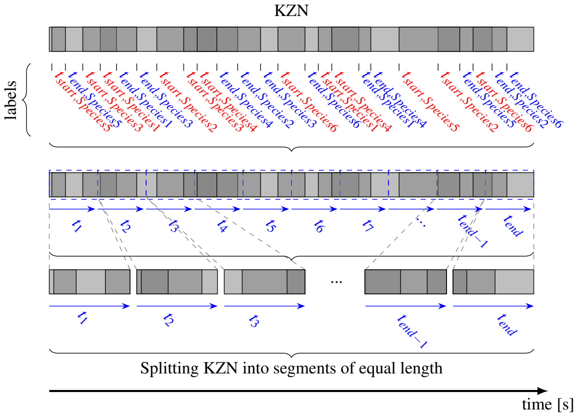


Figure 4: Example of splitting randomly sampled labeled data into segments of equal length of 1 second.

---

[2] https://xeno-canto.org

[3] https://www.macaulaylibrary.org

bedded samples of the same or different species could be overlapping (Figure 3). The generated dataset finally consisted of around 27 hours of the KZN background soundscape with a maximum of 905 embeddings from *Xeno-Canto*.

3.3 Label Formatting

For an unambiguous automatic labeling and an effective training of the neural network, it was crucial to maintain a uniform format. The recorded KZN dataset varied in length and therefore needed to be split into equally long labeled subsequent segments. In our implementation each segment had a length of one second, which would eventually divide a call into multiple segments of one second as shown in Figure 4. Each segment then had an individual binary label for the presence/absence of each species within that time interval $t_i \in [t_0, t_{end}]$. Start and end points of calls were rounded to the closest second and assigned to all segments that lay in that interval.

3.4 Transformation into Mel-spectrograms

The waveform files from the generated dataset were converted into spectrograms, providing the advantage of separating temporal and spectral information, allowing better feature learning [30].

Each segment was transformed from a 1D-audio waveform into a 2D-Mel-spectrogram [31]. Thereby the x-axis represented the time in seconds (s) and the y-axis the frequencies in Hertz (Hz) on the Mel-scale. The colour intensity depicted the magnitude in decibels (dB) (Figure 5). By stacking each Mel-spectrogram in depth, the output data would get reshaped from multiple 2D-Mel-spectrograms into a 3D-shaped cube (Figure 5). The fixed segment length and the predefined frequency range ensured the uniformed shape of the input image, fitting the model's requirements.

3.5 Impact of Noise

Our work aimed at providing a tool to predict the presence of certain bird species from audio files collected in natural habitats. The soundscape of such recordings encompassed a variety of sounds, including anthropophony (human-made sounds), biophony (sounds from non-human organisms), and geophony (natural sounds like wind and rain) [32].

We generated a labeled training dataset, which included different types and intensities of noise, to train our models. This approach offered the dual benefit of preserving the inherent noise within the original dataset and also adding labels for the model training, enabling a robust model training process.

Our unlabeled KZN dataset as well as our labeled *Xeno-Canto* subset included a wide range of different noises, including biophonetic sounds from a broad spectrum of non-target animal species, like insects, canines (barking dogs, *Canis lupus familiaris*) and various non-categorized avian species (geese, chickens), as well as environmental or geophonetic sounds like wind, rain, flowing streams and vegetative movements (rustling leaves). Additionally, anthropogenic noises such as human voices, mechanical noises (e.g. machinery or technical equipment) and motor sounds (cars, motorbikes and trains) were present in part of the audio files. The KZN dataset, recorded in suburban environments had a high noise pollution from anthropogenic origin, especially in the recordings of the urban neighbourhood area.

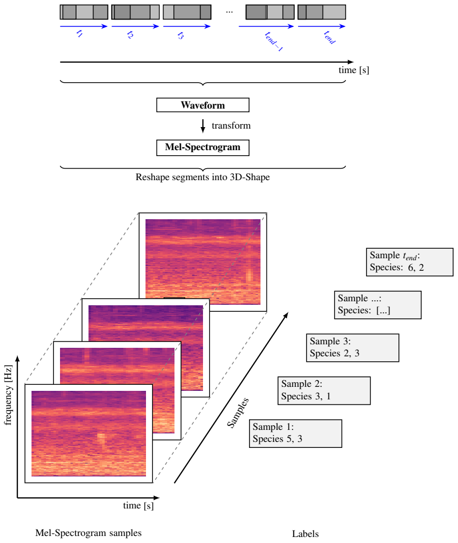


Figure 5: Sequence of reshaping the 2D-Mel-spectrograms into 3D-shaped format to ensure a fixed segment length.

The complexity and variability in noise types obscuring the bird calls of interest, presented a significant challenge for the processing and analysis of the audio data. For machine learning applications, noise could impede the model from identifying and learning relevant features [33]. For the robustness and the ability of the model to adapt for practical use in audio signal processing in diverse and noisy realworld environments, where noise was an inevitable factor, different types of noise had to be considered. By using the KZN samples as a background, the model was confronted with the noise present in the application case during the training process.

4 THE MODELS

The neural network was trained to learn features in the Mel-spectrograms to detect the chosen bird species calls from a soundscape and predicts the species' presence or absence. Because of the overlapping *Xeno-Canto* and KZN background audio data, the model was confronted with variable types of noise during training to simulate a natural environment. Unlabeled bird calls in the original KZN data could not be considered in the training process and were treated as noise.

4.1 Employed Architectures

We utilized convolutional recurrent neural networks (CRNNs), grounded in their widespread recognition as an effective approach for sound event detection (SED) [34]. In particular, we focused on a foundational CRNN model architecture, referred to as *SED-CRNN* [31]. While the *SED-CRNN* in the original study [31] utilized 40 Mel bands as input, we explored an adaptation of this model, named *Adapted SED-CRNN*, where we increased the number of Mel bands to 128, as suggested in their work. Additionally, we employed the *SELDnet* framework [35], omitting its localization component to align with our focus solely on sound event detection.

The initial stage of these CRNN architectures included convolutional blocks, each consisting of a sequence of a convolutional layer, batch normalization, rectified linear unit (ReLU) activation and a max pooling layer, for feature extraction from spectrogram images. Following the convolutional blocks, bidirectional gated recurrent unit (GRU) [36] layers were integrated, which analyzed data sequences in both forward and reverse directions, addressing temporal dynamics. The temporal outputs from the GRU layers were directly fed into dense layers for the final classification task [31, 35]. The model architecture is schematically shown in Figure 6. Sigmoid activation was used in the output layer, paired with binary cross-entropy as loss function [37]. Afterwards a threshold-based binarization step was applied to convert probabilistic outputs into distinct class predictions.

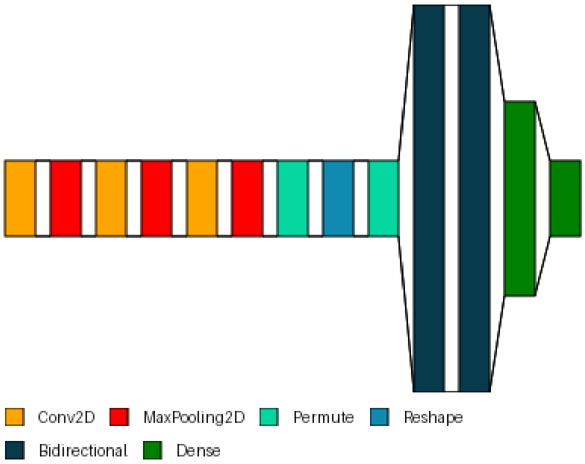


Figure 6: Example of a basic CRNN architecture with different types of layers. All models were based on the this basic structure, but with varying input segments length and number of Mel bands.

4.2 Training the Models

In our experiments, we trained and compared *SED-CRNN*, featuring 40 Mel bands against the *Adapted SED-CRNN* and *SELDnet*, both utilizing higher-dimensional 128 Mel bands. For *SED-CRNN* and the *Adapted SED-CRNN* models, we selected a temporal input window of 5 seconds. The *SELDnet* model required a longer 32 second input window because of additional architectural constraints. Each model processed single-channel inputs. The specifications of the input hyperparameters for each model are summarized in Table 2.

Model	**Mels**	**Segment Length**
SED-CRNN	40	5s
Adapted SED-CRNN	128	5s
SELDnet (w/o loc.)	128	32s

Table 2: Input hyperparameters of the employed model architectures.

All models were trained with a batchsize of 4 and a default of 300 epochs on the generated files combining the KZN dataset and the *Xeno-Canto* subset. Overfitting was mitigated using *EarlyStopping* [38], which terminated training upon halting of loss improvements, resulting in variable epoch count of less than 300 (Table 3). We avoided transfer learning to isolate the impact of our embedding-based pre-processing method. This approach, coupled with non-filtered spectrograms, allowed a direct assessment of how this method performed in bird call classification.

Figure 7: Audio recorder setup in habitat for passive acoustic monitoring.

Model	Fill Density	Epochs	Precision	Recall	F1 Score	Loss	Accuracy
SED-CRNN	10	42	0.34	0.46	0.40	0.30	0.91
	50	39	0.49	0.50	0.50	0.28	0.92
	max	39	0.55	0.45	0.49	0.21	0.95
Adapted SED-CRNN	10	35	0.56	0.74	0.63	0.17	0.95
	50	31	0.67	0.80	0.73	0.14	0.96
	max	31	0.64	0.77	0.70	0.14	0.96
SELDnet (w/o loc.)	10	48	0.32	0.67	0.43	0.35	0.90
	50	80	0.62	0.77	0.69	0.18	0.95
	max	43	0.35	0.76	0.48	0.38	0.89

Table 3: Performance comparison of models based on *fill density* at a threshold of 0.5. *Max fill density* defines that all available labeled data samples for the relevant species were embedded in the background file.

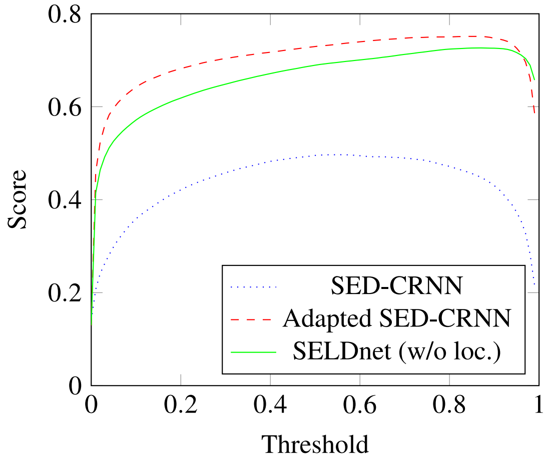


Figure 8: Comparison of F1 scores across different binarization thresholds trained with a fill density of 50.

5 RESULTS

In the evaluation of our models, we primarily focused on precision, recall, and the F1 score, because of the imbalanced nature of our dataset which featured a significantly high proportion of negative samples. We chose precision and recall for their focus on true positives and F1 score to balance these metrics to effectively assess the model performance amidst predominantly occurring negatives. Accuracy might yield misleading results in such contexts, as a model biased towards the majority class could still achieve a high score [39]. However, we have included accuracy in Table 3 to provide a baseline overview of overall model performance and model effectiveness in correctly classifying all samples, both positive and negative.

5.1 Performance of the Models

For a consistent evaluation of all models, we set a threshold of 0.5 to determine the binary classification results, as shown in Table 3. Across all models, we observed a consistent increase in precision, recall, and F1 score for a *fill density* between 10 to 50 samples per KZN background file. At *max fill density*, there was a small decrease in recall and F1 score, indicating an optimal performance around medium *fill density*. The *SED-CRNN* model, characterized by its low-dimensional frequency input, exhibited a unique precision pattern compared with the other models. *SED-CRNN* showed a steady increase in precision, rising from 0.34 at a *fill density* of 10 to 0.55 at *max fill density*, while the remaining models exhibited a slight decrease in precision at *max fill density*. A comparative analysis of F1 scores across the spectrum of binarization thresholds at a *fill density* of 50 (Figure 8), which was the most effective in the evaluation (Table 3). The *Adapted SED-CRNN* model reached a maximum F1 score of 0.73 at a threshold of 0.84. The *SELDnet* model attained its highest F1 score of 0.69 at a threshold of 0.87, indicating a slightly delayed peak performance relative to the *Adapted SED-CRNN*. The *SED-CRNN* model reached its maximum F1 score of 0.5 at a threshold of 0.56.

5.2 Performance in Real-World Application

An unseen validation dataset from the KZN dataset was used as a real world application to validate the framework. The data were recorded in a permanent long-term monitoring, recording continuously 24 hours a day at different locations, capturing different types and levels of noise. Two validation files from two different habitats, one in a suburban neighbourhood and one in a botanical garden with 40 minutes of duration in total. The occurrences of the species: Brown-hooded kingfisher (*Halcyon albiventris*), dark-capped bulbul (*Pycnonotus tricolor*), hadada ibis (*Bostrychia hagedash*), olive thrush (*Turdus olivaceus*), red-eyed dove (*Streptopelia semitorquata*) and village weaver (*Ploceus cucullatus*) were manually labeled by an expert. In the two habitats, different levels of noise were present in the recordings. Both included other species, traffic and biophonic noise, while the data recorded at the neighbourhood had more background noise in terms of human voice and machinery. The detection threshold was set to 0.1 to catch a maximum of potential bird call detections. The best detected bird species, measured by the F1 score, was the brown-hooded kingfisher in the suburban neighbourhood recording with

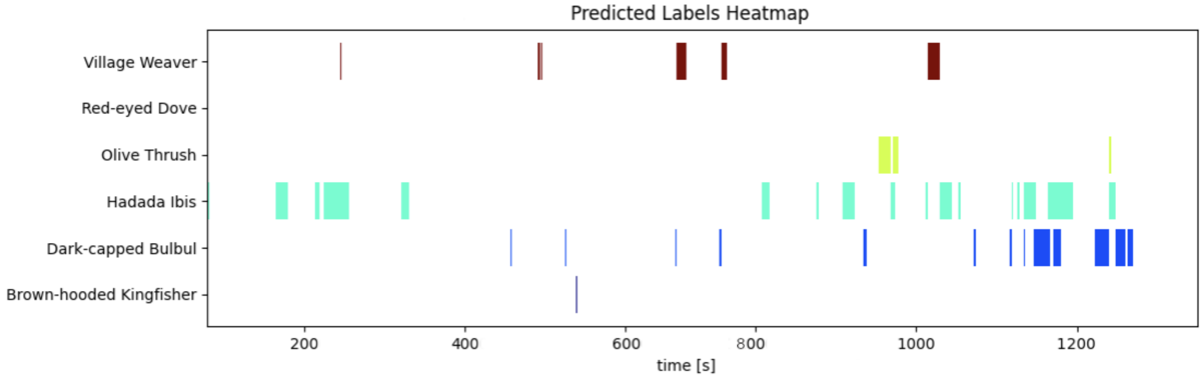


Figure 9: Sample GUI export for predictions. The presence of each species in each time interval is highlighted. Predictions of calls from different species may overlap.

a score of 0.80 (model: *SELDnet*) and a precision of up to 1 (model: *SED-CRNN*), meaning that all brown-hooded kingfisher detections predicted by that model were correct. No brown-hooded kingfisher was present in the botanical garden and therefore no direct comparison was feasible. The best performing model on the present species for the botanical garden reached a F1 score of 0.45 for the dark-capped bulbul (model: *SELDnet*) and a maximum precision for the red-eyed dove (model: *SED-CRNN*). The performance on the raw data was generally lower than on the generated dataset with the embedded *Xeno-Canto* data. The good performance for the dark-capped bulbul species could be reasoned by the highest amount of available training data for this species (Table 1).

The best performing model over all species, measured by the F1 scores, was the *SELDnet* with *max fill density* reaching an F1 score of 0.27 on the botanical dataset and 0.25 on the neighbourhood dataset. The lower value for the neighbourhood recording could be reasoned by the higher level of noise.

Typical misclassifications included children's voices incorrectly classified as hadada ibis. Misclassifications could be caused by the similarity of calls from species that have not been included in the training process or background noises in the *Xeno-Canto* data that were misleadingly learnt by the model as features for a species. The hardest species to classify and validate was the village weaver. At the location of the neighbourhood recording, village weaver nests were present, resulting in nearly constant village weaver callings, whereas at the botanical garden no nests were present and no village weaver called, leading to zeros in the evaluation metrics for that species.

The varying distribution of species, or even their absence at specific recording locations, as well as the proximity of nests to recording sites, posed challenges to the evaluation process.

6 CONCLUSIONS

We developed a robust framework for the detection of certain bird species from noisy PAM data without the requirement of manual labeling. The framework offers a pipeline to extend and train for additional species, not limited to avian species.

The selection of the species might have an impact on the performance of the models, because some species have more prominent calls and therefore were easier to detect and distinguish in spectrograms analyzed by humans and models. The setting of the detection threshold for the model prediction affected the detection rate. Depending on the application, there was a trade-off between prioritizing only secure detections and thereby overlooking less certain ones and wanting all potential detetctions, bearing the risk of having false detections.

In this study we faced the issue of unknown false negative examples. The background noise of the KZN dataset might include calls of the bird species of interest, but were not labeled. More labeled data, also for the background files might improve further model training.

For further research we plan to focus on adding more bird species or even species of other classes like amphibians, mammals or other animals producing sounds. The existing GUI, that was presently only used for testing purposes, could be further improved to be used as a tool by research groups (Figure 9). We plan to implement and train further model architectures besides the *SED-CRNN* and *SELDnet* into the framework. Furthermore, an approach with transformer models will be conducted to test, if timely dependencies in bird calls can be captured more effectively for better classifications [40].

7 ACKNOWLEDGMENTS

The project was made feasible thanks to the assistance of a scholarship provided by the DAAD (German Academic Exchange Service).

REFERENCES

[1] P. Kumar, S. Luo, et al. "A Comprehensive Review of Deep Learning Approaches for Animal Detection on Video Data". In: *International Journal of Advanced Computer Science and Applications* 14 (Dec. 2023). DOI: `10.14569/IJACSA.2023.01411144`.

[2] V. Suessle, M. Arandjelovic, et al. "Automatic Individual Identification of Patterned Solitary Species Based on Unlabeled Video Data". In: *Journal of World Society for Computer Graphics (WSCG)* (2023). DOI: `10.24132/JWSCG.2023.1`.

[3] E. Browning, R. Gibb, et al. "Passive acoustic monitoring in ecology and conservation". In: *WWF Conservation Technology Series* (Oct. 2017). DOI: `10.13140/RG.2.2.18158.46409`.

[4] S. Sharma, K. Sato, et al. "A Methodological Literature Review of Acoustic Wildlife Monitoring Using Artificial Intelligence Tools and Techniques". In: *Sustainability* 15 (Apr. 2023), p. 7128. DOI: `10.3390/su15097128`.

[5] R. Gibb, E. Browning, et al. "Emerging opportunities and challenges for passive acoustics in ecological assessment and monitoring". In: *Methods in Ecology and Evolution* (Oct. 2018). DOI: `10.1111/2041-210X.13101`.

[6] T. Bradfer-Lawrence, C. Desjonqueres, et al. "Using acoustic indices in ecology: Guidance on study design, analyses and interpretation". In: *Methods in Ecology and Evolution* 14.9 (2023), pp. 2192–2204. DOI: `10.1111/2041-210X.14194`.

[7] P. Roe, P. Eichinski, et al. "The Australian Acoustic Observatory". In: *Methods in Ecology and Evolution* 12.10 (2021), pp. 1802–1808. DOI: `10.1111/2041-210X.13660`.

[8] N. Priyadarshani, S. Marsland, et al. "Automated birdsong recognition in complex acoustic environments: a review". In: *Journal of Avian Biology* 49.5 (2018), jav–01447. DOI: `10.1111/jav.01447`.

[9] D. Tuia, B. Kellenberger, et al. "Perspectives in machine learning for wildlife conservation". In: *Nature Communications* 13 (Feb. 2022), p. 792. DOI: `10.1038/s41467-022-27980-y`.

[10] M. Sankupellay and D. A. Konovalov. "Bird call recognition using deep convolutional neural network, ResNet-50". In: 2018. DOI: `10.13140/RG.2.2.31865.31847`.

[11] A. Digby, M. Towsey, et al. "A practical comparison of manual and autonomous methods for acoustic monitoring". In: *Methods in Ecology and Evolution* 4.7 (2013), pp. 675–683. DOI: `10.1111/2041-210X.12060`.

[12] J. Vález, W. McShea, et al. "An evaluation of platforms for processing camera-trap data using artificial intelligence". In: *Methods in Ecology and Evolution* 14.2 (2023), pp. 459–477. DOI: `10.1111/2041-210X.14044`.

[13] Y. Kumar, S. Gupta, et al. "A novel deep transfer learning models for recognition of birds sounds in different environment". In: *Soft Computing* 26 (Feb. 2022), pp. 1–21. DOI: `10.1007/s00500-021-06640-1`.

[14] F. J. B. Sanchez, M. R. Hossain, et al. "Bioacoustic classification of avian calls from raw sound waveforms with an open-source deep learning architecture". In: *Scientific Reports* 11 (2021).

[15] B. Ghani, T. Denton, et al. "Global birdsong embeddings enable superior transfer learning for bioacoustic classification". In: *Scientific Reports* 13 (Dec. 2023). DOI: `10.1038/s41598-023-49989-z`.

[16] L. S. M. Sugai, T. S. F. Silva, et al. "Terrestrial Passive Acoustic Monitoring: Review and Perspectives". In: *BioScience* 69.1 (Nov. 2018), pp. 15–25. DOI: `10.1093/biosci/biy147`.

[17] F. Rehling, A. Delius, et al. "Wind turbines in managed forests partially displace common birds". In: *Journal of Environmental Management* 328 (2023), p. 116968. DOI: `10.1016/j.jenvman.2022.116968`.

[18] S. Richardson, P. Lintott, et al. "Peaks in bat activity at turbines and the implications for mitigating the impact of wind energy developments on bats". In: *Scientific Reports* 11 (Feb. 2021). DOI: `10.1038/s41598-021-82014-9`.

[19] Y. Paumen, M. Maelzer, et al. "Development and test of a bat calls detection and classification method based on convolutional neural networks". In: *Bioacoustics* 31 (Sept. 2021), pp. 1–12. DOI: `10.1080/09524622.2021.1978863`.

[20] H. Klinck, S. Dane, et al. *BirdCLEF 2023*. 2023.

[21] A. Miyaguchi, J. Yu, et al. "Motif Mining and Unsupervised Representation Learning for BirdCLEF 2022". In: (2022), pp. 2159–2167. DOI: `10.48550/arXiv.2206.04805`.

[22] B. M. Van Doren, A. Farnsworth, et al. “Nighthawk: Acoustic monitoring of nocturnal bird migration in the Americas”. In: *Methods in Ecology and Evolution* 15.2 (2024), pp. 329–344. DOI: `10.1111/2041-210X.14272`.

[23] C. Wood, S. Kahl, et al. “The machine learning-powered BirdNET App reduces barriers to global bird research by enabling citizen science participation”. In: *PLoS biology* 20 (June 2022). DOI: `10.1371/journal.pbio.3001670`.

[24] A. Chaturvedi, S. A. Yadav, et al. “Classification of Sound using Convolutional Neural Networks”. In: *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*. 2022, pp. 1015–1019. DOI: `10.1109/IC3I56241.2022.10072823`.

[25] Ã. Incze, H.-B. Jancsó, et al. “Bird Sound Recognition Using a Convolutional Neural Network”. In: *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*. 2018, pp. 295–300. DOI: `10.1109/SISY.2018.8524677`.

[26] N. A. and R. Rajan. “Deep Learning-based Automatic Bird Species Identification from Isolated Recordings”. In: *8th International Conference on Smart Computing and Communications (ICSCC)*. 2021, pp. 252–256. DOI: `10.1109/ICSCC51209.2021.9528234`.

[27] B. Zhu, K. Xu, et al. “Audio Tagging by Cross Filtering Noisy Labels”. In: *Transactions on Audio, Speech, and Language Processing* 28 (2020), pp. 2073–2083. DOI: `10.1109/TASLP.2020.3008832`.

[28] M. Pei, N. Liu, et al. “Self-Supervised Learning for Industrial Image Anomaly Detection by Simulating Anomalous Samples”. In: *International Journal of Computational Intelligence Systems* 16.1 (Sept. 20, 2023), p. 152. DOI: `10.1007/s44196-023-00328-0`.

[29] G. Gupta, M. Kshirsagar, et al. “Comparing recurrent convolutional neural networks for large scale bird species classification”. In: *Scientific Reports* 11.1 (2021), p. 17085. DOI: `10.1038/s41598-021-96446-w`.

[30] K. Choi, G. Fazekas, et al. “Automatic Tagging Using Deep Convolutional Neural Networks”. In: *International Society for Music Information Retrieval Conference*. 2016. DOI: `10.48550/arXiv.1606.00298`.

[31] A. Mesaros, T. Heittola, et al. “Sound Event Detection: A tutorial”. In: *IEEE Signal Processing Magazine* 38.5 (Sept. 2021), 67â83. DOI: `10.1109/msp.2021.3090678`.

[32] B. C. Pijanowski, L. J. Villanueva-Rivera, et al. “Soundscape Ecology: The Science of Sound in the Landscape”. In: *BioScience* 61.3 (Mar. 2011), pp. 203–216. DOI: `10.1525/bio.2011.61.3.6`.

[33] M. J. Bianco, P. Gerstoft, et al. “Machine learning in acoustics: Theory and applications”. In: *The Journal of the Acoustical Society of America* 146.5 (Nov. 2019), pp. 3590–3628. DOI: `10.1121/1.5133944`.

[34] E. Çakır, G. Parascandolo, et al. “Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection”. In: *Transactions on Audio, Speech, and Language Processing* 25.6 (2017), pp. 1291–1303. DOI: `10.1109/TASLP.2017.2690575`.

[35] S. Adavanne, A. Politis, et al. “Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks”. In: *IEEE Journal of Selected Topics in Signal Processing* 13.1 (2019), pp. 34–48. DOI: `10.1109/JSTSP.2018.2885636`.

[36] H. M. Lynn, S. Pan, et al. “A Deep Bidirectional GRU Network Model for Biometric Electrocardiogram Classification Based on Recurrent Neural Networks”. In: *IEEE Access* 7 (2019), pp. 145395–145405. DOI: `10.1109/ACCESS.2019.2939947`.

[37] A. A. Lydia and F. S. Francis. “Multi-Label Classification using Deep Convolutional Neural Network”. In: *2020 International Conference on Innovative Trends in Information Technology (ICITIIT)*. 2020, pp. 1–6. DOI: `10.1109/ICITIIT49094.2020.9071539`.

[38] L. Prechelt. “Early Stopping - But When?” In: *Neural Networks: Tricks of the Trade*. Ed. by G. Montavon, G. B. Orr, et al. Vol. 7700. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 53–67. DOI: `10.1007/978-3-642-35289-8_5`.

[39] P. Lahoti and A. Kumar. “Imbalanced Data Classification using Sampling Techniques and XGBoost”. In: *International Journal of Computer Applications* 182.12 (2018), pp. 19–22. DOI: `10.5120/ijca2018917735`.

[40] Y. Gong, C.-I. Lai, et al. “SSAST: Self-Supervised Audio Spectrogram Transformer”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 10. 2022, pp. 10699–10709. DOI: `10.1609/aaai.v36i10.21315`.

Beyond the Benchmark: Detecting Diverse Anomalies in Videos

Yoav Arad
The Hebrew University of Jerusalem, Israel
yoav.arad@mail.huji.ac.il

Michael Werman
The Hebrew University of Jerusalem, Israel
michael.werman@mail.huji.ac.il

ABSTRACT

Video Anomaly Detection (VAD) plays a crucial role in modern surveillance systems, aiming to identify various anomalies in real-world situations. However, current benchmark datasets predominantly emphasize simple, single-frame anomalies such as novel object detection. This narrow focus restricts the advancement of VAD models. In this research, we advocate for an expansion of VAD investigations to encompass intricate anomalies that extend beyond conventional benchmark boundaries. To facilitate this, we introduce two datasets, HMDB-AD and HMDB-Violence, to challenge models with diverse action-based anomalies. These datasets are derived from the HMDB51 action recognition dataset. We further present Multi-Frame Anomaly Detection (MFAD), a novel method built upon the AI-VAD framework. AI-VAD utilizes single-frame features such as pose estimation and deep image encoding, and two-frame features such as object velocity. They then apply a density estimation algorithm to compute anomaly scores. To address complex multi-frame anomalies, we add deep video encoded features capturing long-range temporal dependencies, and logistic regression to enhance final score calculation. Experimental results confirm our assumptions, highlighting existing models limitations with new anomaly types. MFAD excels in both simple and complex anomaly detection scenarios.

Keywords

Video Anomaly Detection, Computer Vision, Smart Surveillance Systems

1 INTRODUCTION

As the volume of recorded video content continues to grow, the need for robust and efficient video anomaly detection methods increases. The ability to automatically identify unusual events or behaviors within videos not only holds the promise of enhancing security but also offers the potential to reduce the manpower required for monitoring. However, achieving truly effective video anomaly detection remains a significant unsolved challenge, due to the diverse range of anomalies that can occur in real-world scenarios.

By nature, anomalous behaviors are rare. Thus, video anomaly detection (VAD) is often treated as a semi-supervised problem, where models are trained exclusively on normal videos and must subsequently distinguish between normal and abnormal videos during inference.

While current benchmark datasets vary in complexity, they share a common limitation in their narrow definition of anomalies. The three datasets, UCSD Ped2 [1], CUHK Avenue [2], and ShanghaiTech Campus [3], tend to limit anomalies primarily to novel object detection (Ped2, ShanghaiTech) or simple movement anomalies (Avenue).

Recent advancements in video anomaly detection predominantly relied on analyzing a few frames or even individual frames in isolation. Researchers predominantly choose between two approaches: reconstruction-based and prediction-based methods. Reconstruction-based methods [4–8] typically employ auto-encoders to learn representations of normal frames, reconstructing them accurately, while anomalous frames result in a higher reconstruction error. Prediction-based methods [3, 9–11] focus on predicting the next frame from a sequence of consecutive frames.

These few-frame based methods achieved impressive results, surpassing an AUC score of 99% [12, 13] on Ped2, over 93% [12] on Avenue, and exceeding 85% [12, 14] on ShanghaiTech, the most complex of the benchmark datasets.

Without a shift in research focus and assumptions, the existing datasets, results, and recurring research patterns may suggest that the field of video anomaly detection is nearing a plateau.

This paper emphasizes the necessity of broadening the scope of what constitutes an anomaly. We propose two novel datasets specifically designed to assess the detection of complex action-based anomalies.

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.

https://www.doi.org/10.24132/CSRN.3401.5

These datasets, referred to as HMDB-AD and HMDB-Violence, build upon the HMDB51 action recognition dataset and define different actions as normal or abnormal activities. By analyzing the performance of various methods on our datasets, we underscore the limitations of existing approaches and advocate for further research on more comprehensive anomaly types.

Building upon the foundation laid by AI-VAD [12], we introduce Multi-Frame Anomaly Detection (MFAD), a novel method aimed at achieving balanced performance, excelling in both simple and complex anomaly detection. AI-VAD utilizes a two-step approach: first, it extracts multiple features and then employs density estimation algorithms to calculate anomaly scores. In their work, they rely on single-frame features like deep image encoding (using a pretrained encoder) and human pose estimations, along with two-frame features such as object velocity. We extend this method by introducing deep video encoding features to capture multi-frame, long-range temporal relationships. MFAD adheres to the AI-VAD framework, computing final scores for each feature using a density estimation algorithm. Additionally, we incorporate logistic regression to enhance the relationships between different feature scores and achieve more accurate final scores.

We extensively evaluate our method on classic benchmark datasets as well as on our newly proposed datasets. The experiments validate the added value of both video encoding features and the logistic regression module. Our method achieves competitive results on Ped2, Avenue, and ShanghaiTech, and greatly outperforms recent methods on HMDB-AD and HMDB-Violence. As a result, it offers a more versatile video anomaly detection solution capable of detecting a broader range of anomalies across various scenarios.

Our key contributions are:

- We highlight the limitations of current video anomaly detection benchmarks and advocate for further research in general video anomaly detection.
- We present MFAD, a novel method capable of effectively handling both simple, few-frame anomalies and complex, multi-frame anomalies.
- We provide two datasets designed for assessing a model's performance on multi-frame action-based anomalies.

2 RELATED WORK

2.1 Video Anomaly Detection Datasets

The datasets commonly used in video anomaly detection can be broadly categorized into two groups, reflecting the shift brought about by the advent of deep learning from approximately 2013 to 2018.

Early datasets are notably smaller and often considered practically solved, include Subway Entrance [15], Subway Exit [15], UMN [16], UCSD Ped1 [1], UCSD Ped2 [1], and CUHK Avenue [2]. Except UMN, these datasets feature only a single scene.

In contrast, more recent datasets have grown significantly in both scale and complexity. This newer group includes ShanghaiTech Campus [3], Street Scene [17], IITB Corridor [18], UBNormal [19], and the most recent and largest of them all, NWPU Campus [20].

It is worth noting that among these datasets, only three have gained popularity as benchmarks: UCSD Ped2, CUHK Avenue, and ShanghaiTech Campus. However, as discussed in this paper, each of these benchmarks has its own set of limitations that motivate the need for further research in the field of video anomaly detection.

Other datasets that can be considered are UCF-Crime [21] and XD-Violence [22]. These datasets are built for fully supervised VAD learning and therefore are orders of magnitude larger than current benchmarks for unsupervised VAD such as this work. We follow previous studies and don't use them for our comparisons.

2.2 HMDB51 Action Recognition Dataset

The HMDB51 [23] dataset, originally designed for action recognition (AR), is relatively small in scale. It is a collection of 6,766 video clips distributed across 51 distinct categories. Most other datasets are significantly larger and more diverse: SSv2 [24], Kinetics-400 [25], Kinetics-600 [26], Kinetics-700-2020 [27] each consist of hundreds of thousands of frames and hundreds of different classes.

The HMDB51 dataset draws content from various sources, ensuring diversity. In this dataset, each class consists of no less than 101 video clips.

2.3 Video Anomaly Detection Methods

Hand-crafted feature based methods

Numerous methods, spanning both classical and contemporary approaches, adhere to a two-stage anomaly detection framework. This framework involves an initial step of extracting hand-crafted features, specifically selected by the researcher and not learned through a deep neural network model. Subsequently, another algorithm is applied to compute anomaly scores.

Early techniques used classic image and video features, including the histogram of oriented optical flow (HOF) [28–31], histogram of oriented gradients (HOG) [31], and SIFT descriptors [32]. In more recent developments, the proliferation of deep learning has facilitated the adoption of off-the-shelf models, such as object detectors, for feature extraction. For instance, in the case of AI-VAD [12], a combination of pose estimations, optical flow predictions, object detection, and deep image encodings is used to construct robust feature representations.

Following feature extraction, classical methodologies often employed scoring techniques such as density estimation algorithms [33–35]. Recent approaches have demonstrated the effectiveness of integrating these features with another learning model [13].

Reconstruction and prediction based methods

In recent years, the increasing prominence of deep learning has driven the widespread adoption of both reconstruction and prediction based methods in video anomaly detection.

Reconstruction-based [4–8] approaches often utilize auto-encoders to learn representations of normal video frames and subsequently detect abnormal frames by identifying higher reconstruction errors. However, the powerful generalization ability of modern auto-encoders can often also reconstruct anomalies. Thus, making it harder to differentiate normal and abnormal frames.

Prediction-based [3, 9–11] models forecast the subsequent frame by leveraging a sequence of preceding frames, employing time sensitive architectures such as LSTMs, memory networks, 3D auto-encoders and transformers. This predictive approach often yields superior results compared to similar reconstruction-based techniques [11], as it captures more complex forms of anomalies. Nevertheless, with the minimal differences between consecutive video frames, these methods face similar challenges to reconstruction-based approaches with respect to modern generators.

Auxiliary tasks methods

Expanding beyond reconstruction and prediction, some models incorporate diverse self-supervised auxiliary tasks, with task success determining frame anomaly scores. These tasks include jigsaw puzzles [36], time direction detection [37], rotation prediction [38], and more. SSMTL++ [14, 39] train a single deep backbone on multiple self-supervised tasks and achieve state-of-the-art results on the benchmark datasets.

3 PROPOSED DATASETS

We introduce two novel datasets designed to assess the capability of various models in detecting forms of anomalies not covered by existing benchmarks. These datasets emphasize action-based anomalies, a category absent in current benchmarks. The first dataset, referred to as HMDB-AD, aligns with the conventional definition of normal activities (walking and running) but challenges models with abnormal behaviors that demand a broader context for detection (climbing and performing a cartwheel). In contrast, the larger and more intricate HMDB-Violence dataset divides 16 action categories into 7 violent (abnormal) and 9 non-violent (normal) activities. This categorization necessitates models to consider a wide range of behaviors when classifying events as either normal or abnormal, making it a closer representation of real-world scenarios.

HMDB-AD dataset

HMDB-AD is the simpler dataset among the two introduced in this paper. It consists of 995 video clips, divided into 680 training videos and 315 testing videos. Normal activities within this dataset are running and walking, aligning with their respective HMDB51 classes. Abnormal activities are climbing and performing a cartwheel. The training dataset contains only of normal videos: 207 running videos and 473 walking videos. Meanwhile, the test dataset has both abnormal videos and randomly selected normal videos; 107 cartwheel videos, 108 climbing videos, 25 running videos, and 75 walking videos. Frames from the videos can be viewed in Appendix A.1.

HMDB-Violence dataset

HMDB-Violence is the larger and more complex of the two datasets presented in this paper. It has 2,566 videos, with a distribution of 1,601 training videos and 965 testing videos. The train set has nine normal categories: running (221 videos), walking (517), waving (98), climbing (104), hugging (110), throwing (96), sitting (134), turning (222), and performing a cartwheel (99). In the test set, there are seven abnormal categories: falling (136), fencing (116), hitting (127), punching (126), using a sword (127), shooting (103), and kicking (130). Additionally, the test set includes 100 videos randomly sampled from the various normal categories: turning (18), walking (31), running (11), sitting (8), hugging (8), performing a cartwheel (8), climbing (4), throwing (6), and waving (6). The abnormal activities in HMDB-Violence are characterized by their violent nature. Examples can be viewed in Appendix A.2.

Annotations

We maintain a consistent labeling for every frame within a video. If a video represents a normal action category, all its frames are labeled as normal. Conversely, if it belongs to an abnormal action category, all frames are marked as abnormal. This simple labeling approach works, as the actions within these videos effectively occupy the entire duration, leaving minimal room for unrelated "spare" frames.

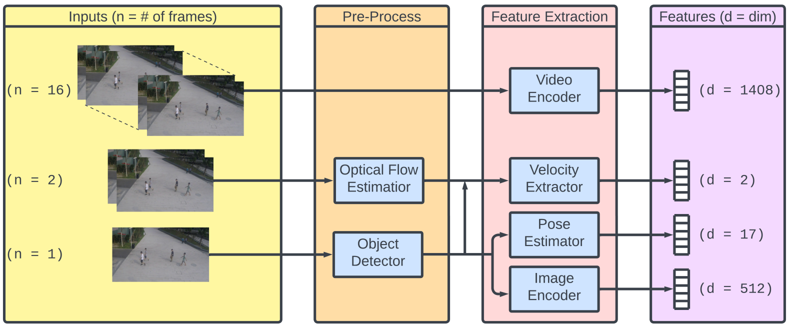


Figure 1: An overview of our feature extraction process.

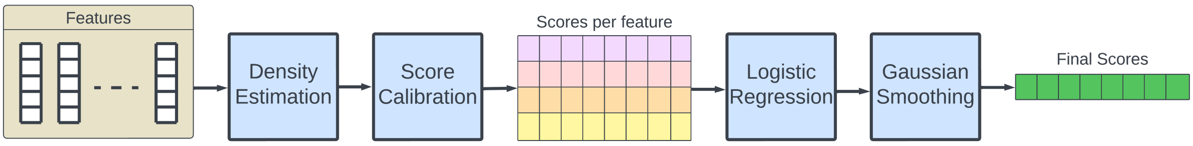


Figure 2: An overview of our anomaly score calculation during inference.

4 MFAD: MULTI-FRAME ANOMALY DETECTION

Our method, MFAD, consists of three key stages: feature extraction, per-feature score computation, and logistic regression. We extract four types of features: object velocities, human pose estimations, deep image encodings, and deep video encodings. For each of these features, we independently calculate density scores. We then employ a logistic regression model to optimally fuse the scores across these four feature kinds. Lastly, we smooth, Gaussian, to produce the final anomaly scores. An overview of our method can be found in Fig. 1, Fig. 2.

4.1 Feature Extraction

Few-Frame Features

In line with AI-VAD [12], HF2-VAD [13], we extract object bounding boxes and optical flows from each frame. We then extract human pose estimations, object velocities, and deep image encodings. These features are derived from individual frames (pose and image encoding) or pairs of frames (velocity) enabling the detection of straightforward anomalies such as novel objects.

Multi-Frame Features

Recognizing the necessity for detecting complex anomalies that span multiple frames, we introduce a deep video encoder. This encoder captures features in a manner similar to deep image encoding but takes into account longer frame sequences (in our case, 16 frames). For this purpose, we leverage VideoMAEv2 [46], a state of the art video foundation model. Subsequently, we process these features in a fashion similar to AI-VAD [12].

4.2 Density Score Calculation

We employ a Gaussian Mixture Model (GMM) for the two-dimensional velocity features and the k-nearest neighbors (kNN) algorithm for the high-dimensional pose, image encoding, and video encoding features. Subsequently, we compute the minimum and maximum density scores for the training set and use them to calibrate the test scores during inference.

Max Feature

We add a fifth feature, denoted as max. After calculating the density scores per feature, we aggregate them into a new feature that holds the maximum feature score per frame.

$$\text{max} = \max\{\text{P, V, IE, VE}\} \in [0,1]^{\#frames}$$

Our experiments show the added value of this feature.

4.3 Logistic Regression

To improve the accuracy of our final anomaly score computation, we incorporate logistic regression as the final step of our method. In this setup, we denote $X \in [0,1]^{\#frames \times \#features}$ as the feature matrix and $y \in \{0,1\}^{\#frames}$ as our ground truth labels. Our final scores are:

$$h_\theta(X) = \sigma(WX + B)$$

Method	Ped2	Avenue	ShanghaiTech	HMDB-AD	HMDB-Violence
HF2-VAD [13]	**99.3%**	91.1%	76.2%	–	–
AED [40]	98.7%	92.3%	82.7%	–	–
HSC-VAD [41]	98.1%	**93.7%**	83.4%	–	–
DLAN-AC [42]	97.6%	89.9%	74.7%	–	–
SSMTL [39]	97.5%	91.5%	82.4%	–	–
LBR-SPR [43]	97.2%	90.7%	72.6%	–	–
AMMCNet [44]	96.6%	86.6%	73.7%	–	–
AI-VAD [12]	99.1%	93.3%	**85.9%**	*70.1%*	*70.5%*
Jigsaw Puzzles [36]	99.0%	92.2%	84.3%	53.8%	52.7%
MNAD [11]	97.0%	88.5%	70.5%	56.3%	51.3%
MPN [45]	96.9%	89.5%	73.8%	58.8%	53.7%
MFAD (Ours)	*99.0% ± 0.5%*	*92.9% ± 0.5%*	*84.8% ± 0.4%*	**90.0% ± 0.4%**	**80.2% ± 0.2%**
MFAD w/o IE (Ours)	98.4% ± 0.7%	90.7% ± 0.5%	85.0% ± 0.4%	86.9% ± 0.5%	76.0% ± 0.2%

Table 1: Comparison to frame-level AUC. Best (bold), second (underlined), and third (italic). IE, denotes image encoding features.

where $\sigma(t) = \frac{1}{1+e^{-t}}$ is the sigmoid function and $\theta = (W,B)$ are the parameters we want to optimize. Our loss function is:

$$L(h_\theta(X),y) = -y\log(h_\theta(X)) - (1-y)\log(1-h_\theta(X))$$

During its training phase, we randomly sample a small fraction of the test frames for model training, while the remainder is used for evaluation. It is crucial to emphasize that the frames utilized for training are excluded from the evaluation process for our reported results, ensuring the validity of our findings.

The final step in our method is applying Gaussian smoothing to the anomaly scores.

5 EXPERIMENTS

5.1 Datasets

In addition to HMDB-AD and HMDB-Violence, we evaluate MFAD on the three benchmark video anomaly detection datasets: UCSD Ped2, CUHK Avenue, and ShanghaiTech Campus. These datasets are primarily outdoor surveillance camera footage, with the sole normal activity being pedestrian movement.

UCSD Ped2

The UCSD Ped2 dataset has 16 training videos and 12 testing videos, all situated within a single scene. Abnormal events in this dataset include the appearance of skateboards, bicycles, or cars within the video frame. Videos are standardized to a resolution of 240×360 pixels.

CUHK Avenue

The CUHK Avenue dataset has 16 training videos and 21 testing videos, all within a single scene. Anomalies within this dataset are activities such as running, throwing objects, and bike riding. All videos have a resolution of 360×640 pixels.

ShanghaiTech Campus

ShanghaiTech Campus stands as the largest and most complex dataset among the three, featuring 330 training videos and 107 testing videos distributed across 13 distinct scenes. Notably, two of these scenes involve non-stationary cameras, resulting in varying angles between videos of the same scene. Abnormal events primarily include running and the presence of cars and bikes. All videos have a resolution of 480×856 pixels.

5.2 Implementation details

We adopt the code from AI-VAD for extracting velocity, pose, and deep image encoding features. For our new deep video features, we leverage the state-of-the-art video foundation model, VideoMAEv2 [46], with the publicly available pretrained weights, fine-tuned on the SSv2 dataset (*vit_g_hybrid_pt_1200e_ssv2_ft*). Our encoding process is carried out on non-overlapping consecutive blocks of 16 frames, extracting Temporal Action Detection (TAD) features for each block. In our experiments, we found no difference in results between non-overlapping blocks and sliding-window blocks. When employing the nearest neighbors algorithm to the video encoding features, we set $k = 1$.

We employ AlphaPose for pose estimation, derive object velocity through optical flows computed via FlowNet2, and utilize YOLOv3 for object detection. For deep image encoding, we leverage CLIP, using a ViT-32 backbone.

Our code and a setup guide are available on `https://github.com/yoavarad/MFAD`.

5.3 Anomaly Detection Results

Our results are based on our optimal model configuration, see Section 5.4. This configuration involves leveraging all four feature types and the max feature while training a logistic regression model on a random 2% of the test set frames for computing final anomaly scores. It is crucial to note that the data used for training the

Configuration	Ped2	Avenue	ShanghaiTech	HMDB-AD	HMDB-Violence
VE	80.3%	87.9%	71.3%	84.9%	75.8%
P + V [12]	98.7%	86.8%	**85.9%**	54.2%	56.1%
P + V + IE [12]	**99.1%**	**93.5%**	85.1%	71.2%	67.9%
P + V + VE	95.8%	91.0%	83.5%	77.8%	70.3%
P + V + IE + VE	96.8%	92.6%	83.0%	82.9%	75.2%
P + V + IE + VE + max	97.0%	92.8%	82.1%	**85.1%**	**76.7%**

Table 2: Comparison of different model configurations, evaluating the impact of various feature types, including pose features (P), velocity features (V), image encoding features (IE), and video encoding features (VE), on the model's performance. max is the max value between {P, V, IE, VE}. Best and second best results are in bold and underlined, respectively.

logistic regression model is not included in the evaluation. To ensure reliability, we repeat this final step 100 times and report the mean AUC result along with the standard deviation. The consistently low standard deviation across all datasets underscores the stability of our method.

MFAD demonstrates competitive results on the well-established benchmark datasets, with modest differences of approximately -0.3%, -0.8%, and -1.1% from the state-of-the-art results on Ped2, Avenue, and ShanghaiTech, respectively. The true strength of our approach becomes evident when applied to the newly introduced datasets, HMDB-AD and HMDB-Violence. On these datasets, we achieve substantial improvements of 19.9% and 9.7%, respectively.

MFAD was tested against four different methods on these new datasets, including AI-VAD [12], upon which our work is built and is the state-of-the-art on the ShanghaiTech dataset. This substantial enhancement highlights the generalizability of our approach to various complex anomalies, without majorly impacting our detection ability of simple anomalies, underscoring the significance of our contributions. For detailed comparison see Table 1. We further report the configuration of MFAD without image encoding (IE) features, improving results on ShanghaiTech by 0.2%.

MFAD faces similar challenges to previous methods when evaluated against the benchmark datasets. Particularly, the object-oriented aspect of MFAD struggles when confronted with scenarios involving closely clustered pedestrians.

In addition to quantitative evaluations, we conducted qualitative analyses on videos from the ShanghaiTech dataset, which feature more complex anomalies beyond novel object detection. These anomalies are shown in Appendix B. The positive impact of our method is clearly evident in Fig. 3, where abnormal frames receive higher anomaly scores, while normal frames receive lower anomaly scores, further validating our method.

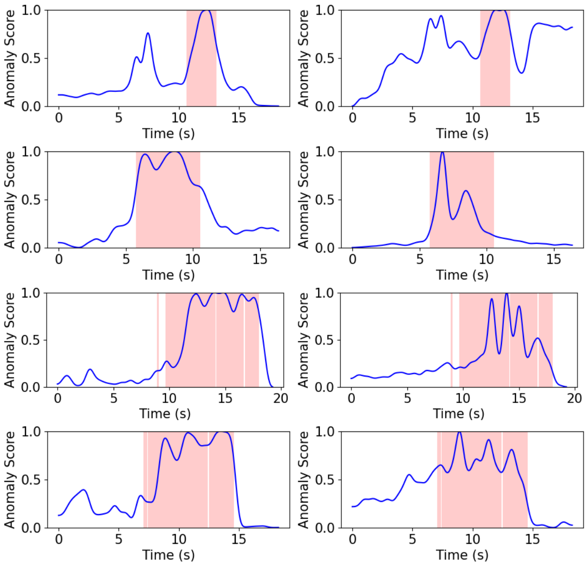


Figure 3: Qualitative results from four ShanghaiTech videos: 01_0028, 03_0032, 03_0039, 07_0008 (respectively). In each pair, MFAD (left) is compared to AI-VAD [12] (right). Anomalous sections are highlighted in red, while the anomaly scores, ranging from 0 to 1, are the blue line. These videos feature complex, behavior-based anomalies rather than novel object detection scenarios, that are more common in this dataset. Clearly, MFAD improves both detecting anomalies and accurately assessing normal parts of the video. Best viewed in color.

5.4 Ablation Study

We perform an ablation study to determine two factors: the added benefit of the video encoding feature, and the most favorable configuration for the logistic regression module.

Feature Selection

In their ablation study, AI-VAD [12] demonstrated the incremental value of their three distinct feature types: pose estimation, deep image encoding, and velocity features, as well as the added effect of Gaussian

Configuration	Ped2	Avenue	ShanghaiTech	HMDB-AD	HMDB-Violence
$\alpha = 0\%$	96.8%	92.6%	83.0%	82.9%	75.2%
$\alpha = 1\%$	98.5% ± 1.1%	92.5% ± 0.6%	84.5% ± 0.6%	89.6% ± 0.6%	79.6% ± 0.3%
$\alpha = 2\%$	99.0% ± 0.6%	92.7% ± 0.7%	84.7% ± 0.4%	89.9% ± 0.4%	79.7% ± 0.2%
$\alpha = 3\%$	99.2% ± 0.5%	92.7% ± 0.7%	84.8% ± 0.4%	89.9% ± 0.3%	79.7% ± 0.1%
$\alpha = 4\%$	99.4% ± 0.3%	92.7% ± 0.7%	84.7% ± 0.3%	89.9% ± 0.3%	79.8% ± 0.1%
$\alpha = 5\%$	99.4% ± 0.4%	93.0% ± 0.6%	84.8% ± 0.3%	90.0% ± 0.2%	79.8% ± 0.1%
$\alpha = 10\%$	99.5% ± 0.2%	92.9% ± 0.6%	84.8% ± 0.2%	90.1% ± 0.2%	79.8% ± 0.1%
$\alpha = 20\%$	99.6% ± 0.1%	93.1% ± 0.5%	84.8% ± 0.2%	90.2% ± 0.2%	79.8% ± 0.1%
$\alpha = 50\%$	**99.7% ± 0.1%**	93.1% ± 0.3%	84.8% ± 0.2%	90.2% ± 0.2%	79.7% ± 0.3%
$\alpha = 90\%$	99.7% ± 0.3%	**93.2% ± 0.6%**	84.8% ± 0.7%	90.2% ± 0.5%	79.8% ± 0.8%
$\alpha = 0\%$ + max	97.0%	92.8%	82.1%	85.1%	76.7%
$\alpha = 1\%$ + max	98.5% ± 0.8%	92.5% ± 0.6%	84.5% ± 0.6%	89.8% ± 0.5%	80.2% ± 0.3%
$\alpha = 2\%$ + max	99.0% ± 0.5%	92.9% ± 0.5%	84.8% ± 0.4%	90.0% ± 0.4%	**80.2% ± 0.2%**
$\alpha = 3\%$ + max	99.1% ± 0.7%	92.9% ± 0.6%	85.0% ± 0.3%	90.0% ± 0.3%	**80.2% ± 0.2%**
$\alpha = 4\%$ + max	99.3% ± 0.5%	93.0% ± 0.5%	85.1% ± 0.3%	90.1% ± 0.3%	**80.2% ± 0.2%**
$\alpha = 5\%$ + max	99.3% ± 0.4%	93.0% ± 0.4%	85.1% ± 0.3%	90.2% ± 0.3%	**80.2% ± 0.2%**
$\alpha = 10\%$ + max	99.5% ± 0.2%	93.0% ± 0.4%	85.2% ± 0.2%	90.2% ± 0.2%	**80.2% ± 0.2%**
$\alpha = 20\%$ + max	99.6% ± 0.1%	93.0% ± 0.3%	85.2% ± 0.2%	90.3% ± 0.2%	80.1% ± 0.2%
$\alpha = 50\%$ + max	**99.7% ± 0.1%**	93.0% ± 0.3%	**85.3% ± 0.2%**	**90.4% ± 0.2%**	80.1% ± 0.3%
$\alpha = 90\%$ + max	99.7% ± 0.2%	93.1% ± 0.6%	85.3% ± 0.6%	90.4% ± 0.5%	80.1% ± 0.8%

Table 3: Performance comparison between various model configurations, with different amounts of training data for the logistic regression model. α represents the proportion of test set frames employed for the training, with these frames excluded from model evaluation. We repeat the process 100 times, and both mean and standard deviation values are reported. The first half uses the basic four features, and the second half also uses the extra max feature. Best and second best results are highlighted in bold and underlined, respectively. The minimal difference in results between different values of $\alpha > 0\%$ is evident.

smoothing. Expanding upon their work, we test the impact of incorporating deep video encoding features in different forms. Our study consists of tests involving both video encoding features in isolation, the combination of all four feature types, and the addition of an extra max feature, that has the value of the maximum feature score per each frame. Furthermore, we explore the substitution of image encoding features with video encoding features due to their semantic similarity.

As presented in Table 2, utilizing solely video encoding features yields impressive performance for multi-frame anomalies. However, this specialization comes at the cost of lower performance on the traditional video anomaly detection datasets. On the other hand, employing all four feature types results in a comprehensive and well-balanced model that performs admirably across all datasets, even though it may not achieve the top rank in any specific dataset.

Logistic Regression

When incorporating the logistic regression model, we conducted experiments to assess the impact of varying amounts of additional training data extracted from the testing set. Specifically, we explored using 1-5%, 10%, 20%, 50%, 90% of the frames for the training. We used both the configuration using only the four basic features and the configuration also using the additional max feature. The results, as presented in Table 3, indicate that the amount of extra training data has minimal effects, as long as there is some extra data. We repeated each configuration 100 times and reported both the mean and standard deviation values. The consistently low standard deviation values observed across all configurations and datasets underscore the robustness of our approach. We chose 2% extra data with the max feature as the optimal trade-off between extra data and efficacy.

6 CONCLUSION

Our paper introduces a broader interpretation of anomalies, encompassing both simple anomalies, commonly found in existing benchmarks, and multi-frame complex anomalies. Building upon the foundation laid by AI-VAD [12], we present a novel method that achieves state-of-the-art performance on our proposed datasets while remaining competitive with recent methods on benchmark datasets. We introduce two new datasets of varying complexity, designed to assess the ability of future models to detect complex action-based anomalies.

In future work, we aim to explore even more intricate types of anomalies, such as location and time-based anomalies (e.g. detecting normal actions occurring at abnormal locations or times) thus further advancing the field of general anomaly detection in videos.

REFERENCES

[1] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, “Anomaly detection in crowded scenes,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, USA: IEEE, Jun. 2010, pp. 1975–1981. [Online]. Available: http://ieeexplore.ieee.org/document/5539872/

[2] C. Lu, J. Shi, and J. Jia, “Abnormal Event Detection at 150 FPS in MATLAB,” in *2013 IEEE International Conference on Computer Vision*, Dec. 2013, pp. 2720–2727, iSSN: 2380-7504.

[3] W. Liu, W. Luo, D. Lian, and S. Gao, “Future Frame Prediction for Anomaly Detection - A New Baseline,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6536–6545. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Liu_Future_Frame_Prediction_CVPR_2018_paper

[4] T.-N. Nguyen and J. Meunier, “Anomaly Detection in Video Sequence With Appearance-Motion Correspondence,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1273–1283. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Nguyen_Anomaly_Detection_in_Video_Sequence_With_Appearance-Motion_Correspondence_ICCV_2019_paper

[5] W. Luo, W. Liu, and S. Gao, “Remembering history with convolutional LSTM for anomaly detection,” in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2017, pp. 439–444, iSSN: 1945-788X.

[6] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel, “Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection,” Aug. 2019, arXiv:1904.02639 [cs]. [Online]. Available: http://arxiv.org/abs/1904.02639

[7] Y. Fan, G. Wen, D. Li, S. Qiu, and M. D. Levine, “Video Anomaly Detection and Localization via Gaussian Mixture Fully Convolutional Variational Autoencoder,” May 2018, arXiv:1805.11223 [cs]. [Online]. Available: http://arxiv.org/abs/1805.11223

[8] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, “Learning Temporal Regularity in Video Sequences,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 733–742. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/Hasan_Learning_Temporal_Regularity_CVPR_2016_paper

[9] Y. Lu, K. M. Kumar, S. s. Nabavi, and Y. Wang, “Future Frame Prediction Using Convolutional VRNN for Anomaly Detection,” in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Sep. 2019, pp. 1–8, iSSN: 2643-6213.

[10] G. Yu, S. Wang, Z. Cai, E. Zhu, C. Xu, J. Yin, and M. Kloft, “Cloze Test Helps: Effective Video Anomaly Detection via Learning to Complete Video Events,” in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM ’20. New York, NY, USA: Association for Computing Machinery, Oct. 2020, pp. 583–591. [Online]. Available: https://dl.acm.org/doi/10.1145/3394171.3413973

[11] H. Park, J. Noh, and B. Ham, “Learning Memory-Guided Normality for Anomaly Detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 372–14 381. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Park_Learning_Memory-Guided_Normality_for_Anomaly_Detection_CVPR_2020_paper

[12] T. Reiss and Y. Hoshen, “Attribute-based Representations for Accurate and Interpretable Video Anomaly Detection,” Dec. 2022, arXiv:2212.00789 [cs]. [Online]. Available: http://arxiv.org/abs/2212.00789

[13] Z. Liu, Y. Nie, C. Long, Q. Zhang, and G. Li, “A Hybrid Video Anomaly Detection Framework via Memory-Augmented Flow Reconstruction and Flow-Guided Frame Prediction,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 588–13 597. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2021/html/Liu_A_Hybrid_Video_Anomaly_Detection_Framework_via_Memory-Augmented_Flow_Reconstruction_ICCV_2021_paper

[14] A. Barbalau, R. T. Ionescu, M.-I. Georgescu, J. Dueholm, B. Ramachandra, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, “SSMTL++: Revisiting Self-Supervised Multi-Task Learning for Video Anomaly Detection,” Feb. 2023, arXiv:2207.08003 [cs]. [Online]. Available: http://arxiv.org/abs/2207.08003

[15] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 3, pp. 555–560, Mar. 2008.

[16] U. of Minnesota, "Unusual crowd activity dataset of university of minnesota," 2006. [Online]. Available: http://mha.cs.umn.edu/proj_events.shtml

[17] B. Ramachandra and M. J. Jones, "Street Scene: A new dataset and evaluation protocol for video anomaly detection," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2020, pp. 2558–2567, iSSN: 2642-9381.

[18] R. Rodrigues, N. Bhargava, R. Velmurugan, and S. Chaudhuri, "Multi-timescale Trajectory Prediction for Abnormal Human Activity Detection," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2020, pp. 2615–2623, iSSN: 2642-9381.

[19] A. Acsintoae, A. Florescu, M.-I. Georgescu, T. Mare, P. Sumedrea, R. T. Ionescu, F. S. Khan, and M. Shah, "UBnormal: New Benchmark for Supervised Open-Set Video Anomaly Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 143–20 153. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/html/Acsintoae_UBnormal_New_Benchmark_for_Supervised_Open-Set_Video_Anomaly_Detection_CVPR_2022_paper.html

[20] C. Cao, Y. Lu, P. Wang, and Y. Zhang, "A New Comprehensive Benchmark for Semi-Supervised Video Anomaly Detection and Anticipation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 392–20 401. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2023/html/Cao_A_New_Comprehensive_Benchmark_for_Semi-Supervised_Video_Anomaly_Detection_and_CVPR_2023_paper

[21] W. Sultani, C. Chen, and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6479–6488. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Sultani_Real-World_Anomaly_Detection_CVPR_2018_paper

[22] P. Wu, J. Liu, Y. Shi, Y. Sun, F. Shao, Z. Wu, and Z. Yang, "Not only Look, But Also Listen: Learning Multimodal Violence Detection Under Weak Supervision," in *European Conference on Computer Vision (ECCV)*, vol. 12375. Cham: Springer International Publishing, 2020, pp. 322–339, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-030-58577-8_20

[23] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A Large Video Database for Human Motion Recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[24] Qualcomm, "Moving Objects Dataset: Something-Something v. 2," 2018. [Online]. Available: https://developer.qualcomm.com/software/ai-datasets/something-something

[25] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The Kinetics Human Action Video Dataset," May 2017, arXiv:1705.06950 [cs]. [Online]. Available: http://arxiv.org/abs/1705.06950

[26] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, "A Short Note about Kinetics-600," Aug. 2018, arXiv:1808.01340 [cs]. [Online]. Available: http://arxiv.org/abs/1808.01340

[27] L. Smaira, J. Carreira, E. Noland, E. Clancy, A. Wu, and A. Zisserman, "A Short Note on the Kinetics-700-2020 Human Action Dataset," Oct. 2020, arXiv:2010.10864 [cs]. [Online]. Available: http://arxiv.org/abs/2010.10864

[28] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 1932–1939, iSSN: 1063-6919.

[29] J. Pers, V. Sulic, M. Kristan, M. Perse, K. Polanec, and S. Kovacic, "Histograms of optical flow for efficient representation of body motion," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1369–1376, Aug. 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865510001121

[30] R. V. H. M. Colque, C. Caetano, M. T. L. de Andrade, and W. R. Schwartz, "Histograms of Optical Flow Orientation and Magnitude and Entropy to Detect Anomalous Events in Videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 673–682, Mar.

2017, conference Name: IEEE Transactions on Circuits and Systems for Video Technology.

[31] B. Sabzalian, H. Marvi, and A. Ahmadyfard, "Deep and Sparse features For Anomaly Detection and Localization in video," in *2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, Mar. 2019, pp. 173–178, iSSN: 2049-3630.

[32] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94

[33] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier Detection with Kernel Density Functions," in *Machine Learning and Data Mining in Pattern Recognition*, ser. Lecture Notes in Computer Science, P. Perner, Ed. Berlin, Heidelberg: Springer, 2007, pp. 61–75.

[34] M. Glodek, M. Schels, and F. Schwenker, "Ensemble Gaussian mixture models for probability density estimation," *Computational Statistics*, vol. 27, pp. 127–138, Dec. 2013.

[35] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A Geometric Framework for Unsupervised Anomaly Detection," in *Applications of Data Mining in Computer Security*, ser. Advances in Information Security, D. BarbarÃ¡ and S. Jajodia, Eds. Boston, MA: Springer US, 2002, pp. 77–101. [Online]. Available: https://doi.org/10.1007/978-1-4615-0953-0_4

[36] G. Wang, Y. Wang, J. Qin, D. Zhang, X. Bao, and D. Huang, "Video Anomaly Detection byÂ Solving Decoupled Spatio-Temporal Jigsaw Puzzles," in *Computer Vision - ECCV 2022*, ser. Lecture Notes in Computer Science, S. Avidan, G. Brostow, M. CissÃ©, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 494–511.

[37] D. Wei, J. J. Lim, A. Zisserman, and W. T. Freeman, "Learning and Using the Arrow of Time," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8052–8060. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Wei_Learning_and_Using_CVPR_2018_paper.html

[38] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised Representation Learning by Predicting Image Rotations," Mar. 2018, arXiv:1803.07728 [cs]. [Online]. Available: http://arxiv.org/abs/1803.07728

[39] M.-I. Georgescu, A. Barbalau, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, "Anomaly Detection in Video via Self-Supervised and Multi-Task Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 742–12 752. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Georgescu_Anomaly_Detection_in_Video_via_Self-Supervised_and_Multi-Task_Learning_CVPR_2021_paper.html

[40] M. I. Georgescu, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, "A Background-Agnostic Framework With Adversarial Training for Abnormal Event Detection in Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4505–4523, Sep. 2022, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[41] S. Sun and X. Gong, "Hierarchical Semantic Contrast for Scene-Aware Video Anomaly Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 846–22 856. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2023/html/Sun_Hierarchical_Semantic_Contrast_for_Scene-Aware_Video_Anomaly_Detection_CVPR_2023_paper

[42] Z. Yang, P. Wu, J. Liu, and X. Liu, "Dynamic Local Aggregation Network with Adaptive Clusterer for Anomaly Detection," Jul. 2022, arXiv:2207.10948 [cs]. [Online]. Available: http://arxiv.org/abs/2207.10948

[43] G. Yu, S. Wang, Z. Cai, X. Liu, C. Xu, and C. Wu, "Deep Anomaly Discovery From Unlabeled Videos via Normality Advantage and Self-Paced Refinement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 987–13 998. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/html/Yu_Deep_Anomaly_Discovery_From_Unlabeled_Videos_via_Normality_Advantage_and_CVPR_2022_paper.html

[44] R. Cai, H. Zhang, W. Liu, S. Gao, and Z. Hao, "Appearance-Motion Memory Consistency Network for Video Anomaly Detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, pp. 938–946, May 2021, number: 2. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16177

[45] H. Lv, C. Chen, Z. Cui, C. Xu, Y. Li, and J. Yang, "Learning Normal Dynamics in

Videos with Meta Prototype Network," May 2021, arXiv:2104.06689 [cs]. [Online]. Available: http://arxiv.org/abs/2104.06689

[46] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, and Y. Qiao, "VideoMAE V2: Scaling Video Masked Autoencoders With Dual Masking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 549–14 560. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2023/html/Wang_VideoMAE_V2_Scaling_Video_Masked_Autoencoders_With_Dual_Masking_CVPR_2023_paper.html

A SAMPLES FROM PROPOSED DATASETS

A.1 HMDB-AD Examples

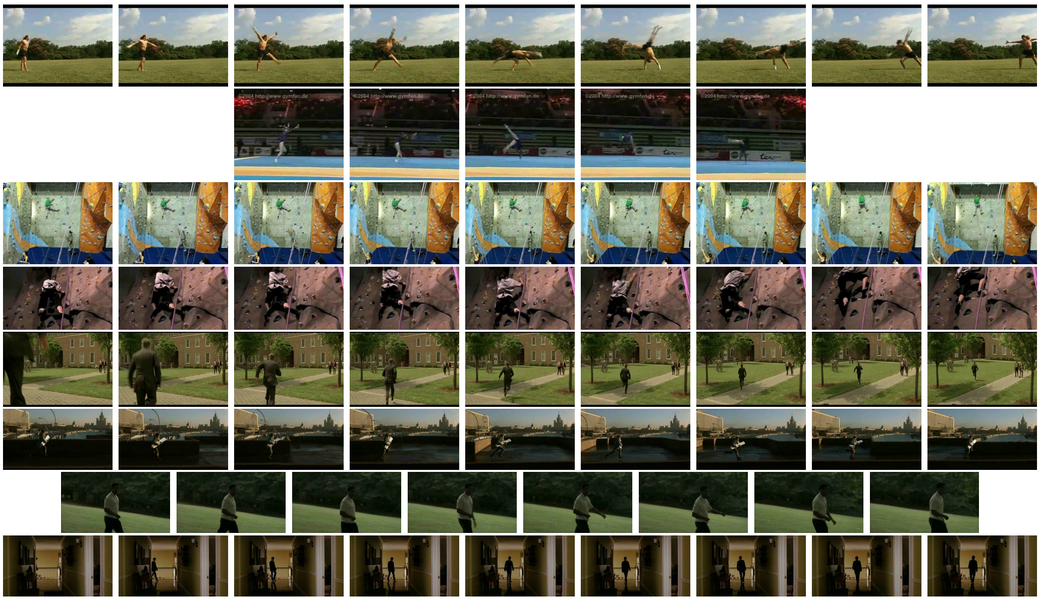


Figure 4: HMDB-AD examples: cartwheel (2), climb (2), run (2), walk (2).

A.2 HMDB-Violence Examples

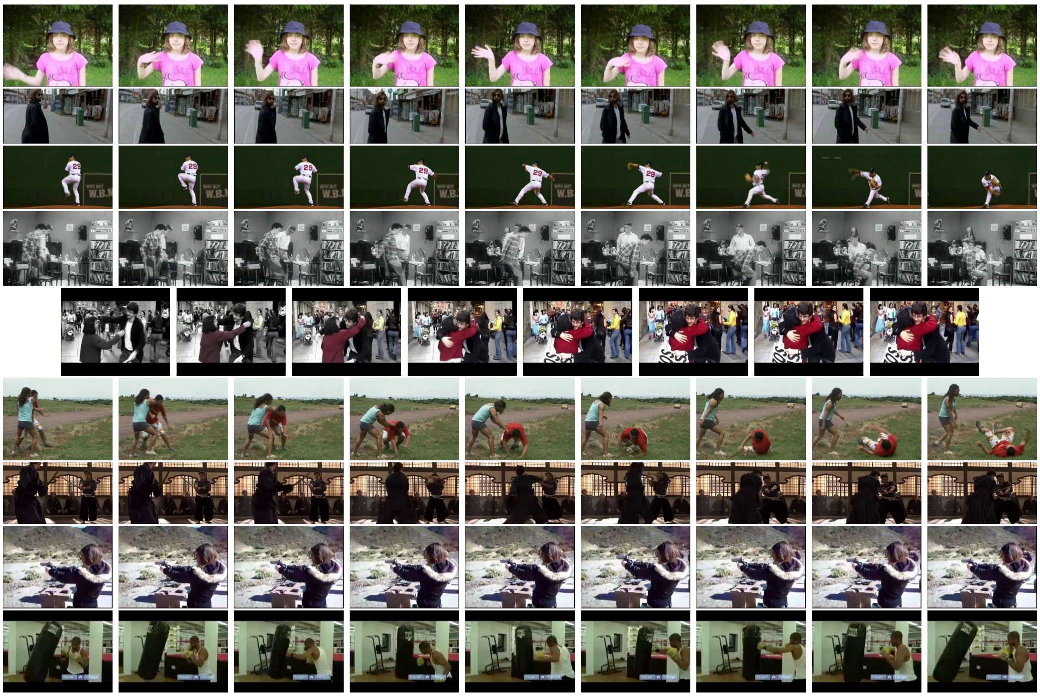


Figure 5: HMDB-Violence examples: wave, turn, throw, sit, hug, fall, sword, shoot, punch.

B VIDEOS FOR QUALITATIVE ANALYSES

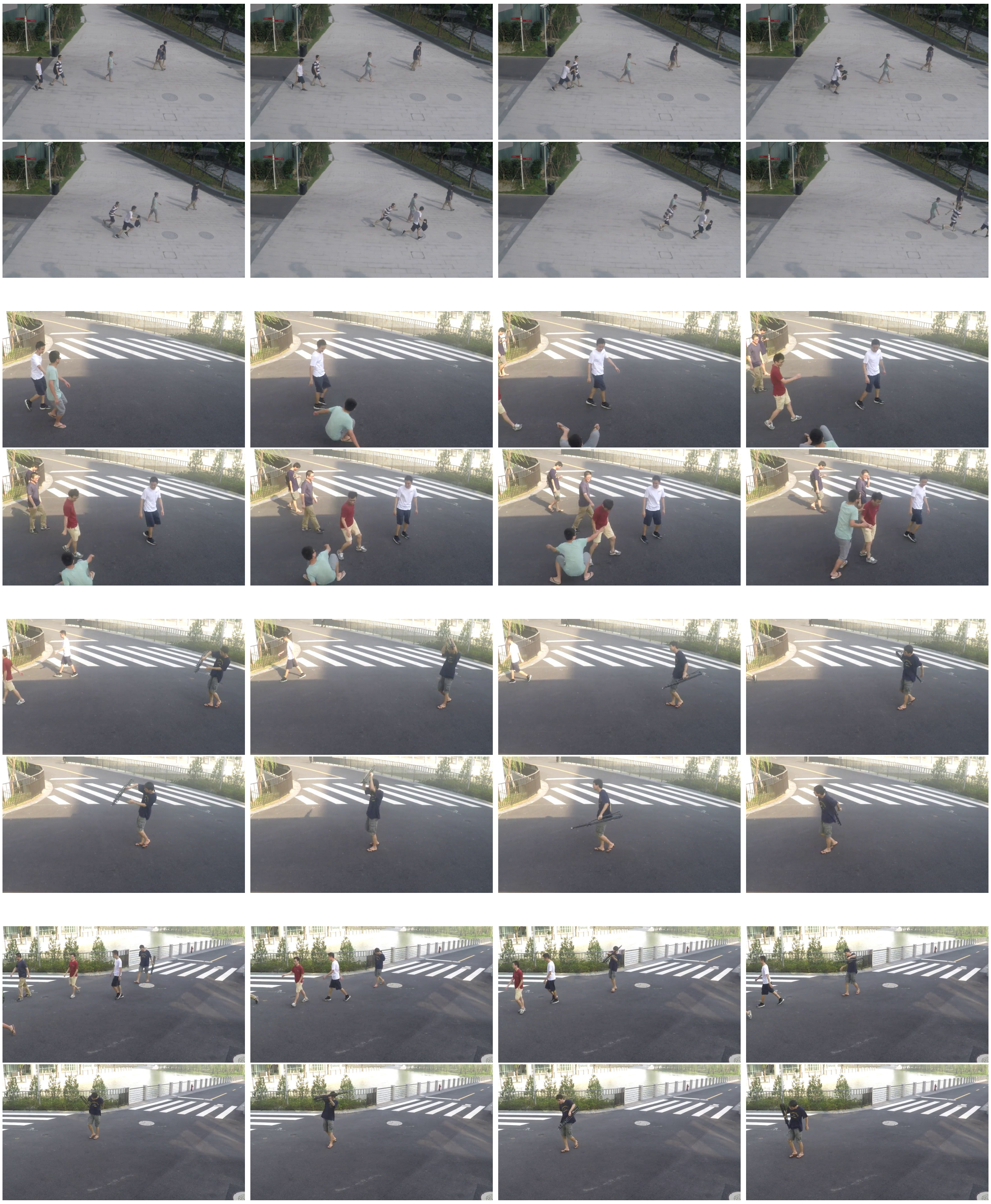


Figure 6: The anomalies from 01_0028, 03_0032, 03_0039, 07_0008 (top to bottom, respectively) videos from ShanghaiTech Campus dataset.

PixLabelCV - Labeling images for semantic segmentation fast, pixel-precise and offline

Dominik Schraml
Ilmenau University of Technology
SQB Ilmenau
Werner-von-Siemens Str. 9,
98693 Ilmenau, Germany
dominik.schraml@tu-ilmenau.de

Konstantin Trambickii
SQB Ilmenau
Werner-von-Siemens Str. 9,
98693 Ilmenau, Germany
konstantin.trambitckii@sqb-ilmenau.de

Gunther Notni
Ilmenau University of Technology
Ehrenbergstrasse 29,
98693 Ilmenau, Germany
gunther.notni@tu-ilmenau.de

ABSTRACT

Image annotation, also called labeling is a necessary task for any supervised learning approach to obtain ground truth data for model training. This article offers a comprehensive survey of contemporary image annotation tools, grouping freely accessible ones based on their service range, speed, and data privacy assurances.

In our exploration for tools capable of executing pixel-precise semantic labeling, we identified a shortage of swift, free image annotation tools that don't require users to upload their data to third-party servers. Therefore, we introduce "PixLabelCV" - a lightweight, fast, offline, and standalone annotation tool primarily developed to aid human annotators in achieving pixel-perfect labels promptly. Uniquely crafted to be freely available (open source) and non-server-based, it ensures enhanced privacy and efficiency. Hence, it is aimed to serve as an ideal tool to facilitate labeling data for smaller labs and businesses.

At its core, PixLabelCV fuses conventional labeling techniques such as delineating objects with rectangles or polygons with multiple computer vision algorithms. Spanning basic thresholding in RGB or HSV color space to more intricate procedures like flood fill or watershed the tool instantaneously computes and exhibits the resulting segmentations. Annotators can swiftly add these segments to a class label or refine them by adjusting parameters or markers before a quick repetition. To further augment the user experience, additional functionalities like morphological closing are incorporated, facilitating an intuitive labeling process. Another standout feature is its ability to uniquely assign pixels to singular classes, eliminating any potential overlap-induced ambiguities.

Keywords

Semantic Segmentation, Image Annotation Tools, Pixel-Precise Labeling, Computer Vision Algorithms, Annotation Software

1 INTRODUCTION

With the increasing use and application of deep neural networks for various tasks, labeled[1] training data for supervised learning is also increasingly needed. In this context, for many tasks such as autonomous driving or medical image analysis, it is not sufficient to identify only the objects present in the image, but it is also necessary to determine the exact position and size of the objects in terms of pixels. This process of assigning every pixel of an image to an object class is called semantic segmentation. In practice, this usually means that the resulting label is an equal-sized image in which each pixel has the value of the resulting class. However the cost of annotating every pixel to generate training data for a semantic segmentation network is immense [9]. Furthermore, studies by Zlateski et al. [17] on the quality of labels for semantic segmentation of urban scenes have shown that a larger coarsely annotated dataset can yield the same performance as a smaller finely, meaning pixel-precise, annotated dataset. Conversely, this means that a smaller number of images with precise semantic class labels is sufficient to achieve the same performance as is possible with a larger number of coarsely annotated 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.

---

[1] The terms "labeling" and "annotation" are used interchangeably in the context of this work. In general, the term "label" is most often used to describe any output from a machine learning model, while "annotation" refers to the type of label, such as an image-sized pixel mask.

https://www.doi.org/10.24132/CSRN.3401.6

A particular burden for assigning a semantic class label to each pixel is the fact that object boundaries can be complex and therefore difficult to accurately annotate [14]. More precisely Zlateski et al. [17] deduce from the time annotators took during their experiment that it is very hard for humans to draw pixel-precise labels, as it depends on their vision, dexterity, etc., and will vary greatly among different individuals.

Our goal is to reduce this effort by providing an image annotation tool for semantic segmentation that is pixel-perfect yet fast and easy to use, after initial acclimatization.

2 IMAGE ANNOTATION PROGRAMS

2.1 Evaluation Criteria

We base our evaluation of image annotation tools on several critical criteria, essential for optimizing both the annotation process and the quality of the resulting data.

The first criterion is the **range of functions** offered by the tool. This encompasses the variety of shapes that can be annotated and extends to the inclusion of sophisticated tools, such as integrated AI models and advanced computer vision algorithms.

The most important evaluation factors are **speed** and **offline usability**. The latter includes setup and installation and goes hand in hand with security, as there is no need to upload your confidential data to third-party servers, which enables faster processing times and contributes to data security.

Label precision forms the cornerstone of effective semantic segmentation. High precision, particularly pixel-perfect mask creation over simpler coordinate-based shapes or bounding boxes, directly impacts the quality of machine learning models trained with annotated data. Precise pixel masks circumvent issues such as depth ordering in overlapping objects, ensuring each pixel is accurately classified without ambiguity.

While **ease of use** significantly contributes to a tool's appeal, its subjective nature makes it challenging to quantify. In this context, it is also important to note the tools excluded from our evaluation. Commercially licensed platforms and machine learning annotation tools, such as Hasty.ai, Labelbox, Prodigy, RectLabel and Supervisely were not considered due to their exclusive commercial licensing. Additionally, tools that were not accessible, faced technical issues, or failed to meet minimum usability requirements, such as GTCreator [4], RhobanTagger, DeepLabel, semantic-image-label-tool Scalabel, labelImg, LabelStudio, Labelflow and MedTagger were also excluded. Furthermore, tools demanding extensive user data rights, such as V7labs and Diffgram were omitted due to our data protection criteria.

2.2 Comparative Analysis of Image Annotation Tools

In our systematic evaluation of image annotation tools, as detailed in Table 1, we sought tools that balance functionality, user experience, and data privacy. The analysis showed a preference for server-based, browser-operated tools, predominantly outputting in JSON, XML, and CSV formats. These formats are less suited for semantic segmentation where precision is paramount, as they typically save only the coordinates of drawn shapes, potentially leading to inaccuracies and overlapping shapes.

Among the tools listed, only CVAT, label-studio, PixelAnnotationTool, Semantic Segmentation Editor and S3A offer the capability to save labeling results as pixel masks and are open source. While specialized tools like PixelAnnotationTool (utilizing the watershed algorithm for region creation) and S3A (semi-automatic labeling of printed circuit boards) target specific annotation tasks, they are limited in broader usability due to their restricted toolsets. Label Studio provides basic brush tools for pixel-precise labeling beyond simple shape annotations, while Semantic Segmentation Editor introduces a more advanced "magic tool" for efficient segmentation. However, CVAT stands out as the most feature-rich tool, equipped with a comprehensive suite of automation tools including advanced computer vision techniques like smart scissors and automatic annotation capabilities via the TensorFlow Object Detection API. It supports a wide range of export formats, making it a robust tool for complex labeling projects.

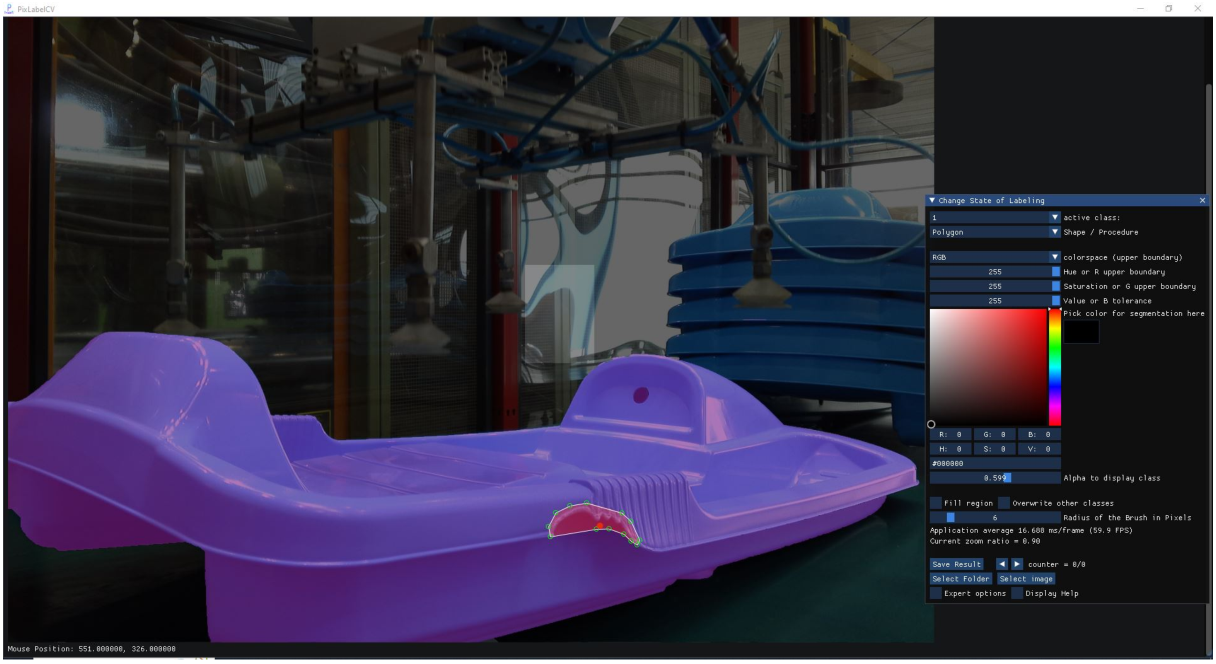
However, CVAT's extensive setup requirements for projects, annotators and "jobs", as well as the need to set up the server for local use, are not neglectable hurdles. Moreover, CVAT, like its counterparts, does not solve the depth order conflict inherent in overlapping object annotations.

3 PIXLABELCV

3.1 Structure of the Program

Recognizing a gap in the availability of lightweight, swift, and offline image annotation tools for semantic segmentation, that allows half-automated pixel-precise annotation, we developed our own solution.

The basic principle of using the program differs from that of the other software tested. By allowing the application of computer vision algorithms a segmentation in the chosen region of interest (ROI) is performed in the time of milliseconds, leading to a preliminary class mask, which is presented to the annotator. Provided the mask is precise enough to meet the annotator's needs, it can be swiftly added to the overall image mask. This iterative process continues until the whole image is labeled.



1 active class:
Polygon Shape / Procedure
RGB colorspace (upper boundary)
255 Hue or R upper boundary
255 Saturation or G upper boundary
255 Value or B tolerance
Pick color for segmentation here
R: 0 G: 0 B: 0
H: 0 S: 0 V: 0
#000000
0.599 Alpha to display class
☐ Fill region ☐ Overwrite other classes
6 Radius of the Brush in Pixels
Application average 16.688 ms/frame (59.9 FPS)
Current zoom ratio = 0.90
counter = 0/0
☐ Expert options ☐ Display Help
Mouse Position: 551.000000, 326.000000

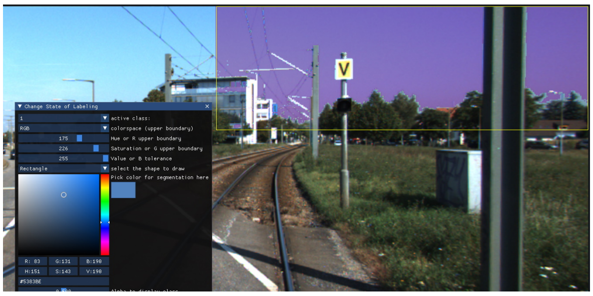
Figure 1: PixLableCV interface, annotations of the sled (purple) using GrabCut or Watershed algorithm and the defect (red) using polygon annotation.

The PixLabelCV software[2] facilitates the annotation process by enabling human annotators to demarcate regions within images. These regions can be assigned to various classes using standard annotation shapes, including rectangles, circles, and polygons. Additionally, individual pixels in the image can be marked to belong to a specific class. After delineating a ROI, users have the option to apply specific computer vision algorithms. For instance, they can employ thresholding in either HSV or RGB color space, setting both upper and lower limits for each color channel. Alternatively, the floodfill algorithm can be initiated within the ROI, originating from the current cursor location. Furthermore, the watershed algorithm can be utilized to segment the image based on the marked points. Details on the technical implementation can be found in Appendix A.

3.2 Usage of the Program

After starting the program, the user can either load an individual image or select a directory containing multiple images. When a directory is chosen, the program automatically recognizes images in BMP, JPG, PNG, or TIFF formats and starts with the first image.

Class Selection: Choose the label class using the combobox or by pressing the respective class number key.



1 active class:
RGB colorspace (upper boundary)
175 Hue or R upper boundary
226 Saturation or G upper boundary
255 Value or B tolerance
Rectangle select the shape to draw
Pick color for segmentation here
R: 83 G:131 B:190
H:151 S:143 V:190
#5383BE

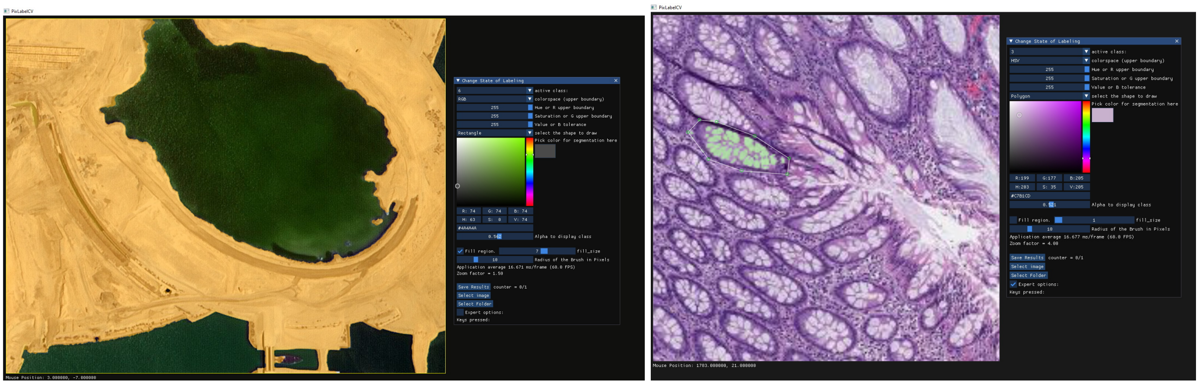
Figure 2: Applying a Threshold to segment the region "sky".

For classes greater than 10, hold the *Alt* key while pressing the class number key. The selected class can be changed anytime even after the segmentation to add the resulting region to another class or remove it.

Drawing Shapes and Markers: For basic annotations, enclose the object within a rectangle. More intricate shapes can be outlined using polygons, which can be further refined as shown in Figure 1. Round structures are best represented using the circle tool: initiate by positioning it on an edge contour, adjust segmentation parameters or apply flood fill, then fine-tune by dragging the contour and modifying the radius. A pixel brush, whose radius is adjustable, is ideal for adding or subtracting pixels from a specific class region. Additionally, marker points can be set for each class, which serve as seed for the application of the watershed algorithm in segmentation tasks. The same applies to the GrabCut implementation.

---

[2] PixLabelCV software will be available at `https://sqb-ilmenau.de/pixlabelcv`. The source code can be found on GitHub: `https://github.com/dominiks-dev/PixLabelCV`



6 ▼ active class:
RGB ▼ colorspace (upper boundary)
255 Hue or R upper boundary
255 Saturation or G upper boundary
255 Value or B tolerance
Rectangle ▼ select the shape to draw
Pick color for segmentation here
R: 74 G: 74 B: 74
H: 63 S: 0 V: 74
#4A4A4A
0.562 Alpha to display class
☑ Fill region. 7 fill_size
10 Radius of the Brush in Pixels
Application average 16.671 ms/frame (60.0 FPS)
Zoom factor = 1.50
counter = 0/1
☐ Expert options:
Keys pressed:
Mouse Position: 3.000000, -7.000000

3 ▼ active class:
HSV ▼ colorspace (upper boundary)
255 Hue or R upper boundary
255 Saturation or G upper boundary
255 Value or B tolerance
Polygon ▼ select the shape to draw
Pick color for segmentation here
R:199 G:177 B:205
H:203 S: 35 V:205
#C7B1CD
0.521 Alpha to display class
☐ Fill region. 1 fill_size
10 Radius of the Brush in Pixels
Application average 16.677 ms/frame (60.0 FPS)
Zoom factor = 4.00
counter = 0/1
☑ Expert options:
Keys pressed:
Mouse Position: 1783.000000, 21.000000

(a) Segmenting large ground area (highlighted yellow) accurately from the water area (landcover dataset).

(b) Pixel-precise cell segmentation using thresholding in a polygon (Histopathology dataset).

Figure 3: Examples of efficient PixLabelCV usage.

Segmentation Process: During the segmentation process, algorithms like thresholding, flood fill, watershed [12] or GrabCut [13] are applied using previously set markers or shapes. For thresholding, users can activate a color picker, which auto-selects the object's pixel color for segmentation. This selection can be refined by adjusting the upper and lower boundary HSV values, as illustrated in Figure 2. Segmentation is initiated with mouse actions or through an optimized keyboard setup.[3] Once segmentation is satisfactory, it can be added to the chosen class. If adjustments are needed, parameters can be tweaked and segmentation can be performed again within a few milliseconds. Flood fill segmentation starts at the cursor's pixel and users can choose if the current segmented pixel mask should replace pixels that have already been assigned to other classes. The same applies to the GrabCut algorithm for separating the current class from the background (see Figure 4).

Modifications: For any modifications, unwanted regions or individual pixels can be removed by either allocating them to class 0 or background. This class is designed specifically for erasing unnecessary elements, with the brush tool also available for this task. It is also possible to overwrite all pixels that already belong to another class by explicitly setting this option. The pixel brush tool is versatile, enabling both the addition and erasure of pixels from a given class region. Lastly, the software facilitates precise labeling with its zoom functionality, swiftly magnifying up to 4x when needed.

Efficient Annotation Strategy: To optimize the annotation workflow, a highly effective and swift technique involves beginning with the placement of watershed marker points across the entire image for every class. This initial segmentation can be computed in a few hundred milliseconds, even for 4k images, producing a relatively accurate pixel mask. If necessary, adding a few more markers can refine this result before further enhancing the mask's accuracy using other functions such as floodfill or the pixel-brush. This approach not only speeds up the process but also leverages the software's capabilities for achieving precise pixel masks with minimal manual intervention.

Moreover, it's advisable to initially focus on segmenting objects or classes that are enclosed within others. For shared boundaries between distinct classes, start the segmentation with simpler-to-demarcate regions, employing the flood fill algorithm for efficiency. Afterward, encompass the entire region within a bounding box and designate it to the secondary class. Due to the program's design, which by default prevents pixels already assigned to one class from being overridden by another, any pixels not segmented in the initial step will automatically be attributed to the secondary class. This strategy streamlines the annotation process by eliminating the need to meticulously draw an additional polygon, with these computations being rapidly performed in just milliseconds.

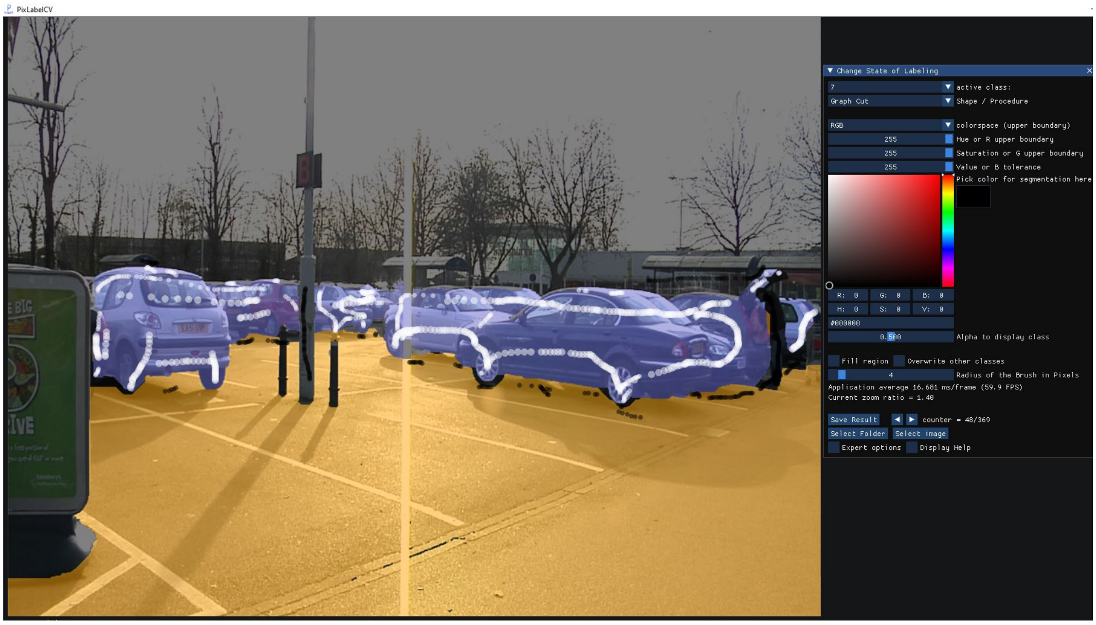
Saving and Advancing: Upon completing the annotation, the output is a pixel mask where each pixel corresponds to its class value. This mask can be stored as an 8-bit depth PNG file or if specifically chosen each class mask can be saved in a separate file so that multiple labels per image are possible. If a directory path was initially provided containing multiple images, a single button press both saves the current annotation and loads the subsequent image for labeling.

3.3 Comparison to Segment Anything Model (SAM)

The Segment-Anything Model (SAM) [10] is notable for its ability to segment unknown objects within images. Despite its extensive capabilities, the authors

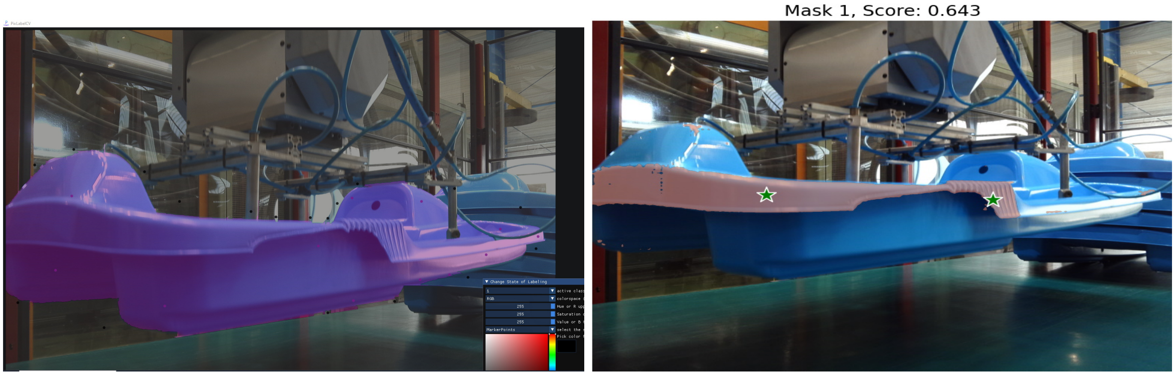
---

[3] All operations mentioned are assigned to keys conveniently located around the 'ASDF' keys, optimizing for dual-hand usage thereby enhancing efficiency in labeling.



active class: 7
Shape / Procedure: Graph Cut
colorspace (upper boundary): RGB
Hue or R upper boundary: 255
Saturation or G upper boundary: 255
Value or B tolerance: 255
Pick color for segmentation here
R: 0 G: 0 B: 0
H: 0 S: 0 V: 0
#000000
Alpha to display class: 0.500
Radius of the Brush in Pixels: 4
Application average 16.681 ms/frame (59.9 FPS)
Current zoom ratio = 1.40
counter = 48/369

Figure 4: Graph Cuts - white (class) and black (background) - to segment cars.



(a) Results of our tool (PixLabelCV) setting marker points and applying watershed algorithm.

(b) SAM segmentation result on sled using only few point inputs and no bounding rectangle.

Figure 5: Comparing SAM to watershed algorithm on custom data.

of SAM have acknowledged specific shortcomings, including occasionally missing finer structures in images, creating minor hallucinated disconnected components, and not consistently delineating sharp boundaries. This last limitation is particularly evident when juxtaposed against techniques that use a ’zoom-in’ approach. Moreover, while SAM’s design emphasizes versatility for a broad range of applications, this generality can sometimes result in precision trade-offs.

While SAM demonstrates proficiency when provided with a clear bounding rectangle around an object, as exemplified by its performance on public datasets like KITTI, it struggles when domain-specific expertise is crucial for pixel-accurate segmentation. We encountered this limitation with medical images and our proprietary dataset, especially for objects lacking a clear bounding rectangle, such as the sled in our injection-molded dataset (see Figure 5). Additionally, while the decoder part of SAM operates quickly, encoding an image can take many seconds to multiple minutes, depending on the image size and is restricted to a maximum input image size of 1024x1024 pixels. In terms of operational efficiency, SAM proved to be slower with larger images and consumed substantial amounts of memory compared to our tool. When provided with ample input points, SAM’s performance becomes comparable to our tool.

In conclusion, while SAM offers a versatile approach to image segmentation, our labeling program adeptly mitigates some of its shortcomings, particularly in refining masks that are either too expansive or too constricted. Although SAM’s one-shot characteristic presents clear advantages for well-known datasets such as urban scenes where the model has been extensively trained, it struggles with large images and user-specific datasets from sectors like industrial manufacturing.

4 USER STUDY

4.1 Setup of the Study

To evaluate our labeling program for semantic segmentation, we conducted a user study with five participants. The small number was chosen for feasibility, focusing more on gaining qualitative insights into the usability and effectiveness of our program rather than conducting a quantitative hypothesis test. The study was carried out as follows. First, the students were given a brief introduction to each tool to ensure they had a basic understanding of how to use them. Then the task was to annotate one to three images from four different categories and datasets (KITTI, landcover, MHIST and our custom dataset of an injection molded part[4]). After performing the task each participant filled out a questionnaire in open-ended design which contained the following questions about the two tools:

1. Which tool did you find faster for annotation images? Why? Options: CVAT, PixLabelCV, no difference
2. Which tools did you use the most often?
3. How much faster is it to use the computer vision tools to annotate instead of only annotating with shapes?
4. Which tool do you believe allowed you to annotate images more accurately? Options: CVAT, PixLabelCV, no difference
5. Rate your overall satisfaction with each tool: Scale from 1 (very unsatisfied) to 5 (very satisfied)

4.2 User Study Evaluation and Findings

Analysis of the user study results indicated that PixLabelCV, with an average satisfaction score of 3.4 (SD 0.49), is slightly behind CVAT's 3.6 (SD 0.49) out of 5 in terms of usability and overall user satisfaction. Given that all ratings were either 3 or 4 out of 5, we consider PixLabelCV to be roughly on par with one of the leading state-of-the-art labeling tools.

Regarding the speed of annotation, half of the participants found PixLabelCV to be faster, while the other half favored CVAT, with one participant reporting no significant difference between the two tools. It should be noted that this assessment was based on the time taken to label a series of images from different domains (with only one per domain). For both tools, the combination of advanced computer vision techniques (*watershed* for PixLabelCV and *"smart scissors"* for CVAT) with shape tools, particularly the polygon shape, was credited for the perceived increase in labeling speed.

The most notable distinction between the tools emerged in their accuracy. A vast majority - 80% of participants - identified PixLabelCV as enabling more precise labeling compared to CVAT. This underlines PixLabelCV's capability to support detailed and accurate annotations, essential for high-quality semantic segmentation.

Participants identified key features and advantages of each program. PixLabelCV's incorporation of the *watershed algorithm* was particularly valued for its effectiveness in images with distinct color differences and blurriness, such as medical images. This feature positions PixLabelCV as particularly advantageous for projects requiring complex segmentation tasks. Conversely, the *"smart scissors"* feature in CVAT was acknowledged for its proficiency in swiftly generating accurate polygon labels, illustrating CVAT's efficiency in more conventional annotation scenarios.

4.3 Analysis

While the relatively small number of participants may be perceived as a limitation of our user study, it is important to note that our focus was on evaluating the practical utility and relevance of PixLabelCV rather than conducting a quantitative hypothesis test. The study reveals that PixLabelCV is on par with CVAT in terms of usability and excels in enabling more precise annotations, affirming its capability as a powerful tool for complex segmentation tasks and as a viable alternative for unique and custom data applications.

Integration and Efficiency: PixLabelCV distinguishes itself through the seamless integration of diverse tools to achieve enhanced boundary precision. The process typically begins with an initial, coarse delineation of boundaries using the watershed method. This is followed by further refinements via the pixel-brush and other tools for precise, custom-tailored results. This integrated toolset proves especially advantageous for repetitive tasks, such as quality control in industrial settings, where speed and accuracy are paramount. Notably, PixLabelCV demonstrates an inherent advantage in speed over AI-model-dependent approaches like SAM, particularly for sets of similar scenes.

Offline Usability and Performance: PixLabelCV is engineered for flexibility, leveraging GPU capabilities for accelerated algorithm performance when available, with fallback to CPU processing. This architecture supports mobile use on laptops, eliminating the need for

---

[4] The dataset contains images of injection molded sleds for quality assessment. There are images of good products and ones of sleds that show one of the various defects that occasionally occur during injection molding, such as short shot, flash and color streak.

server connectivity and data transfer, thereby streamlining the labeling process.

Handling Large Images: The tool's capability to efficiently process and segment large images, such as those found in the Landcover.ai dataset, without significant delays, highlights its suitability for extensive geographic or environmental datasets.

Deployment and Precision: PixLabelCV's minimalistic design and straightforward deployment process facilitate quick setup and immediate use. Its precision, particularly evident in medical imaging tasks, leverages subtle color and brightness variations for meticulous segmentation, further enhanced by features like floodfill for better edge distinction.

Considerations: While PixLabelCV excels in many areas, it encounters limitations in generic scenes where pre-trained AI models are prevalent. In such scenarios, tools capable of pre-labeling with task-specific AI may offer more efficiency. Additionally, the tool's rich feature set and reliance on image processing principles may pose a learning challenge for novices, contrasting with the more intuitive nature of platforms like CVAT or the straightforwardness of SAM for basic tasks.

Domain Applicability: PixLabelCV's effectiveness varies by application domain. While it offers considerable advantages for medical imaging and specialized custom datasets, its utility may not extend as effectively to areas well-served by existing AI models, such as street scenes.

In conclusion, PixLabelCV represents a significant advancement in semantic image annotation, particularly for users seeking high-speed, precision-driven tools for specialized applications. However, potential users must balance these benefits against the need for a foundational understanding of image processing and the tool's specific capabilities.

5 CONCLUSION

Image annotation remains a pivotal component in the realm of supervised machine learning and deep learning paradigms. Our comprehensive examination of current image annotation tools revealed a significant gap in the market: the absence of swift, freely available annotation tools that also prioritize data privacy. In response, we introduced "PixLabelCV," a streamlined yet powerful offline tool crafted to enable annotators to quickly generate pixel-perfect labels. By harnessing core computer vision techniques such as the watershed and flood fill algorithms and integrating these with basic shape tools, PixLabelCV significantly enhances precision, facilitating the rapid development of high-quality pixel masks.

The user study conducted to evaluate PixLabelCV against established tools like CVAT highlighted its strengths, particularly in precision and specialized task performance. PixLabelCV demonstrated a competitive edge, standing on par with CVAT in terms of usability while outperforming it in accuracy as acknowledged by the majority of participants. This underscores PixLabelCV's potential as a potent tool for complex segmentation tasks, offering a robust alternative for unique and custom data and use cases.

Looking forward, we aim to amplify PixLabelCV's capabilities by integrating the Segment-Anything Model (SAM) through the ONNX Runtime. The fusion of SAM with PixLabelCV's existing computer vision methodologies promises to enhance initial segmentation efforts, paving the way for a more simple usability for novices.

In summation, PixLabelCV aims to catalyze both the precision and speed of the annotation process, thereby optimizing the generation of premium-quality labeled datasets.

6 REFERENCES

[1] Alper Aksac, Douglas J. Demetrick, Tansel Ozyer, and Reda Alhajj. Brecahad: a dataset for breast cancer histopathological annotation and diagnosis. *BMC research notes*, 12(1):82, 2019. 8

[2] B Albertina, M Watson, C Holback, R Jarosz, S Kirk, Y Lee, and J Lemmerman. Radiology data from the cancer genome atlas lung adenocarcinoma [tcga-luad] collection. *The Cancer Imaging Archive*, 10:K9, 2016. 8

[3] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)*, 2018. 8

[4] Jorge Bernal, Aymeric Histace, Marc Masana, Quentin Angermann, Cristina Sánchez-Montes, Cristina Rodríguez de Miguel, Maroua Hammami, Ana García-Rodríguez, Henry Córdova, Olivier Romain, Gloria Fernández-Esparrach, Xavier Dray, and F. Javier Sánchez. Gtcreator: a flexible annotation tool for image-based datasets. *International Journal of Computer Assisted Radiology and Surgery*, 14(2):191–201, 2019. 2

[5] Adrian Boguszewski, Dominik Batorski, Natalia Ziemba-Jankowska, Tomasz Dziedzic, and Anna Zambrzycka. Landcover.ai: Dataset for automatic mapping of buildings, woodlands, water and roads from aerial imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1102–1110, June 2021. 8

[6] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation

and recognition using structure from motion point clouds. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, 2008, Proceedings, Part I 10*, pages 44–57. Springer, 2008. 8

[7] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, et al. The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26:1045–1057, 2013. 8

[8] Omar Cornut. Dear imgui. `https://github.com/ocornut/imgui`, Accessed on: January 2023. 8

[9] Tejaswi Kasarla, Gattigorla Nagendar, Guruprasad M Hegde, Vineeth Balasubramanian, and CV Jawahar. Region-based active learning for efficient labeling in semantic segmentation. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1109–1117. IEEE, 2019. 1

[10] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 4

[11] Thomas Köhler, Attila Budai, Martin F Kraus, Jan Odstrčilik, Georg Michelson, and Joachim Hornegger. Automatic no-reference quality assessment for retinal fundus images using vessel segmentation. In *Proceedings of the 26th IEEE international symposium on computer-based medical systems*, pages 95–100. IEEE, 2013. 8

[12] Fernand Meyer. Color image segmentation. In *1992 international conference on image processing and its applications*, pages 303–306. IET, 1992. 4

[13] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004. 4

[14] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77:157–173, 2008. 2

[15] Amber L Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram Van Ginneken, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv:1902.09063*, 2019. 8

[16] Jerry Wei, Arief Suriawinata, Bing Ren, Xiaoying Liu, Mikhail Lisovsky, Louis Vaickus, Charles Brown, Michael Baker, Naofumi Tomita, Lorenzo Torresani, et al. A petri dish for histopathology image analysis. In *19th International Conference on Artificial Intelligence in Medicine, Proceedings*, pages 11–24. Springer, 2021. 8

[17] Aleksandar Zlateski, Ronnachai Jaroensri, Prafull Sharma, and Frédo Durand. On the importance of label quality for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1479–1487, 2018. 1, 2

A PIXLABELCV - TECHNICAL IMPLEMENTATION

Our program, named "PixLabelCV", operates as a Direct3D11 graphics application. It integrates DearImGui [8] to manage graphics, user input, and the display of GUI elements. Image processing is primarily conducted using the Open Source Computer Vision Library (OpenCV). Once processed, the resultant image is transferred to the Direct3D graphics buffer for rendering. All other operations, including filesystem access and basic computations, are executed using plain C++ in compliance with the ISO C++17 standard. The program is compiled into an executable file, which needs only to be copied together with a few dynamically linked libraries and can then be started directly without installation required.

B DATASETS USED

In addition to other and customer-specific data sets with injection molded objects, the following data sets were used to test the segmentation programs. Our thanks go to the providers at this point.

- BrainTumour from Medical Segmentation Decathlon [15]
- CamVid (Cambridge-Driving Labeled Video Database) dataset [6] obtained from kaggle
- CT Medical images from cancer imaging archive [2] [7] obtained from kaggle
- HRF (High-Resolution Fundus) [11]
- KITTI dataset [3]
- Landcover.ai V1 [5]
- MHIST: A Minimalist Histopathology Image Analysis Dataset [16]
- BreCaHAD [1]

C TABLE OF LABELING PROGRAMS

Name	**Type of tool**	**Shapes for annotation**	**Export Formats**	**Pixel Mask**	**License**	**Source code**
Coco-annotator	server, docker image	rectangle, circle, polygon, pixel brush, magic wand,	JSON	no	MIT Licence	yes
Colabler	offline installer	only brush for pixel wise labeling	JSON, XML, PNG	yes, separate 32bit Png per class	free for personal use	no
CVAT	server, can be run locally, docker images	rectangle, polygon, ellipse, pixel-brush, AI assistant, Scissors using CV	JSON, XML, PNG and more	yes, 16 bit png and 24 color png	MIT license	yes
DarkLabel	offline installer	rectangle	XML, TXT, CSV	no	free for non-commerical use	no
imglab	server, free online tool available	rectangle, circle, polygon, point	JSON, XML	no	MIT license	yes
labelme	offline, python package	rectangle, polygon, circle, line, point	JSON	JSON convert to PNG	GPL-3.0	yes
label-studio	server, can be run locally via python script	rectangle, polygon, pixel-brush (alternatively)	JSON, CSV, COCO, PNG	yes, separate .png file (8bit) per class	Apache-2.0	yes
Make-Sense	server, can be run locally, docker image	rectangle, polygon, line, point	JSON, CSV, TXT, XML	no	GPL-3.0	yes
MyVision	online, browser based	rectangle, polygon	JSON, CSV, TXT, XML	no	GPL-3.0 license	yes
PixelAnnotationTool	offline, executable .exe	pixel-brush (for watershed)	Binary PNG	yes, png with color values	LGPL-3.0	yes
Ratsnake	offline, executable .jar	polygon, gridbased-"brush"	JPG	yes, separate .jpg file (8bit) per class	sourcecode	no
Remo	server running on local machine, access via browser	rectangle, polygon	JSON, CSV	no	CC BY-ND 4.0	yes
Roboflow	machine learning platform with annotation tool, browser-based	rectangle, polygon, "smart polygon", ai assistant	JSON	no	commercial, free for academic and personal use	no
Semantic Segmentation Editor	server, can be run locally, docker image	rectangle, polygon, magic tool	PNG, JSON	yes, 32 bit color png	MIT license	yes
Semi-Supervised Semantic Annotator (S3A)	python package (py qt)	automated, rectangle, polygon, ellipse, point	CSV, PNG, cusom zip file	yes, 16 bit png	free to use (except Qt licence)	yes
Superannotate	desktop installer and online, browser based	rectangle, ellipse, polygon, polyline, cuboid	JSON, PNG	yes, 32 bit color png	free for startups and academic use	no
Universal Data Tool	server, can be installed locally and online tool	rectangle, polygon, point	JSON, CSV	no, coverter does not work	MIT Licence	yes
VGG Image Annotator (VIA)	online, browser based	rectangle, ellipse, polygon, line	JSON, CSV	no	BSD 2-Clause	yes
VoTTÂ	server, can be hosted locally	rectangle, polygons	JSON, CSV, Azure	no	MIT license	yes

Table 1: Overview of labeling tools

Show Me the GIFference! Using data-GIFs as Educational Tools

Lorenzo Amabili
TU Wien
lorenzo.amabili@gmail.com

M. Eduard Gröller
TU Wien and VRVis
groeller@cg.tuwien.ac.at

Renata G. Raidou
TU Wien
rraidou@cg.tuwien.ac.at

ABSTRACT

We investigate the use of data-GIFs, i.e., graphics interchange format files containing short animations, to engage visualization viewers in learning about data visualization design pitfalls. A large number of data visualizations—among which, also several with bad data designs—are generated every day to convey information to lay audiences. To support non-expert viewers in recognizing common visualization design mistakes, we propose *data-GIFs*. Data-GIFs are short educational animations played in automatic repetition with a single core message on how the design flaws of a given visualization can be identified. After defining what bad data visualization design entails, we inform the design requirements for the data-GIFs. We, subsequently, design four variants: two data-GIFs, which use respectively interchangeability and smooth transitions, a static variant with juxtaposition, and a data-video approach with audio. In a controlled user study with 48 participants, we compare the four variants. We demonstrate that interchangeability and smooth transitions effectively support viewers in assessing *why* elements characterizing bad data visualizations are indeed bad. Yet, smooth transitions are more engaging, and data-videos are more efficient for the identification of differences between bad and good data visualization designs.

Keywords
Data-GIF, visualization education, visualization design pitfalls, design study, information visualization.

1 INTRODUCTION

Among the many data visualizations produced every day, we often encounter cases with unsuitable encodings and visuals [20, 21]. These visualizations may confuse non-expert viewers and make data interpretation difficult, or may even (un)intentionally communicate misleading information. Understanding what makes data visualizations bad, i.e., confusing or less effective in communicating data, and raising awareness about the existence of bad visualization designs is a core research topic of our field [8, 17, 30]. In this work, we investigate *how to engage visualization viewers in effectively identifying design components that distinguish bad (i.e., poorly designed) from good (i.e., well-designed) visualizations*.

We focus on designing and evaluating an engaging approach that communicates to large audiences how to effectively identify pitfalls when interpreting visualizations. We start by setting the formal definition of a *bad data visualization*, in conjunction with established taxonomies [20, 21]. After defining what bad visualization design entails, we determine a set of *learning goals* to drive further our approach design.

We identify data-GIFs as a suitable medium to show comparatively the differences between two visualization designs of interest (bad vs. good). The concept behind this solution is to expose viewers to bad and good visualizations of the same data and to make them aware of *design differences*. *Data-GIFs* are data-driven graphics interchange format (GIF) files containing short animations played in automatic repetition. Being concise in size and duration, they are versatile in conveying a single core message about the pitfalls of the visualization design, and in facilitating the comparison of different designs of the same data. Moreover, the use of data-GIFs is anticipated to engage viewers in learning visualization design concepts [1].

The *contribution* of this work is the development and assessment of data-GIFs as an effective and engaging approach for communicating pitfalls in data visualization designs to lay audiences, and for conveying how to recognize and interpret such pitfalls.

2 RELATED WORK

Recent works shed light on common design errors affecting visualizations by analyzing several misleading real-world cases and by developing a taxonomy to categorize them. Lo et al. [20] categorize 74 types of issues and form a taxonomy of misleading elements in

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.

https://www.doi.org/10.24132/CSRN.3401.7

visualizations to propose guidelines for the community. They identify five main categories of issues; namely *input*, *visualization design*, *plotting*, *perception*, and *interpretation*. McNutt et al. [21] propose a conceptual model to show how visualization mirages appear at every stage of the visual analytics process, distinguishing between *data-driven mirages* (or *data representation* issues) and *design-driven mirages* (or *data presentation* issues). Also, there is a large number of early contributions in the visualization community, providing tools for improving visualization literacy concerning the employed design and visual encodings [9, 30]. In this work, we focus on how all this knowledge can be harvested to educate lay audiences.

2.1 Tools for Visualization Literacy

Prior work has stressed the importance of the concept of *deconstruction and construction* in visualization education, and how deconstructing and constructing data visualizations can support their interpretation and design. Bishop et al. [4] developed *Construct-A-Vis*, a tablet-based tool that can guide visualization activities with children based on the learning paradigm of *constructionism*, scaffolding mechanisms, and shared interactions. This work shows the potential of a free-form constructive approach, which can lead to engaging children with data and their related mapping processes. Börner et al. [6] proposed visualization exercises based on the construction–deconstruction concept to teach visualization. The authors describe how to assess learners' insights by defining a visualization literacy framework.

Current approaches are built upon (more-or-less) complex models and implementations with particular target groups in mind, e.g., children [9]. Given their level of complexity or specific audiences, these solutions are not suitable for mass consumption, such as within a social media setup, or for more general audiences. To our knowledge, there is no previous investigation that targets lay audiences through an easy-to-implement—yet, effective and engaging—strategy for improving visualization literacy.

2.2 Data-GIFs to Convey Information

Graphics interchange format (GIF) files were released in 1987 to convey automatic, looped animations of individual, short messages without sound. Despite the increasing interest in employing GIFs to communicate information [3], there is not much research on the direction of data-driven GIFs yet. Shu et al. [28] introduced a review of data-GIFs used in the wild and investigated what makes a data-GIF understandable, by conducting a qualitative analysis. The work demonstrates the impact of the design factors of a data-GIF on how the users understand the core message presented by it. The work further proposes guidelines for designing understandable data-GIFs, without focusing on their user engagement level or the educational power. Other sources of inspiration include the large collection of data-GIFs by Jeremy Singer-Vine, the work of Lena Groeger, and Dark Horse Analytics—all showcasing different data-GIFs or short video formats categorized based on their main visualization goal.

3 CONCEPTUALIZING DATA-GIFS

Our goal is to educate *lay audiences* in recognizing a *poor visualization design* and to inform them about *a good visualization design alternative*—effectively and engagingly. The choice of focusing on lay audiences is based on the assumption that this group of viewers is anticipated to suffer the most from poorly designed visualizations. Based on the definition of our goal, we make two conceptual choices.

The first conceptual choice links to the **design space of a visualization**. The available design space is multidimensional and very complex for laypeople without prior knowledge in data visualization [20]. We, therefore, reduce this complex space to two "simple" categories: *bad* and *good* visualizations. We define *bad data visualization* as a representation that fails to effectively convey information, misleads the viewer, or obscures the underlying meaning of the data due to design flaws, inaccuracies, or poor choices in visual encoding. Oppositely, a *good data visualization* is a representation that effectively conveys information, enhances understanding, and facilitates insights by using appropriate visual design principles and techniques. An example of a bad data visualization is depicted in Fig. 1, together with its transformations into a good one.

The second conceptual choice is associated with the use of the **construction–deconstruction concept** [6, 14]. Deconstructing a visualization is simpler than starting from scratch. By comparing bad and redesigned good visualizations, learners can easily grasp the differences. This learning-by-contrast approach engages viewers and facilitates understanding.

4 DESIGNING DATA-GIFS

In this section, we identify the learning goals and requirements that drive the design of our approach.

4.1 Learning Goals

Bloom et al.'s [5] taxonomy of learning objectives classifies educational learning objectives into six complexity levels: *knowledge, comprehension, application, analysis, synthesis,* and *evaluation*. Considering our target audience, this work targets only the first three levels of the taxonomy. Based on Bloom et al.'s taxonomy and the learning-by-contrast approach, we define our learning goals:

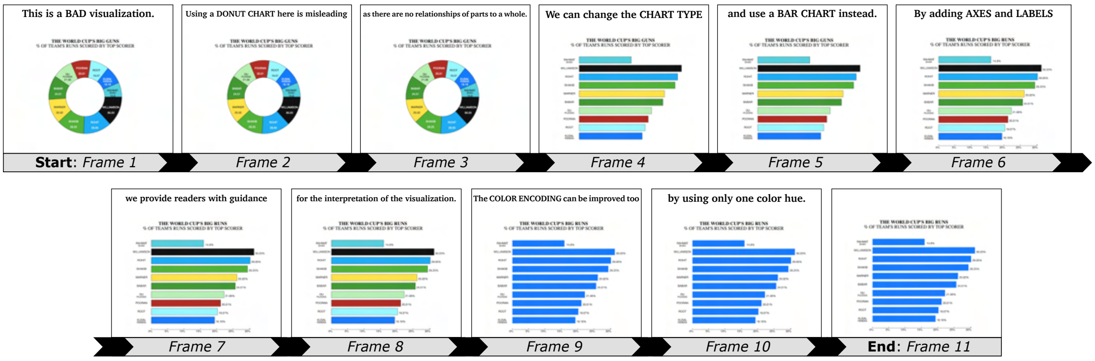


Figure 1: A data-GIF fractionated into eleven frames with a total duration of 38 *s*, where transitions (smooth or non-smooth) occur at frames 3→4, 5→6, and 8→9. Text is employed as guidance in each frame (top).

L1 ***Identify the attribute(s) of a bad data visualization, as opposed to those of a good visualization.*** For instance, learners should recognize that 3D visuals are unnecessary (bad) if the data can be sufficiently displayed in 2D (good).

L2 ***Recognize why the attribute(s) of a bad data visualization is (are) actually bad.*** For instance, learners should understand that a 3D chart is redundant if the data have only two dimensions.

L3 ***Comprehend how the attribute(s) of a bad data visualization could be improved.*** For instance, learners should recognize that an unnecessary 3D chart can be improved by simplifying it to 2D.

4.2 Design Requirements

The main requirements for our approach design are:

R1 ***Support the learning goals.*** The design of our approach should support the specific learning goals set in Sec. 4.1 to ensure that the target audience can identify *what* a bad visualization is (**L1**), *why* it is characterized as bad (**L2**), and *how* it can be transformed into a good one (**L3**).

R2 ***Be simple, understandable, and self-contained.*** Our approach must have a clear closed linear structure with a beginning (bad visualization) and an end (good visualization).

R3 ***Avoid information overload.*** The amount of information conveyed to the audience must not be overwhelming. Therefore, we should convey one message at a time.

R4 ***Maintain object constancy.*** During visual comparison, our approach must maintain object constancy, i.e., every graphical object can be visually tracked. So, secondary details should be de-emphasized to keep viewers oriented [27].

4.3 Exploring the Data-GIF Design Space

Based on what we defined in Sec. 4.1 and 4.2, we envision data-GIFs as short animations that incrementally progress from depicting a bad visualization state to a good one. For example, if a visualization has two flaws, then it comprises three *states*: the initial bad state, the state after fixing the first flaw, and the state after fixing the second flaw, which is also the final good state. Each state consists of *frames*. The way each frame transitions to the next one and how its content is communicated to the audience affects how viewers perceive, consume, and compare information through different states.

Therefore, our data-GIFs heavily rely on *comparison strategies* and a *messaging medium*, as means of communication (**R1,4**) with the audience. The former affects how the data-GIFs present their content, i.e., how they convey the comparison of a bad vs. a good data visualization state to the viewers. The latter affects which additional mechanisms the data-GIFs employ to support the viewers in understanding the conveyed message. In addition to these two aspects, we also consider in our design the guidelines proposed previously by Shu et al. [28] to improve the understandability of data-GIFs. Their work gives insights into designs that support **R2–4**, such as recommendations for the use of animation, the use of pauses to denote the end, and the structure of the visual content (i.e., text and animation).

4.3.1 Content Presentation by Comparison

We consider three different cases for the presentation of the content, inspired by previous work from the domains of comparative visualization [15] and narrative visualization [12, 16, 27]:

Smooth transitions—By interpolating the differences between frames within a specified time interval, one can smoothly transition from one frame to the next. Visualizations integrated with smooth transitions are generally preferred, being more engaging and effective in facilitating understanding [12]—given that they follow the *congruence* and *apprehension* principles [31]. We consider smooth transitions as potentially the most engaging and effective means for directing the viewer's attention to the most relevant information in the visualization, i.e., the differences between visualization states.

Interchangeability—Transitions by interchangeability consist of switching from one frame to another one without interpolation. Previous work found that interchangeable transitions are as effective as smooth transitions in some cases [16] and this solution may be preferred if the data-GIF frames are not largely different from each other [28]. Otherwise, object constancy may be affected.

Juxtaposition—In juxtaposition, all states are put next to each other [15] and there are no transitions. We consider this a special case that requires only one frame per state (i.e., it is a static visualization). Previous studies [25, 31] claim static visualizations to be equally or more effective than animation in some cases. Hence, the animation is not indispensable to effectively compare two or more visualization states—yet, the number of states should be limited.

4.3.2 Messaging Medium

During the viewing process, we need messaging mechanisms to help the viewer understand and remain attentive to the conveyed message. GIFs often include text [28] and—technically—do not support audio, which is a strong messaging medium. Other visual cues and guidance mechanisms can also be supported [18, 19, 27], but we consider them out of scope for this initial investigation. We consider only:

Text—Integrating text into our proposed data-GIFs augments their communicative value, as a viewer can easily extract information from textual explanations [22]. Previous work found that effectively linking captions, headlines, introductory text, summaries, and text annotations improves user engagement in interpreting visualizations [32]. The *intra-frame text* aims to explain how to read the visualization within the data-GIF and provides additional information about it. The *inter-frame text* plays the role of a narrator (often across states). Still, text should be staged, i.e., designed with attention to wording, number of words, style, and arrangement, to avoid information overload.

Audio—GIFs do not technically support audio. Previous studies observed that the influence of audio narration is higher than the influence of visual cues on learning outcomes [19]. Audio narrations can substitute the inter-frame text, allowing viewers to focus faster on the intra-frame text (e.g., legends or labels of the visualization). We expect that vocal guidance may reduce the number of times that a data-GIF needs to be observed by using another sensory channel to reduce visual overload. We intend to investigate whether the inherent lack of audio in GIFs is a limiting factor.

5 IMPLEMENTING DATA-GIFS

5.1 Attributes of Bad Data Visualizations

To showcase (and later assess) the design of the data-GIFs, we first need a good overview of the potential attributes characterizing bad visualizations. We start our search from existing taxonomies [20, 21] that comprehensibly summarize severe or prominent types of issues encountered in visualization designs. These taxonomies guided us in building a corpus of bad data visualizations from real-life examples. These examples come from various sources in the wild, i.e., blogs, social media, or already-existing collections of bad data visualizations such as VisLies. During a meta-selection process, we excluded scenarios, where many different issues could be encountered and kept only samples with a few (i.e., up to three) simple pitfalls. This inclusion strategy aimed to exclude complex scenarios requiring an excessively long viewing time and, therefore, not compatible with the purpose of our approach (and the typical short duration of GIFs). Hence, the attributes included in our corpus is not an exhaustive list—rather, an *indicative registry* for a first investigation. The final corpus includes 92 examples and is available in our online repository. Samples are shown in Fig. 1 and 2.

Upon collection, the first author analyzed our corpus of bad data visualizations to identify and list their pitfall(s). The other authors went through this analysis and verified its appropriateness. The arising disagreements were discussed among the authors and solved collaboratively. Successively, the first author conducted a qualitative analysis (also available in our repository) to code the encountered bad data visualization attributes into meaningful categories. Several visualizations were affected by more than one attribute and, therefore, were assigned to multiple categories. This was done in an iterative process consisting of identifying and repeatedly refining the coding until the list was saturated. The same process as in the pitfall identification verification was followed for the verification of the coding results by all authors. The coding resulted in the identification of eight indicative attributes, which are shown in Fig. 2 (together with their corresponding improvements). We hereby report them, ordered by frequency of occurrence in our corpus of bad data visualizations. Some cases were assigned to more than one category, therefore the denoted percentages do not sum up to 100%.

A1 ***Excessive display of visual elements*** *(34.8%; 32/92)*, which leads to clutter and information overload. This is a presentation issue related to *plotting* (chaotic canvas) in Lo et al.'s taxonomy [20]. An example is shown in Fig. 2 (**A1**).

A2 ***Misleading visualization axes*** *(28.3%; 26/92)*, which is a data representation issue and relates to the *visualization design* (choice of axis) stage in Lo et al.'s taxonomy. Fig. 2 (**A2**) shows an example with misleading, truncated axes.

A3 ***Inappropriate choice of visualization typology*** *(22.8%; 21/92)*, which relates to choosing the appropriate representation for a given data set.

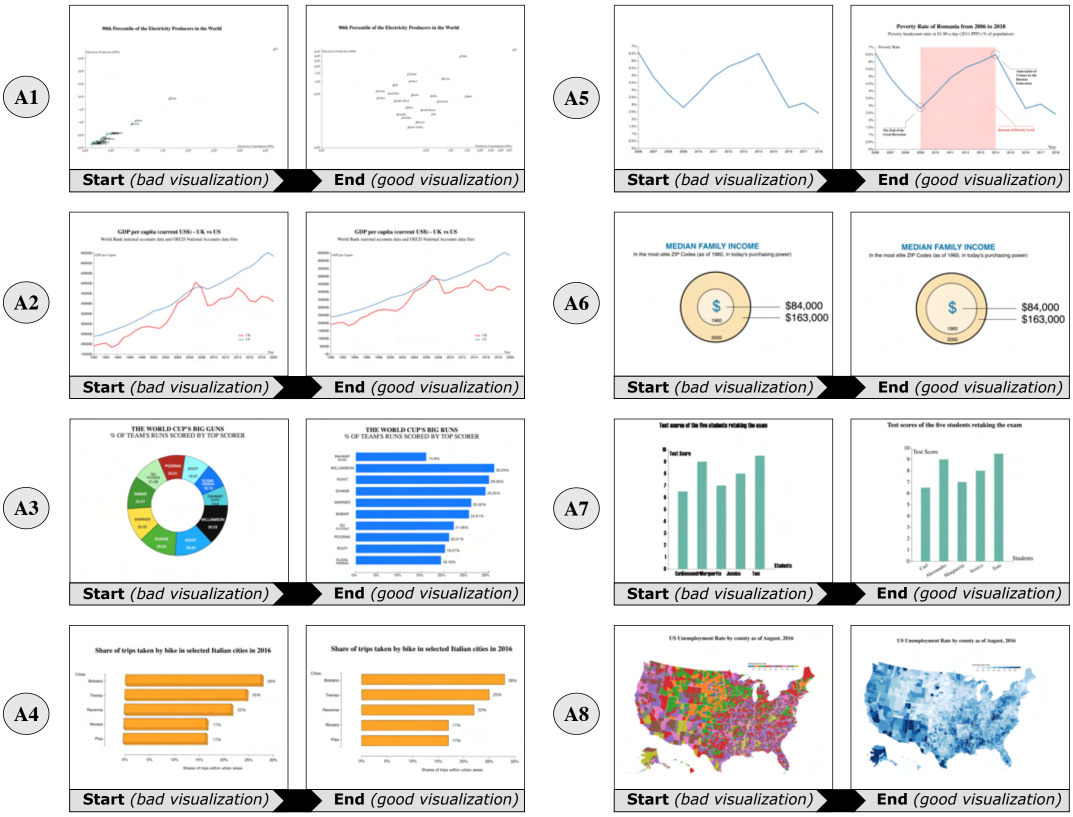


Figure 2: The eight scenarios used in the study, each covering a bad data visualization attribute (**A**): (**A1**) excessive display of visual elements; (**A2**) misleading visualization axes; (**A3**) inappropriate choice of the visualization typology; (**A4**) unnecessary use of 3D visuals; (**A5**) lack of guidance; (**A6**) miscalculated geometric areas; (**A7**) difficult-to-read text; (**A8**) wrong use of color encoding. For each scenario, we also denote its good state.

This is a data representation issue that occurs in the *visualization design* (choice of chart) stage in Lo et al.'s taxonomy. Fig. 2 **(A3)** exemplifies an inappropriate choice of visualization, where a part-to-whole visualization is used for depicting percentages that add up to more than 100%.

A4 ***Unnecessary use of 3D visuals*** *(20.7%; 19/92)*, which is also a common type of chartjunk. This is a data presentation issue that relates to the *perception* stage in Lo et al.'s taxonomy. Fig. 2 (**A4**) shows such an example.

A5 ***Lack of guidance*** *(19.6%; 18/92)*, where annotations, legends, captions, chart titles, axis labels, etc. are omitted. This is a data presentation issue and occurs in the *plotting* (incomplete chart) stage in Lo et al.'s taxonomy. An example is shown in Fig. 2 (**A5**), where annotations reveal additional information.

A6 ***Miscalculated geometric areas*** *(15.2%; 14/92)*, where data are not represented faithfully. This is a data representation issue that also happens in the *perception* stage in Lo et al.'s taxonomy. An example is shown in Fig. 2 (**A6**).

A7 ***Difficult-to-read text*** *(14.1%; 13/92)*, which is a data presentation issue and relates to the *plotting* (chaotic canvas) stage in Lo et al.'s taxonomy. Fig. 2 (**A7**) gives an example of a visualization with difficult-to-read text.

A8 ***Wrong use of color encoding*** *(13%; 12/92)*, which is a data representation issue in the *visualization design* (color mess) stage in Lo et al.'s taxonomy. Fig. 2 (**A8**) shows an example of how color can be misused and distract the audience.

5.2 Visual Narratives with Data-GIFs

We implement our data-GIFs in a linear structure that depicts an initially bad data visualization, which we then correct and transform into a good visualization. We include a visualization *state* for each design correction (i.e., the stages between the bad and the good visualization) to gradually reveal improvements. All bad visualization cases that we implemented are presented in Fig. 2 together with their final transformations into good visualizations. In this figure, we show only the bad vs. good state, but Fig. 1 shows an example with all intermediate steps for **A7**.

Fig. 1 showcases the storyboard of a data-GIF with four states shown in frames 1, 4, 6, and 9. The data-GIF has been fractionated into eleven frames (1–11). The first frame announces that "*This is a bad visualization*" and we present the depicted donut chart as a bad design choice. Then, we show *what* is wrong with the depicted visualization and explain *why*: the chart choice is not appropriate (frame 2) because there is no parts-to-a-whole relationship (frame 3). We also indicate *how* to improve the visualization by exchanging the chart type for a bar chart (frames 3→4). Similarly, we incrementally introduce and correct two more flaws: we add axes and labels for readability (frames 5→6), and we change the color encoding (frames 8→9). At the end (frame 11), we show the good state that concludes the GIF.

The frames can be set one after the other with a smooth or non-smooth *transition*. To determine the optimal transition duration for the smooth transitions in data-GIFs, we follow the suggestions by Heer and Robertson [12]. We achieve object constancy between each frame by only changing one design pitfall at a time and by keeping constant all other visual elements. If this is not possible, we opt for a keyframing animation involving tweening [29]. The duration of the animation is set based on the study by Heer and Robertson [12]. For example, the data-GIF depicted in Fig. 1 requires 19 *s*. The last frame with the good visualization includes also a pause to denote the end [28]. Alternatively, the bad vs. good data visualization states can be put side-by-side [27], similarly to the configuration in Fig. 1.

For the *messaging medium*, we use inter-frame text as in Fig. 1 (at the top of each frame), or audio. In the first case, the number of words included in the inter-frame text of data-GIFs takes into account the results of a previous study, claiming that people can read 175-300 words per minute [7]. Considering that the text may include terms unknown to the audience, either due to specific visualization and data-related terminology, or due to non-native command of English, we use the lower boundary, i.e., 175 words per minute. We, then, double the time to ensure that viewers can process the message in the data-GIF [26]. For audio, we use the same guidelines and resulting text as for the textual messaging medium case. We, then, employ an automatically generated female voice-over at a normal speed (150 words per minute) and pitch (200 Hz).

6 USER STUDY

We conducted a user study to assess four *variants* of our approach, resulting from the design space of Sec. 4.3:

V1 ***Data-GIFs with smooth transitions***, featuring smooth transitions and text as messaging medium.

V2 ***Data-GIFs with interchangeability***, featuring interchangeability and text as messaging medium.

V3 ***Static visualizations***, featuring juxtaposition and text as messaging medium.

V4 ***Data-videos***, featuring smooth transitions and audio as messaging medium.

The main goal of our study is to assess whether data-GIFs can communicate to a general audience how to distinguish bad visualizations from good ones and how to recognize common mistakes in visualization designs—*effectively*, *engagingly*, and *efficiently*. We are primarily interested in understanding which of the four variants is a more engaging and more effective tool. However, efficiency is also an interesting factor in learning as it correlates negatively with cognitive load and negative emotions, such as frustration [24]. We formulate three hypotheses for our user study:

H1 *Data-GIFs* **(V1,2)** *are* ***more effective*** *for the identification of differences between bad and good data visualization designs than static visualizations* **(V3)** *and data-videos* **(V4)**.

H2 *Data-GIFs* **(V1,2)** *support a* ***more efficient*** *identification of differences between bad and good data visualization designs than static visualizations* **(V3)** *and data-videos* **(V4)**.

H3 *Data-GIFs* **(V1,2)** *are* ***more engaging*** *for the identification of differences between bad and good data visualization designs than static visualizations* **(V3)** *and data-videos* **(V4)**.

6.1 Participants

We recruited 48 participants between 24 and 37 years old, located in 13 different countries, with different educational backgrounds and a good command of English. The recruitment was done by snowball sampling. Among the 48 participants, 16 are laypeople, 16 are professionals working with data visualizations (e.g., data scientists or journalists), and the remaining 16 are visualization experts (e.g., researchers). For conciseness, we will hereby refer to these three groups as *low*, *medium*, and *high visualization literacy* groups respectively—although the categorization reflects rather their prior experience in working with data.

6.2 Study Design

We designed eight scenarios for the user study—one for each bad data visualization attribute (**A1–8**) defined in Sec. 5.1. These are also depicted in Fig. 2. We implemented all variants (**V1–4**) for all scenarios in `d3.js` with the strategy described in Sec. 5.2. All cases are English-based, and their mean duration is 26 *s* (*SD* = 9.86 *s*), with the exception of variant **V3**, which is static. All cases are included in our repository.

We separated the 48 participants into four groups (one for each variant). Each group consists of 12 participants and is balanced with four participants from each visualization literacy group. During the study, we exposed each of the four groups to one variant using the eight designed scenarios, in randomized order. We employ a 12×8×4 mixed design with two between-subjects independent variables (i.e., variant and visualization literacy level) and one within-subjects independent variable (i.e., attribute scenarios **A1–8**). In total, the study included 384 runs (i.e., 48 participants × 8 attributes). To deal with possible effects of confounding factors, we randomized our sample by arbitrarily assigning each participant to a group, as well as the order of attribute scenarios **A1–8**.

6.3 Tasks

During the study, each participant consumed the eight attribute scenarios (**A1–8**) through one of the four variants (**V1–4**). Subsequently, they were asked to view and understand the content of each scenario, before conducting **two tasks**. First, they watched the eight scenarios and, for each scenario, performed an **XYZ test** [11] following the think-aloud method. This task targeted **H1**–**H2**. Second, they completed an **engagement questionnaire** inspired by previous works [2, 13]. This questionnaire was completed at the end of the entire session to address **H3**.

6.4 Study Setting

We conducted all studies individually and online through video calls. All meetings were recorded (screen sharing and audio) and each meeting lasted around 25–45 minutes. First, we shared the materials with the participants and instructed them verbally on what to do. For each scenario, the participants were allowed to view the given variant for as long as needed to understand its story. Yet, they were informed that the consumed time was being measured and they had to verbally communicate when "processing" was complete. Then, we started posing questions to the participants, as part of the first task (XYZ test, Sec. 6.5). The second task was provided as a link to an online engagement questionnaire (Sec. 6.6). After the study, we used the recordings to analyze the answers.

6.5 The XYZ Test

We employed the XYZ test, introduced in psychology by Haim Ginott [11], to assess whether our learning goals are met. The XYZ test consists of three open questions, which directly map back to our learning goals, expressed in a less technical language:

X ***What*** *are the differences between the initial and the final data visualization design?* (links to **L1**)

Y ***Why*** *was the initial data visualization design changed into the final one?* (links to **L2**)

Z ***How*** *would you change the initial data visualization design to obtain the final one?* (links to **L3**)

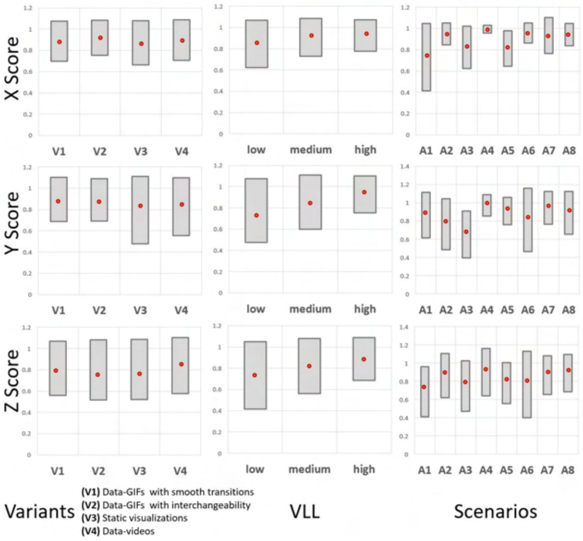


Figure 3: Crossbar plots of the X score (first row), Y score (second row), and Z score (third row) w.r.t. variant (first column), visualization literacy level (second column, VLL), and scenario (third column). The red dots indicate the mean values for each measurement.

To assess the accuracy of the responses to the XYZ questions, the first author used a weighted scoring method to assign a score (i.e., 0 = wrong, 0.25 = mostly wrong, 0.5 = semi-correct, 0.75 = mostly correct, 1 = correct) and reference answers, generated before the study. The XYZ test responses help assess **H1**, while consumption time was measured before the XYZ test to assess **H2**. A visual summary of these results is shown in Fig. 3. Our repository includes the details of the statistical analysis and we hereby summarize only the most significant outcomes.

Statistical Analysis for H1—We performed a statistical analysis on the scores related to the XYZ test, i.e., the score related to the X, Y, and Z components to investigate the effectiveness of the tested variants. A visual summary of these results is shown in Fig. 3. Given that the scores are based on the weighted scoring method mentioned above, they are categorical and ordered. For this reason, we performed multiple ordinal logistic regressions and Pearson's χ-squared tests to analyze the relationship between the XYZ scores and the factors of our study, namely the variants, the scenarios, and the visualization literacy level. We found that *participants watching data-GIFs with smooth transitions* (**V1**) are 2.16 and 2 times more likely (p=0.011 and p=0.045) to obtain *a higher Y score* (i.e., to better recognize *why* the attributes of bad data visualization are actually bad) than those watching static visualizations (**V3**) or data-videos (**V4**), respectively. This observation is shown in Fig. 3 (see first column, second row).

Additionally, we found that differences in *the XYZ scores also depend on the visualization literacy level*

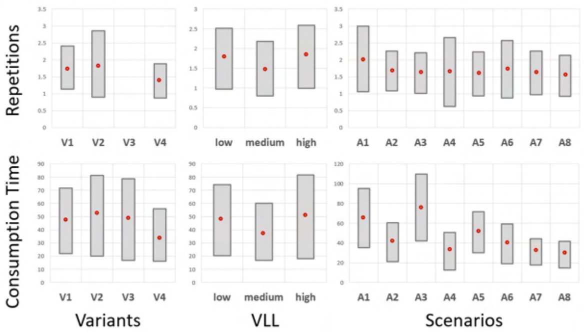


Figure 4: Crossbar plots of the watching repetitions (i.e., the ratio of consumption time over data-GIF duration, in the first row) and consumption time (second row), w.r.t variant (first column), visualization literacy level (second column, VLL), and scenario (third column). Repetitions of **V3** (static visualization) cannot be calculated. The red dots indicate the mean values for each measurement.

of the participants, as shown in the second column of Fig. 3. Our analysis results indicate participants with a high visualization literacy level are 2.19 times more likely (p=0.012) to obtain a higher Y score than participants with a medium visualization literacy level. While an average visualization literacy level is enough to correctly identify the attributes of bad data visualizations as opposed to those of good visualizations, recognizing why the attributes of bad data visualizations are actually bad requires a high visualization literacy level. As expected, *the higher the visualization literacy level of the participants the better they reflect on the acquired knowledge on how the bad data visualization can be improved* (Z score, p=0.026). Our findings also support that the *scenarios influence the score* of the participants, as depicted in the third column of Fig. 3. There are no other underlying patterns with regard to attributes or insights in this observation.

Statistical Analysis for H2—We also statistically analyzed the consumption time and the ratio of consumption time over data-GIF duration (i.e., how many times a data-GIF is watched) to assess efficiency. For these time-related continuous variables, we used ANOVA and for all the analyses, we checked all test assumptions. A summary of these results is shown in Fig. 4.

The number of repetitions needed by the participants to understand the content of the given variant was on average 1.865 for the data-GIFs with smooth transitions (**V1**), 1.739 for the data-GIFs with interchangeability (**V2**), and 1.375 for the data-videos (**V4**). For the static variant (**V3**), time cannot be measured without additional methods (e.g., eye-tracking). In essence, *data-videos need to be watched fewer times than the other two variants* (p=0.0009 and p=0.0012, respectively). This is also shown in Fig. 4 (see first row, first column). On the other side, *the consumption time is influenced by the variant, the participants' visualization literacy level, and the scenario* (p=0.004) (see the second row of Fig. 4). According to our results, scenarios **A1** and **A3** (and in some cases also **A5**) require more time than other scenarios with most of the variants (see second row, last column of Fig. 4). Participants with low and medium levels of visualization literacy needed more time for other attributes, such as **A2** (see second row, second column of Fig. 4). This was expected, as scenarios **A1**, **A3**, and **A5** contain more states between bad and good visualization, i.e., they might be more complex, or not so easy to process and remember.

6.6 The User Engagement Questionnaire

Our questionnaire for user engagement (**H3**) is inspired by previous work [2, 13] and consists of 28 questions on a seven-point Likert scale. With this questionnaire, we assess user engagement including cognitive involvement, affective involvement, enjoyment, presence, experience, and aesthetics. Our supplementary material includes the questionnaire and the statistical analysis.

Statistical Analysis for H3—We performed multiple ordinal logistic regressions to learn more about the differences between the variants. First of all, we observed no differences among the study participants concerning the perceived *aesthetics* of the variants. However, we found that the *cognitive involvement* of the participants can change based on the variant watched. According to our results, the participants watching data-GIFs with smooth transitions (**V1**) are 6.57 times more likely to *reflect on the content* during viewing than participants watching the static visualizations (**V3**) (p=0.024). Also, *learning something* new by watching the data-GIFs with smooth transitions (**V1**) is 14.56 times more likely than by static visualizations (**V3**) (p=0.001).

The visualization literacy level of participants plays also a role in their perceived *cognitive involvement*. Participants with a low visualization literacy level are 4.69 times more likely to reflect on the content and 7.32 times more likely to remember some parts of the data-GIFs than participants with a medium visualization literacy level (p=0.038 and p=0.007, respectively). Similarly, participants with a high visualization literacy level are 5.77 times more likely to remember some parts of the GIFs and 4.01 times more likely to learn something new than participants with a medium visualization literacy level (p=0.020 and p=0.047, respectively).

Finally, for data-GIFs with smooth transitions (**V1**) participants are 8.06 times more likely of being involved than for data-GIFs with interchangeability (**V2**) (p=0.007). They are also 5.24 times more likely to concentrate on the GIFs while watching data-GIFs as compared to static visualizations (**V3**) (p=0.040). Regarding the *experience*, data-GIFs with smooth transitions (**V1**) are 4.99 times more likely to be watched than static visualizations (**V3**) (p=0.036).

6.7 Summary of Findings and Discussion

Regarding our Hypotheses—The effectiveness of data-GIFs (**V1,2**) surpasses static visualizations (**V3**) in discerning bad from good data visualization designs—**affirming H1**. Data-GIFs (**V1,2**) help viewers understand why certain elements are flawed and this observation is particularly influenced by the viewers' visualization literacy and depicted scenarios. Conversely, the efficiency of data-GIFs (**H2**) is challenged, leading to **rejecting H2**. Data-videos (**V4**) consume less time and require fewer repetitions due to their audio aids, while other variants need multiple viewings. Finally, data-GIFs with smooth transitions (**V1**) and data-videos (**V4**) excel in engagement, fostering cognitive involvement and motivation for learning. Thus, **supporting H3**.

Link to Learning Goals—Identifying what distinguishes a bad visualization from a good one (**L1**) relies on the individuals' visualization literacy and the specific attributes of the flawed visualization scenario. Higher literacy correlates with better X score performance, but not necessarily with reduced performance time. Participants struggled more with complex scenarios, such as excessive visual elements or inappropriate visualization types. The suitability of tested variants for **L1 remains inconclusive**. Variants significantly impact identifying why changes were made (**L2**), with **data-GIFs yielding higher performance and smooth transitions enhancing engagement.** Finally, literacy level and scenarios affect participants' ability to reflect on transforming bad visualizations, with higher literacy aiding elaboration (**L3**). Variant suitability for **L3 remains inconclusive**.

Lessons Learnt and Future Recommendations—Overall, we recommend choosing data-GIFs with smooth transitions (**V1**) to prioritize audience engagement. Smooth transitions guide attention between states, aiding visualization interpretation. Techniques like "do-it-yourself" and "pair analytics" could be further employed to refine frame duration settings. Also, we suggest audio to expedite interpretation, especially with non-expert or complex scenarios. Juxtaposed static visualizations (**V3**), oppositely, can reduce cognitive burden when multiple tasks are involved.

The potential **generalization** of data-GIFs in educational contexts beyond mere design comparisons is significant; yet, it is accompanied by notable **limitations**. Data-GIFs offer a passive viewing experience with fixed sequences, restricting viewer interaction and dynamic engagement. Their brevity and limited information capacity pose challenges outside of contrast-based learning contexts and may impede accessibility for viewers with disabilities. While GIFs offer simplicity in communication, interactive formats provide greater versatility and engagement, ensuring broader usability and personalized experiences.

Regarding the **design of effective data-GIFs**, beyond content presentation and messaging medium, other features such as interactive capabilities, visual cues like highlighting and annotations, and aesthetic considerations warrant exploration to enhance engagement and effectiveness. Individual differences, including visualization literacy levels and background, influence learning performance and user engagement, necessitating further investigation [10, 23]. **Further studies** incorporating diverse demographics, and cultural or learning backgrounds, as well as other evaluation methods, are encouraged to ascertain the effectiveness of data-GIFs in visualization education—particularly in terms of memorability and perceived workload.

7 CONCLUSION

We proposed data-GIFs as educational tools for engaging viewers in identifying effectively differences between bad and good data visualization designs. Our study with 48 participants indicates that smooth transitions are engaging and effective, especially for recognizing *why* visualization designs are misleading. However, our results show that data-videos are generally more efficient than the other variants in terms of consumption time, suggesting that the inherent lack of audio messaging in GIFs is a limiting factor. In our future work, we intend to investigate further the design space of data-GIFs and explore its full potential for visualization education.

8 ACKNOWLEDGMENTS

The paper was partially written in collaboration with the VRVis Competence Center (funded by BMVIT, BMWFW, Styria, SFG, and Vienna Business Agency in the scope of COMET under Grant 854174 managed by FFG).

9 REFERENCES

[1] E. Altintas et al. Evaluation of Use of Graphics Interchange Format (GIF) Animations in Mathematics Education. *Educational Research and Reviews*, 12(23):1112–1119, 2017.

[2] F. Amini et al. Hooked on Data Videos: Assessing the Effect of Animation and Pictographs on Viewer Engagement. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, pages 1–9, 2018.

[3] S. Bakhshi et al. Fast, Cheap, and Good: Why Animated GIFs Engage Us. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 575–586, 2016.

[4] F. Bishop et al. Construct-A-Vis: Exploring the Free-Form Visualization Processes of Children. *IEEE Transactions on Visualization and Computer Graphics*, 26:451–460, 2020.

[5] B. S. Bloom et al. Taxonomy of Educational Objectives: The Classification of Educational Goals. *Handbook I: Cognitive domain*, 1956.

[6] K. Börner et al. Data Visualization Literacy: Definitions, Conceptual Frameworks, Exercises, and Assessments. *Proceedings of the National Academy of Sciences*, 116(6):1857–1864, 2019.

[7] M. Brysbaert. How Many Words do we Read per Minute? A Review and Meta-analysis of Reading Rate. *Journal of Memory and Language*, 109:104047, 2019.

[8] A. Cairo. *How Charts Lie: Getting Smarter about Visual Information*. WW Norton & Company, 2019.

[9] F. Chevalier et al. Observations and reflections on visualization literacy in elementary school. *IEEE Computer Graphics and Applications*, 38(3):21–29, 2018.

[10] Y. Cui et al. Adaptive assessment of visualization literacy. *arXiv preprint arXiv:2308.14147*, 2023.

[11] H. G. Ginott. *Teacher and Child: A Book for Parents and Teachers*. Scribner Paper Fiction, 1972.

[12] J. Heer and G. Robertson. Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, 2007.

[13] Y.-H. Hung and P. Parsons. Assessing User Engagement in Information Visualization. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 1708–1717, 2017.

[14] S. Huron et al. Constructive visualization. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS 14, pages 433–442, 2014.

[15] K. Kim et al. Comparison Techniques Utilized in Spatial 3D and 4D Data Visualizations: A Survey and Future Directions. *Computers & Graphics*, 67:138–147, 2017.

[16] Y. Kim et al. Designing Animated Transitions to Convey Aggregate Operations. *Computer Graphics Forum*, 38(3):541–551, 2019.

[17] C. N. Knaflic. *Storytelling with Data: A Data Visualization Guide for Business Professionals*. John Wiley & Sons, 2015.

[18] H.-K. Kong et al. Internal and External Visual Cue Preferences for Visualizations in Presentations. *Computer Graphics Forum*, 2017.

[19] H.-K. Kong et al. Understanding Visual Cues in Visualizations Accompanied by Audio Narrations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–13, 2019.

[20] L. Y.-H. Lo et al. Misinformed by visualization: What do we learn from misinformative visualizations? In *Computer Graphics Forum*, volume 41, pages 515–525, 2022.

[21] A. McNutt et al. Surfacing visualization mirages. In *Proceedings of the 2020 CHI Conference on human factors in computing systems*, pages 1–16, 2020.

[22] A. Ottley et al. The Curious Case of Combining Text and Visualization. In *EuroVis (Short Papers)*, pages 121–125, 2019.

[23] S. Pandey and A. Ottley. Mini-vlat: A short and effective measure of visualization literacy. *arXiv preprint arXiv:2304.07905*, 2023.

[24] R. Pekrun et al. Measuring emotions in students' learning and performance: The Achievement Emotions Questionnaire (AEQ). *Contemporary Educational Psychology*, 36(1):36–48, 2011.

[25] G. Robertson et al. Effectiveness of Animation in Trend Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, 2008.

[26] F. Schmidt-Weigand et al. A Closer Look at Split Visual Attention in System- and Self-paced Instruction in Multimedia Learning. *Learning and Instruction*, 20(2):100–110, 2010.

[27] E. Segel and J. Heer. Narrative Visualization: Telling Stories with Data. *IEEE Transactions on Visualization and Computer Graphics*, 16:1139–1148, 2010.

[28] X. Shu et al. What Makes a Data-GIF Understandable? *IEEE Transactions on Visualization and Computer Graphics*, 27:1492–1502, 2021.

[29] J. Thompson et al. Understanding the Design Space and Authoring Paradigms for Animated Data Graphics. In *Computer Graphics Forum*, volume 39, pages 207–218. Wiley Online Library, 2020.

[30] E. R. Tufte et al. Visual Explanations: Images and Quantities, Evidence and Narrative, 1998.

[31] B. Tversky et al. Animation: Can it Facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262, 2002.

[32] Q. Zhi et al. Linking and Layout: Exploring the Integration of Text and Visualization in Storytelling. In *Computer Graphics Forum*, volume 38, pages 675–685. Wiley Online Library, 2019.

Attention-Aware DAE for Automated Solar Coronal Loop Segmentation

Prabal Man Dhaubhadel[1], Jong Kwan Lee[1], Qing Tian[1,2]
[1]Dept. of Computer Science, Bowling Green State University, Bowling Green, OH 43403, USA
[2]Dept. of Computer Science, University of Alabama at Birmingham, Birmingham, AL 35294, USA
`pdhaubh@bgsu.edu, leej@bgsu.edu, qtian@uab.edu`

ABSTRACT

This paper introduces an enhanced Denosing Autoencoder (DAE) model, incorporating a novel attention mechanism, for the segmentation of solar coronal loops. This work is based on DAE framework to address the segmentation challenges posed by intricate structures of coronal loops which also appear with other solar features and image noises. Specifically, we introduce Encoding-Aware Decoding Attention (EADA) to all decoding stages of DAE, which resulted in improvement in coronal loop segmentation. Our models are validated through experiments on a synthetic image dataset of 11,000 images and a test dataset of 165 real coronal images of the NASA's Solar Dynamics Observatory (SDO) satellite mission. Compared to the state-of-the-art coronal loop segmentation baseline, our attention-enhanced model results in better loop gap-filling and higher segmentation metrics (i.e., 3.6% increase in accuracy, 11.4% better recall and 5.6% higher precision).

Keywords
Denoising Autoencoders (DAE), Attention Block, Solar Coronal Loop Segmentation, Loop Gap-Filling

1 INTRODUCTION

In recent years, deep learning algorithms have emerged as powerful tools in image processing and computer vision, offering significant improvements in handling complex patterns and structures in various application domains. In this paper, we introduce an attention-enhanced deep learning model for accurate solar feature segmentation.

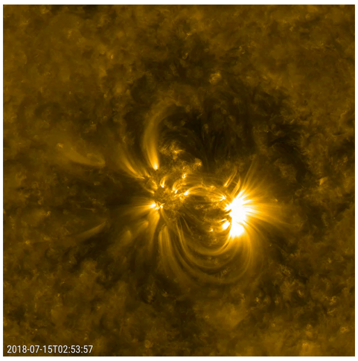


Figure 1: A sub-image of SDO coronal image (image credit: NASA's SDO Mission).

Solar coronal loops, which are highly structured and dynamic features in the solar corona, play a crucial role in understanding solar physics and the mechanisms behind solar activities. These solar activities can impact our daily life (e.g., disrupting electrical grids, damaging communication satellites and navigation systems). The coronal loops, visible in extreme ultraviolet (EUV) and X-ray wavelengths, are fundamentals in studying the Sun's magnetic field structure and its evolution. However, the accurate segmentation of these loops from solar images is a very challenging task due to their intricate structures, varying brightness, and the presence of other solar features and image noises. Fig. 1 is an example of coronal loop sub-image from NASA's Solar Dynamics Observatory (SDO) satellite mission. As shown in the image, the coronal loops (i.e., the bright arcing structures) have complex shapes with varying intensities and blurry boundaries. There are also non-loop features and noises on the image.

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 paper, we present a Denoising Autoencoder (DAE) enhanced by a new attention module, aimed at advancing coronal loop segmentation. To our knowledge, our work is the first attempt to consider attention-based DAE for coronal loop segmentation. The paper is organized as follows. In Section 2, we discuss related works. Section 3 introduces our new method and Section 4 outlines the experimental setup followed by experimental results. Section 5 concludes the paper.

https://www.doi.org/10.24132/CSRN.3401.8

2 RELATED WORK

2.1 U-Net and DAE

Machine learning/deep learning algorithms have been used in many image segmentation applications (e.g., [**gon16**, **lk16**, **tme22**]). Among them, U-Net [**rfb15**] and Denoising Autoendcoder (DAE) [**vlb08**] have widely been used recently.

U-Net (e.g., [**rfb15**, **bkc17**]) is based on the encoder-decoder architecture that follows a symmetric contracting and expansive path. U-Net's unique feature lies in its use of skip connections, facilitating the direct information propagation from contracting layers to corresponding layers in the expansive path. These horizontal propagation structures, known as skip connections, allow the concatenation of contractive path information and expansive path information before up-sampling.

DAE [**vlb08**] is a class of autoencoders that normally operate on an unsupervised learning paradigm, where the network is trained to reconstruct clean data from corrupted input during the training phase. The training process involves introducing corruption to the input data and optimizing the model to minimize the reconstruction error, encouraging the network to capture meaningful features while filtering out irrelevant noise. Feature extraction and dimensionality reduction tasks are efficiently handled by the denoising autoencoder (e.g., [**fbh17**, **vll10**, **xmy16**]). DAEs showed promising results in removing noise while completing the encoding-decoding process. However, not as many DAEs have not been studied for segmentation.

2.2 Solar Coronal Loop Segmentation

Several methods have been introduced for solar coronal loop segmentation. Oriented Connectivity-based method (OCM) [**lng06a**] and Dynamic Aperture-based method (DAM) [**lng06b**] employed image processing and constructive feature segmentation by exploring physical constraints and coronal loop feature's intensity profiles, respectively. An enhanced version of the OCM has also been introduced [**asc10**]. A solar loop mining system which includes a block-by-block loop segmentation for retrieving coronal loop images has been introduced by Durak *et al.* [**dnc09**, **dnc10**]. McAteer *et al.* [**mka10**] utilized a 2D Wavelet-based smoothing function as an edge detector for segmenting coronal loops. Lee and Tang [**let11**] presented an active contour-based model where the minimization of the spline energy constrained by the physical shape of the coronal loop's intensity profiles allowed more robust loop segmentation. Zhiming *et al.* [**zxz19**] presented a coronal loop segmentation method that exploits a clustering algorithm based on approximated local directionality determined by a match and image enhancing filters. To our best knowledge, a recent work by Moradi *et al.* [**mlt21**] was the only work that explored a deep learning algorithm (i.e., a U-Net) for segmenting the coronal loops.

2.3 Attention Module for Enhanced Image Segmentation

Self-Attention [**gsy21**], Bottleneck Attention Module (BAM) [**pwl18**], Convolutional Block Attention Module (CBAM) [**wpl18**] that focus on channel and spatial attention, and criss-cross attention [**hwh19**] are all attention-equipped approaches to reinforce their respective neural networks. Using denoising autoencoders, missing data imputation is achieved by implementing mask attention in DAEMA [**tfj21**]. Real Image Denoising [**anb19**] with feature attention uses attention in a modular manner and achieves superior performance. Motivated by the use of attention in neural networks, we propose our feature attention block for the denoising autoencoder that directs the model's attention to foreground pixels in segmenting coronal loops. As we will show later, this novel approach leads to improved performance metrics. It can also enhance the segmentation of loops previously categorized as "disconnected". We refer to this segmentation of disconnected loops into one cohesive loop as the *gap-filling effect* in this paper.

3 OUR METHODOLOGY

In this section, we introduce our new attention-aware denoising autoencoder (AtnDAE) for coronal loop segmentation. Particularly, we propose the *Encoding-Aware Decoding Attention (EADA)*, a new attention mechanism that enhances the segmentation of coronal loops by integrating attention at each stage of the decoding path, taking into account the corresponding layer from the encoding path. Fig. 2 illustrates the

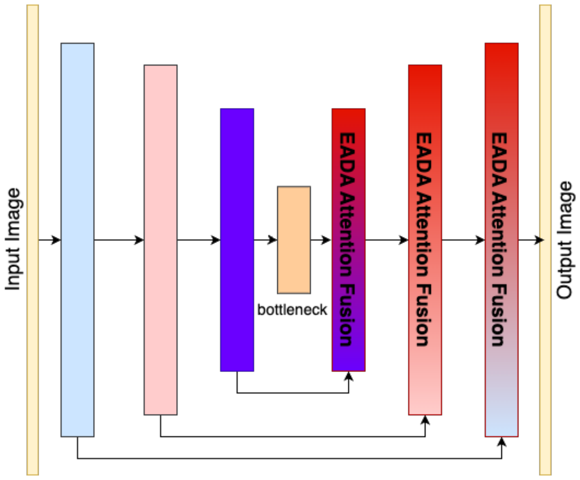


Figure 2: Illustration of our Attention-aware DAE architecture. Details of the Encoding-Aware Decoding Attention (EADA) block are shown in Fig. 3.

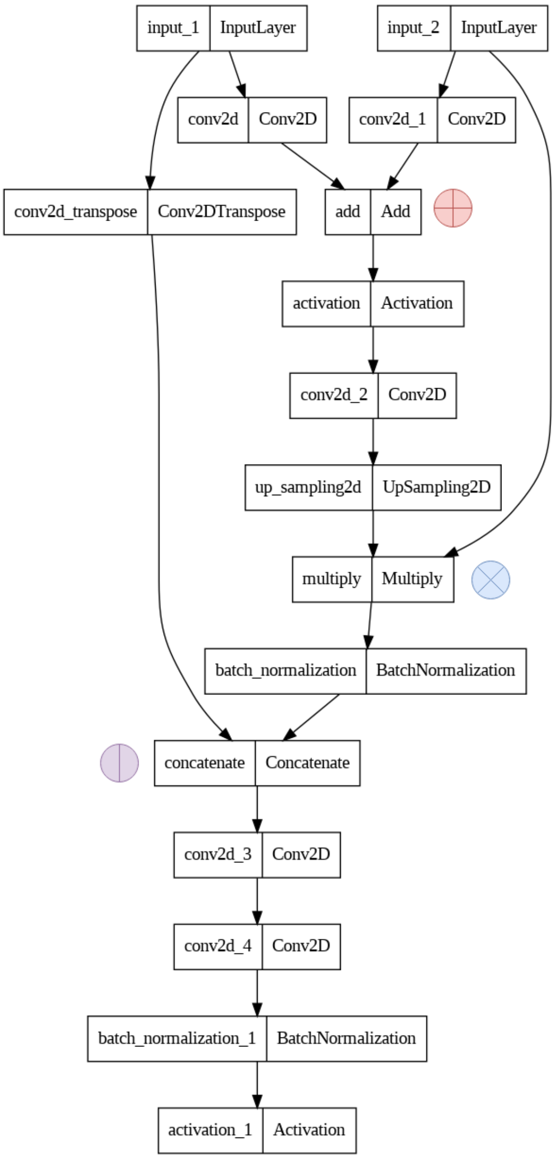

input_1 | InputLayer
input_2 | InputLayer
conv2d | Conv2D
conv2d_1 | Conv2D
conv2d_transpose | Conv2DTranspose
add | Add
activation | Activation
conv2d_2 | Conv2D
up_sampling2d | UpSampling2D
multiply | Multiply
batch_normalization | BatchNormalization
concatenate | Concatenate
conv2d_3 | Conv2D
conv2d_4 | Conv2D
batch_normalization_1 | BatchNormalization
activation_1 | Activation

Figure 3: Encoding-Aware Decoding Attention (EADA) block. This block is integrated at each stage of our attention-enhanced DAE (Fig. 2). It utilizes information from both the decoding input (input_1) and its corresponding encoding features (input_2).

overall architecture of our AtnDAE, with the details of our EADA attention fusion block depicted in Fig. 3. Our novel attention mechanism is particularly beneficial for addressing imbalanced data like corona images, where the background pixels substantially outnumber the foreground pixels. Each EADA block selectively emphasizes the foreground loop structures, which are often overshadowed by the complex background of solar images.

As shown in Fig. 2, our EADA attention fusion block is integrated at every stage of the decoding path, leveraging information from both the decoding input (i.e., input_1 in Fig. 3) and the corresponding encoding layer (i.e., input_2 in Fig. 3) at each stage. This setup ensures that the EADA mechanism has access to both more location-accurate features in the encoder and highly semantic features in the decoder. The information from the encoding phase directs more accurate attention toward the foreground loop structures, facilitating the refinement and amplification of the distinctive curvilinear features found in coronal loops. Through dot multiplication, such attention is applied to a transformed combination of encoding and decoding features. The above attention-enhanced results are then concatenated with the regular decoding output, followed by further convolutions before arriving at the output for the EADA block.

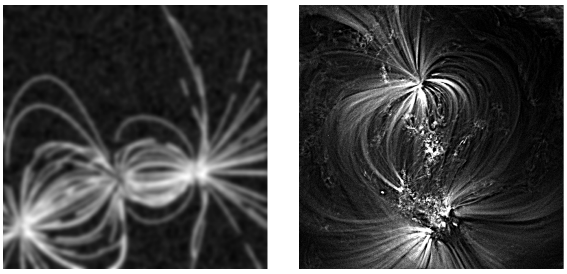

(a) synthetic image (b) real image

Figure 4: A sample of synthetic and real images.

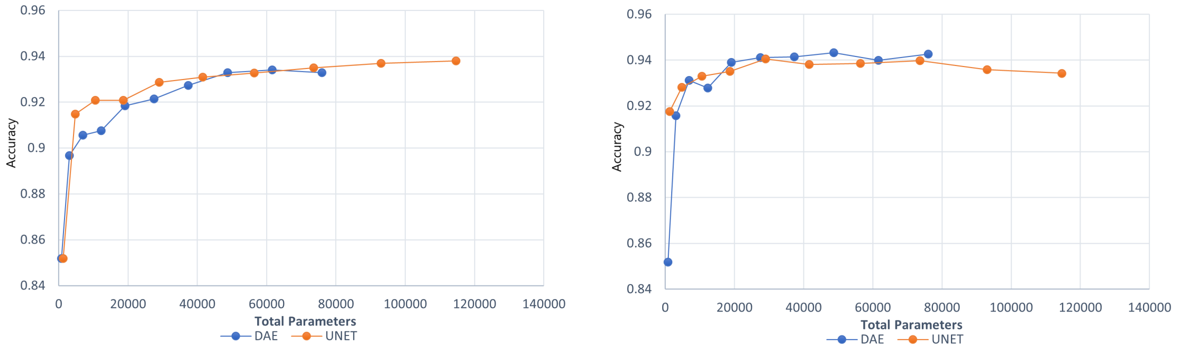
In our attention-aware DAE, batch normalization is used to reduce the chance of overfitting and improve the training stability. ReLU is used as the activation function.

4 EXPERIMENTS AND RESULTS

This section first outlines the dataset and experimental setup utilized in our study. Then, we provide a comparative assessment of our novel attention-aware denoising autoencoder (AtnDAE), including both quantitative and qualitative analyses. Specifically, we compare the performance of the U-Net and traditional DAE models in the context of coronal loop segmentation and present the results of our improved attention-aware DAE.

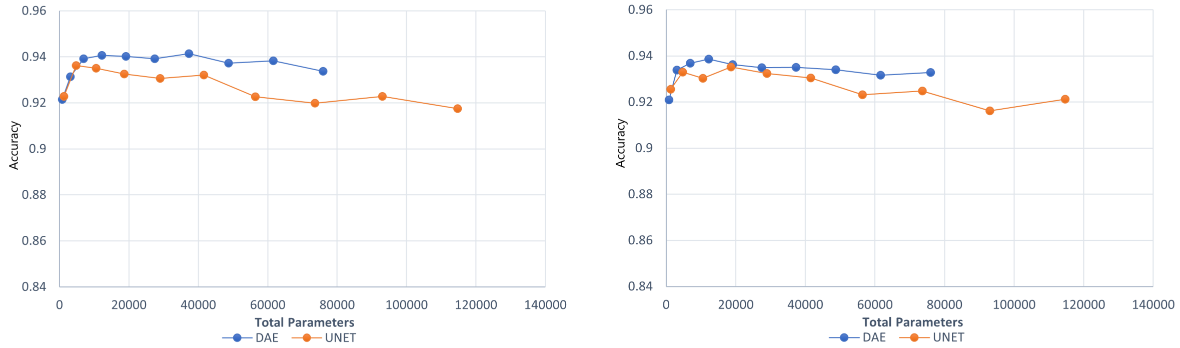
4.1 Dataset

In our experiments, 11,000 synthetic coronal images and 165 real coronal images were used. The synthetic coronal images were created using the scheme, including the image noises, used in [**lng06a**, **lng06b**, **let11**] as it was one of the most popularly used synthetic images of the solar corona. The synthetic images were of 256×256 and the coronal loops were represented as the magnetic field lines derived from a physical magnetic model. (Solar physicists consider the corona loops as the traces of the magnetic fields around the Sun.) The primary benefit of using the synthetic images based on this scheme is that the ground truth masks (i.e., labels)



(a) 16×16

(b) 32×32



(c) 64×64

(d) 128×128

Figure 5: Accuracy over range of parameters for different bottleneck sizes-16×16, 32×32, 64×64, 128×128.

are available for training. In our synthetic image set, we have included loops that are broken (e.g., “gaps" within a loop, abruptly ending loops) which are often present in the real coronal loops due to 3D-to-2D projection, viewpoint occlusion, and noise and non-loop features in the image. No other prior study have considered this in coronal loop segmentation. 10,000 synthetic images were used for 80-20 training-validation split. The rest of 1,000 synthetic images were used as the test set. The real coronal images were obtained from the NASA’s SDO satellite mission. The ground truth masks (i.e., labels) were determined from manual segmentation. Fig. 4 shows a sample of synthetic and real corona images.

4.2 Experimental Setup

All executions utilize a conda environment with TensorFlow version 2.8, cuda version 11.2.2, and Python version 3.8.12. A test on a range of 10 models is performed to select the best-performing model on each of the bottleneck sizes. Each model is trained for 200 epochs. The training of AtnDAE is optimized using the Adam optimizer [**kib14**] with a learning rate carefully selected to balance fast convergence and training stability. A range of learning rates between 0.1 to 0.00001 was tested. For DAE, a learning rate of 0.01 is selected while a learning rate of 0.002 is used for U-Net. To make a fair comparison, all the key hyperparameters are kept the same for both types of architectures, including bottleneck size, filter count, layer count, input and output dimensions, and batch size.

4.3 Quantitative Comparison

In this and the next subsection, we conduct quantitative and qualitative comparative analyses between U-Net and DAE models in the segmentation of coronal loops, providing rationale for selecting the DAE model as the basis for our approach. The improved results of our attention-aware DAE will be shown in Sec. 4.5.

4.3.1 Accuracy, Recall, Precision

Fig. 5 illustrates the accuracy over a range of different numbers of parameters for four different bottleneck sizes. The top sub-figures demonstrate that accuracy tends to plateau at approximately 94% for bottleneck sizes of 16×16 and 32×32. For larger bottleneck sizes of 64×64 and 128×128, the accuracy of the DAE saturated at around 94%, whereas the U-Net’s accuracy decreases, leveling off at about 93%. The DAE reached its highest accuracy with a bottleneck size of 32×32. The most significant discrepancy in accuracy between DAE and U-Net occurred at bottleneck sizes larger than 64×64. Among all 40 models tested, the DAE reached

Table 1: Performance metrics across bottleneck sizes using the synthetic dataset.

Category	Accu.	Recall	Prec.	#Params
16×16_DAE	0.921	0.649	0.783	27,457
16×16_U-Net	**0.929**	**0.699**	**0.795**	28,986
32×32_DAE	**0.941**	0.766	**0.825**	27,457
32×32_U-Net	0.940	**0.774**	0.816	28,986
64×64_DAE	**0.939**	**0.767**	**0.812**	19,101
64×64_U-Net	0.931	0.722	0.792	18,653
128×128_DAE	**0.939**	0.749	**0.821**	12,257
128×128_U-Net	0.930	**0.794**	0.750	10,588

a peak accuracy of 94.1%, slightly higher than the U-Net counterpart.

Next, the accuracy, recall, and prevision for the models of similar sizes are compared and shown in Table 1. Each model had 4 bottlenecks. The bold values indicate the best performance within each bottleneck size for the respective metric. As the bottleneck size increases, the best performance shifts between the models for different metrics. Overall, DAE outperforms U-Nets, particularly in terms of accuracy and precision at large bottleneck sizes.

4.3.2 *ROC Curve and AUC*

The Receiver Operator Characteristic (ROC) curve plots the true positive rate against the false positive rate, and the Area Under the Curve (AUC) summarizes the ROC Curve. Across the different bottleneck sizes, the ROC Curves were very similar, converging nearly to 1.0 before false positive rate reaching 0.2. (The ROC Curve plots are omitted due the space limit.) The AUCs for U-Net and DAE were also similar, approximately around 0.973. (Here, we note that the AUC of our new attention-aware DAE was 0.992.)

4.3.3 *Confusion Matrix*

Fig. 6 and Fig. 7 illustrate the confusion matrices for DAE and U-Net, respectively. (For an imbalanced dataset like the coronal images, the number of 0s, i.e. background, is much higher than the number of 1s, i.e., loop pixels.) According to the results, the percentage of predicted true positives is higher for DAE for the bottleneck sizes of 32×32, 64×64, and 128×128. (This matches the accuracy shown in Table 1.)

4.3.4 *Performance and Training Data Size*

For the next analysis, we explore how the size of the training data impacts model performance. This investigation aids in identifying which model type is more data efficient and which is more susceptible to overfitting. Fig. 8 shows the accuracy of the U-Net and DAE for training data sizes of 500, 1,000, 5,000, and 10,000. According to the results, DAE performed better across

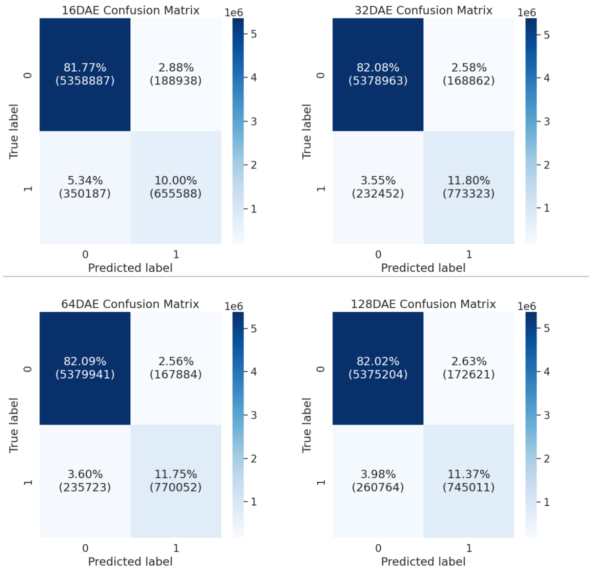


Figure 6: Confusion Matrices for DAE on our dataset across each of the tested bottleneck sizes - 16×16, 32×32, 64×64 and 128×128.

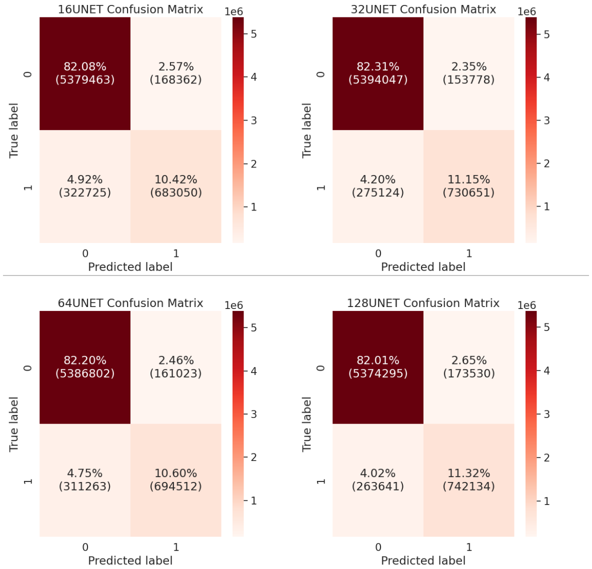


Figure 7: Confusion Matrices for U-Net on our dataset across each of the tested bottleneck sizes - 16×16, 32×32, 64×64 and 128×128.

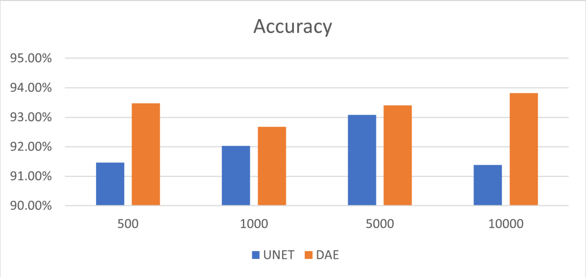


Figure 8: Accuracy vs. training data size (e.g., 500, 1,000, 5,000, and 10,000). Model bottleneck size: 16×16.

all training data sizes. Table 2 shows different model performance metrics across various training data sizes. As we can see, U-Net is less data efficient as the accuracy is much lower when the training data is limited (e.g., 500). This lower accuracy also demonstrates U-Net's higher vulnerability to overfitting. As the training data amount increases, the accuracy of U-Net improves. However, it decreases again when the data size reaches 10,000. On the other hand, DAE consistently maintains higher accuracy across varying sizes of training data.

4.4 Qualitative Comparison

In this subsection, we offer a qualitative assessment of U-Net and DAE by discussing visual observations of model predictions and evaluating the models' effectiveness in gap-filling.

4.4.1 *Visual Observations*

Fig. 9 shows the predictions of U-Net and DAE for both a synthetic coronal image sample (in (a)) and a real image sample (in (b)). The test image and its ground truth (i.e., labels) are shown on the left-most column of each sub-figure. The prediction results of the models using four different bottleneck sizes are shown on the four right columns, labeled by the bottleneck sizes.

In the predictions on synthetic images, both U-Net and DAE produced very comparable results, with the predicted curves appearing fuzzier when smaller bottleneck sizes were utilized in both models. In predictions on the real image, U-Net demonstrated a great ability to accurately capture curvilinear structures when the bottleneck size is small (16×16). This is likely attributed to the skip connections in U-Net. However, as the bottleneck size increased, DAE's predictions improved and seemed more accurate, whereas U-Net produced significantly more false positives than DAE.

4.4.2 *Gap-Filling Effects*

As described in Section 4.1, certain loops may exhibit discontinuities. Fully segmenting such loops would help solar physicists in understanding the underlying properties of the coronal loops. Fig. 10 demonstrates the gap-filling performance by the U-Net and DAE (we select the bottleneck size of 32×32 according to Table 1). Sub-figure (a) shows the ground truth magnetic field lines. Sub-figure (b) is the synthetic corona image generated from the magnetic field lines shown in sub-figure (a). As shown in the figure, there are disconnected loops, such as those in the top left and bottom left corners. Sub-figures (c) and (d) show the predictions of U-Net and DAE. As shown in the figures, both prediction models achieved reasonably well accurate segmentation. However, the DAE demonstrated a more noticeable gap-filling effect than the U-Net.

Table 2: Results of 16×16×256 model on various training data sizes. # of parameters were 784,385 and 1,179,121 for DAE and U-Net, respectively

Models	Size	Accu.	Recall	Prec.
DAE	500	**93.48%**	**78.67%**	**78.77%**
U-NET	500	91.47%	78.67%	68.47%
DAE	1,000	**92.68%**	66.94%	**80.35%**
U-NET	1,000	92.03%	**79.70%**	70.40%
DAE	5,000	**93.41%**	**86.00%**	73.82%
U-NET	5,000	93.09%	80.28%	**74.88%**
DAE	10,000	**93.82%**	80.30%	**78.51%**
U-NET	10,000	91.38%	**80.52%**	67.56%

4.5 Results of New Attention-aware DAE

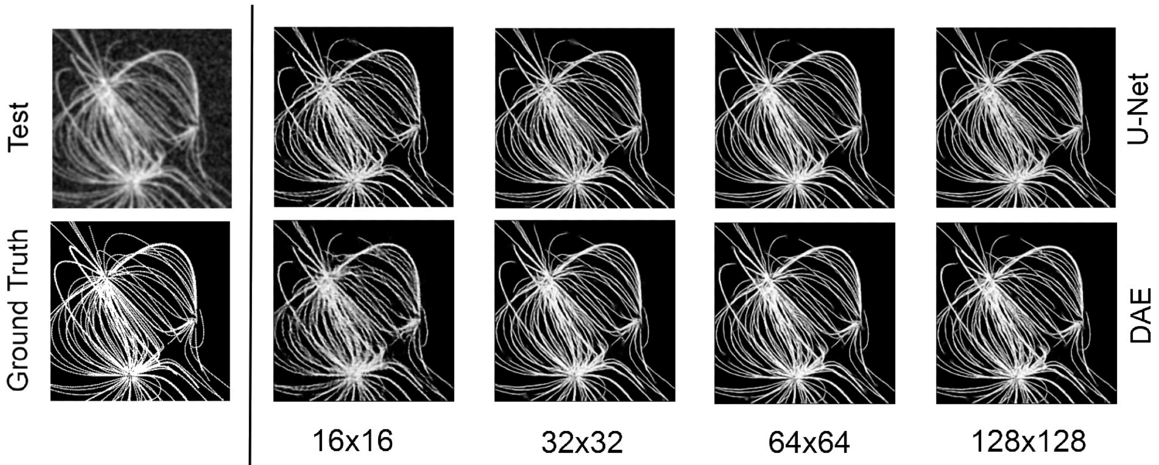
Next, we will analyze the performance of our attention-aware DAE (i.e., AtnDAE) based on the proposed Encoding-Aware Decoding Attention (EADA) mechanism. Compared with competing baselines, our novel AtnDAE exhibited superior segmentation performance, encompassing the gap-filling effect, in both quantitative and qualitative dimensions. As shown in Table 3, our AtnDAE outperformed DAE significantly in accuracy, recall, and precision, underscoring the effectiveness of our attention mechanism.

Fig. 11 shows the prediction of our AtnDAE on a synthetic image example. In the figure, regions of the image where the gap-filling effect is noticeable are highlighted with red circles. As shown in the figure, all loops were not only distinctly segmented (with no blurring or missing loops), but all gaps were also effectively filled. Fig. 12 demonstrates the comparison of the gap-filling effect among U-Net, DAE, and our new AtnDAE. Our AtnDAE not only achieved superior clarity in segmenting the loops but also effectively addressed gaps within them, areas overlooked by both the U-Net and DAE models. Fig. 13 shows the confusion matrix for our AtnDAE model. Compared to the DAE and U-Net models, there was an increase in both true positives and true negatives, accompanied by a decrease in both false positives and false negatives. For example, the false positives and false negatives were 3.55% and 2.58% for the DAE, 4.20% and 2.35% for the U-Net, and 1.54% and 1.81% for the AtnDAE. (Refer to the top-right sub-images in Fig. 6 for the DAE and in Fig. 7 for the U-Net, respectively.)

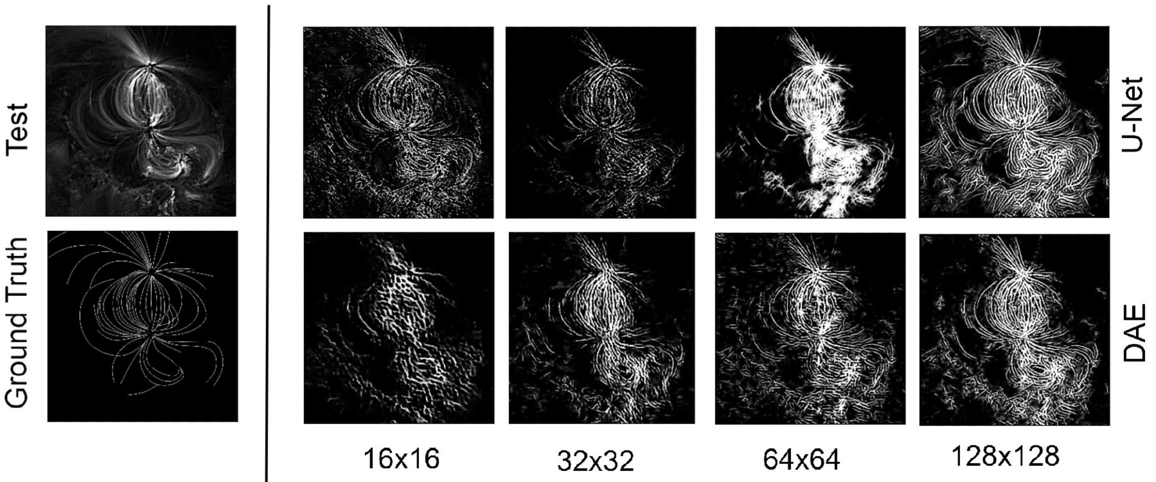
More segmentation results obtained with our AtnDAE are displayed in Fig. 14.

Table 3: Performance of DAE and AtnDAE.

Model	Accu.	Recall	Prec.
32x32-DAE	0.939	0.741	0.829
32x32-AtnDAE	**0.967**	**0.888**	**0.872**



(a) Predictions on Synthetic Image



(b) Predictions on Real Image

Figure 9: Predictions of U-Net and DAE using different bottleneck sizes.

4.6 Ablation Study: Influence of Loss Functions

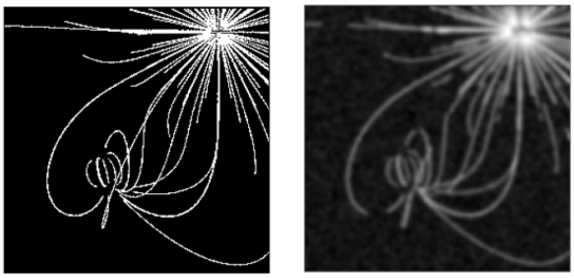
We have investigated nine loss functions for coronal loop segmentation. Table 4 presents the selected loss functions with their corresponding performance metrics. In the table, the top three values for each metric are highlighted. The binary cross-entropy, Binary Dice, and Dice Loss had the highest accuracy. Combo Loss, Focal Tversky, and Tversky Loss had the highest recall. The binary cross-entropy, Focal Loss, and Binary Tversky had the highest precision. Given the results, we use binary cross entropy for all previous experiments.

Table 4: Performance of different loss functions. Base model: 32x32 DAE. Accu.: Accuracy, Prec.: Precision.

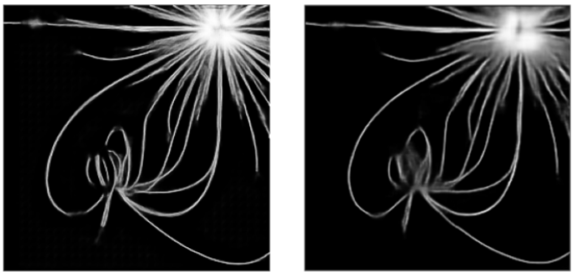
Loss Function	Accu.	Recall	Prec.
BinaryCrossEntropy	**0.9390**	0.7670	**0.8120**
Binary Dice	**0.9388**	0.8511	0.7633
Dice Loss	**0.9378**	0.8446	0.7615
Focal Loss	0.9375	0.7426	**0.8189**
HED Loss	0.9242	0.9325	0.6775
Combo Loss	0.9216	**0.9387**	0.6673
Binary Tversky	0.9190	0.5160	**0.8913**
Focal Tversky	0.9121	**0.9514**	0.6359
Tversky Loss	0.9117	**0.9517**	0.6347

5 CONCLUSION

In conclusion, we present a new approach for automated segmentation of solar coronal loops through the development of the attention-aware Denoising Autoencoder (AtnDAE). This novel approach, integrating an Encoding-Aware Decoding Attention (EADA) mechanism, has significantly improved the segmentation accuracy, demonstrating superior performance over competing models such as the U-Net and DAE. The quantitative and qualitative assessments highlight the AtnDAE's improved capacity for achieving higher levels of accuracy, precision, and recall. Additionally, they underscore its effectiveness in addressing the inherent challenges of gap-filling in coronal loop imagery. These findings underscore the significance of integrat-

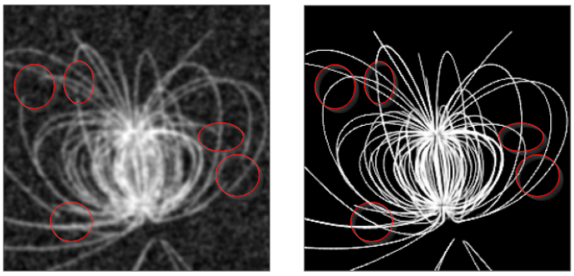


(a) ground truth mask (b) test image with gap



(c) U-Net prediction (d) DAE prediction

Figure 10: Gap-filling by U-Net and DAE. Number of layers: 6. Bottleneck size: 32×32.



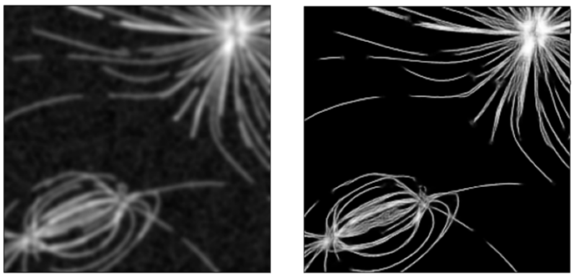
(a) Synthetic Image (b) AtnDAE Prediction

Figure 11: AtnDAE prediction on a synthetic corona image with gap-filling effects marked. Number of layers: 6. Bottleneck size: 32×32.

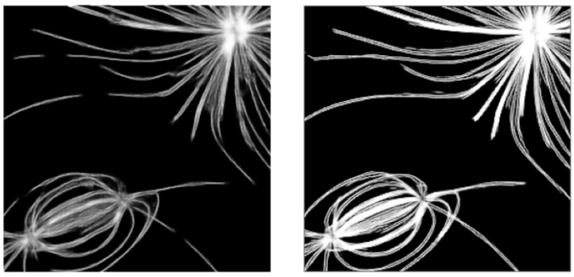
ing attention mechanisms into deep learning models for intricate pattern recognition tasks. Moreover, they emphasize the potential of the AtnDAE to make substantial contributions to our understanding of solar physics, offering a more reliable tool for analyzing coronal loops.

The comparative analyses showed the robustness of DAE models against varying conditions, including different bottleneck sizes and training data volumes, affirming its resilience to overfitting. The exploration of multiple loss functions has provided additional insights, guiding the selection of the most effective loss function for this specific task.

For future work, our aims include devising a neighborhood-aware loss function capable of addressing more intricate features and a systematic evaluation of synthetic data. Additionally, we seek to investigate the integration of our approach with other deep learning architectures, such as transformers. Furthermore, we aspire to expand the application of this methodology to tackle challenging segmentation tasks in various other fields of scientific interest.



(a) test image (b) U-Net prediction



(c) DAE prediction (d) AtnDAE prediction

Figure 12: Gap-filling by U-Net, DAE, and AtnDAE. Number of layers: 6. Bottleneck size: 32×32.

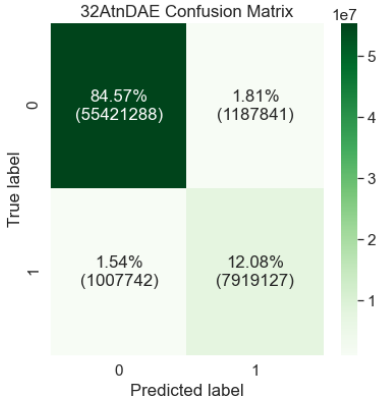
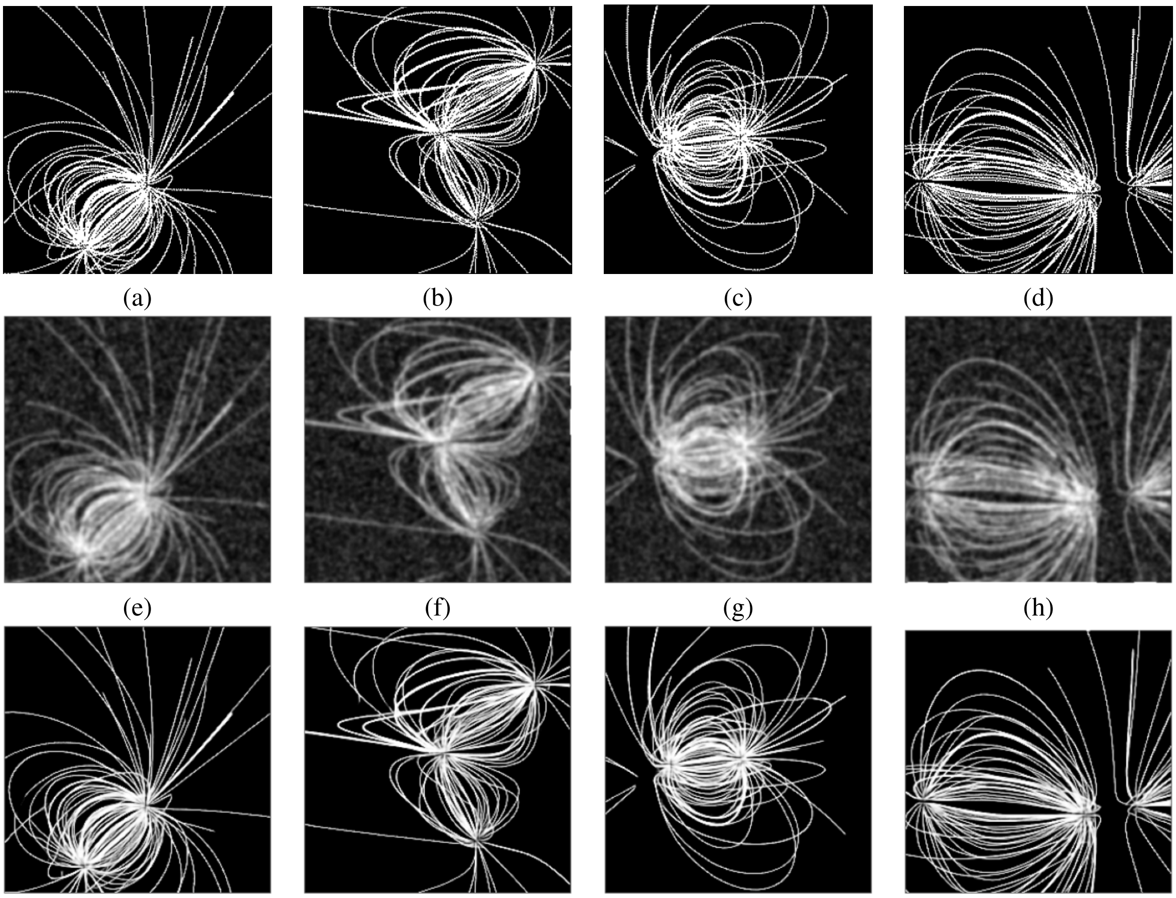


Figure 13: Confusion Matrices for the new AtnDAE on our dataset with the bottleneck size of 32×32.

6 ACKNOWLEDGMENTS

This research was supported in part by the David and Amy Fulton Endowed Professorship in Computer Science at Bowling Green State University. This work would not have been possible without the computing resources provided by the Ohio Supercomputer Center. We also thank the reviewers for their valuable comments which improved our paper.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 14: Segmentation results by AtnDAE: (a-d) ground truth, (e-h) test image, (i-l) AtnDAE predictions. Number of layers: 6. Bottleneck size: 32×32.

7 REFERENCES

[AnB19] Anwar, S., and Barnes, N., Real Image Denoising With Feature Attention, on Proc., IEEE/CVF Int'l Conf. on Computer Vision, October 2019, pp. 3155–3164.

[Asc10] Aschwanden, M.J., A Code for Automated Tracing of Coronal Loops Approaching Visual Perception, Solar Physics, Vol. 262, 2010, pp. 399–423

[BJP17] Benson, B., Jiang, Z., Pan, W.D., Gary, G.A., and Hu, Q., Determination of Linear Force-Free Magnetic Field Constant Alpha using Deep Learning, in Proc., 2017 Int‘l Conf. on Comp. Science and Comp. Intelligence, 2017, pp. 760–765.

[BKC17] Badrinarayanan, V., Kendall, A., and Cipolla, R., Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 39 (12). 2017, pp. 2481–2495.

[DNC09] Durak, N., Nasraoui, O., and Schmelz J., Coronal Loop Detection from Solar Images, Pattern Recognition, Vol. 42 (11), 2009, pp. 2481–2491.

[DNC10] Durak, N., Nasraoui, O., and Schmelz J., Automated Coronal-Loop Detection based on Cuntour Extraction and Contour Classification from the SOHO/EIT Images, Solar Physics, Vol. 264, 2010, pp. 383–402.

[FBH17] Fan, Z., Bi, D., He, L., Shiping, M., Gao, S., and Li, C., Low-level Structure Feature Extraction for Image Processing via Stacked Sparse Denoising Autoencoder, Neurocomputing, Vol. 243, 2017, pp. 12–20.

[Gon16] Gondara, L., Medical Image Denoising using Convolutional Denoising Autoencoders, in Proc., 2016 IEEE 16th Int‘l Conf. on Data Mining Workshops, 2016, pp. 241–246.

[GSY21] Guo, C., Szemenyei, M., Yi, Y., Wang, W., Chen, B., and Fan, C., Sa-UNet: Spatial Attention U-Net for Retinal Vessel Segmentation, in Proc., 25th Int'l Conf. on Pattern Recognition, 2021, pp. 1236–1242.

[HWF21] Hu, X., Wang, Y., Fuxin, L., Samaras, D., and Chen, C., Topology-Aware Segmentation using Discrete Morse Theory, arXiv preprint arXiv:2103.09992, 2021.

[HWH19] Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., and Liu, W., CCNet: Criss-Cross Attention for Semantic Segmentation, in Proc., IEEE/CVF Int'l Conf. on Computer Vision, 2019, pp. 603–612.

[Jad20] Jadon, S., A Survey of Loss Functions for Semantic Segmentation, in Proc., IEEE Conf. on Computational Intelligence in Bioinformatics and Computational Biology, 2020, pp. 1–7.

[KiB14] Kingma, D.P., and Ba, J., Adam: A Method for Stochastic Optimization, in Proc., 3rd Int'l Conf. for Learning Representations, 2014.

[LeT11] Lee, J.K., and Tang, W.K., Snake-based Technique for Automated Coronal Loop Segmentation, in Proc., 19th Int'l Conf. in Central Europe on Computer Graphics, Visualization, and Computer Vision, 2011, pp. 33–40

[LK16] Liskowski, P. and Krawiec, K., Segmenting Retinal Blood Vessels with Deep Neural Networks, IEEE Trans. on Medical Imaging, Vol. 35 (11), 2016, pp. 236–2380.

[LNG06a] Lee, J.K., Newman, T.S., and Gary, G.A., Oriented Connectivity-based Method for Segmenting Solar Loops, Pattern Recognition, Vol. 39 (2), 2006, pp. 246–259.

[LNG06b] Lee, J.K., Newman, T.S., and Gary, G.A., Dynamic Aperture-based Solar Loop Segmentation, in Proc., 2006 IEEE Southwest Symp. on Image Analy. & Interpret., Mar. 2006, pp. 91–94.

[MLT21] Moradi, S., Lee, J.K., and Tian, Q., Exploration of U-Net in Automated Solar Coronal Loop Segmentation, in Proc., 29th Int'l Conf. in Central Europe on Computer Graphics, Visualization, and Comp. Vision, 2021, pp. 227–235.

[MKA10] McAteer, R.T.J., Kestener, P., Arneodo, A., Khalil, A., Automated Dectecion of Coronal Loops Using a Wavelet Transformation Modulus Maxima Method, Solar Physics, Vol. 262, 2010, pp. 387–397.

[MZH20] Meng, Q., Zhang, J., Hu, Q., He, X., and Yu, J., LGNN: A Context-aware Line Segment Detector, in Proc., the 28th ACM Int'l Conf. on Multimedia, 2020, pp. 4364–4372.

[PWL18] Park, J., Woo, S., Lee, J.-Y., and Kweon, I, BAM: Bottleneck Attention Module, British Machine Vision Conf., Sept. 2018, pp. 1–14.

[RFB15] Ronneberger, O., Fischer, P., and Brox, T., U-Net: Convolutional Networks for Biomedical Image Segmentation, in Proc., Int'l Conf. on Medical Image Computing and Computer-assisted Intervention, 2015, pp. 234–241.

[SRS12] Singh, T.R., Roy, S., Singh, O.I., Sinam, T., Singh, K., and Singh, M., A New Local Adaptive Thresholding Technique in Binarization, arXiv preprint arXiv:1201.5227, 2012.

[TJF21] Tihon, S., Javaid, M.U., Fourure, D., Posocco, N., and Peel, T., DAEMA: Denoising Autoencoder with Mask Attention, in Proc., Int'l Conf. on Artificial Neural Networks, 2021, pp. 229–240.

[TME22] Theiner, J., Müller-Budack, E., and Ewerth, R., Interpretable Semantic Photo Geolocation, in Proc., IEEE/CVF Winter Conf. on Appl. of Comp. Vision, 2022, pp. 750–760.

[VLB08] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A., Extracting and Composing Robust Features with Denoising Autoencoders, in Proc., the 25th Int'l Conf. on Machine Learning, 2008, pp. 1096–1103.

[VLL10] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A., Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion, J. of Machine Learning Research, Vol. 11 (12), 2010, pp. 3371–3408.

[WPL18] Woo, S., Park, J., Lee, J.-Y., and Kweon, I.S., Cbam: Convolutional Block Attention Module, in Proc., European Conf. on Computer Vision, 2018, pp. 3–19.

[XiT15] Xie, S., and Tu, Z., Holistically-Nested Edge Detection, in Proc., IEEE Int'l Conf. on Computer Vision, 2015, pp. 1395–1403.

[XMY16] Xing, C., Ma, L., and Yang, X., Stacked Denoise Autoencoder based Feature Extraction and Classification for Hyperspectral Images, Journal of Sensors, 2016, pp. 1–10.

[YTS22] Yan, X., Tang, H., Sun, S., Ma, H., Kong, D., and Xie, X., After-UNet: Axial Fusion Transformer UNet for Medical Image Segmentation, in Proc., IEEE/CVF Winter Conf. on Appl. of Computer Vision, 2022, pp. 3971–3981.

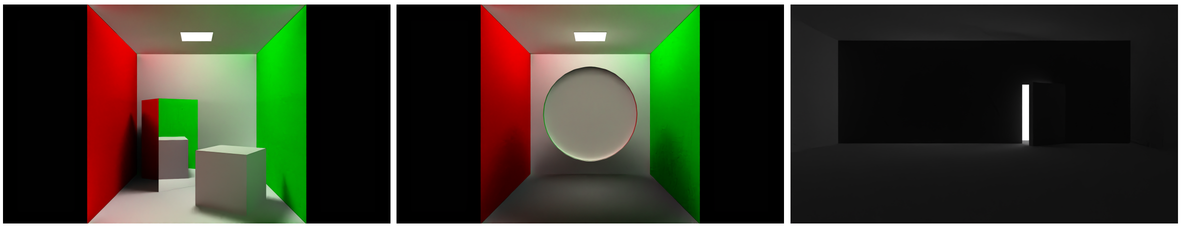
[ZXZ19] Zhiming, S., Xiaoli, Y., Zhongquan, Q., and Hong-Bo, L., Automatic Detection and Extraction Algorithm of Coronal Loops Based on Match Filter and Oriented Directivity, Monthly Notices of the Royal Astronomical Society, Vol. 490 (4), 2019, pp. 5567–5584.

Combining bidirectional path tracing, DDGI, and ReSTIR to improve real-time rendering quality

Viktor Kretzschmar
Hochschule Darmstadt
h_da
Schöfferstraße 3
64295, Darmstadt,
Germany
viktor.kretzschmar
@h-da.de

Prof. Dr. Benjamin Meyer
Hochschule Darmstadt
h_da
Schöfferstraße 3
64295, Darmstadt,
Germany
benjamin.meyer
@h-da.de

Prof. Dr. Elke Hergenröther
Hochschule Darmstadt
h_da
Schöfferstraße 3
64295, Darmstadt,
Germany
elke.hergenroether
@h-da.de



Three example images generated by the algorithm described in this paper. Each image shows soft shadows, indirect illumnination and refraction.

Abstract

This paper introduces a new algorithm for real-time rendering that builds upon bidirectional path tracing, reservoir-based spatio-temporal importance resampling, and dynamic diffuse global illumination. The combination of these algorithms produces an image with reduced noise compared to real-time run algorithms like path tracing, while retaining details like caustics. The resulting darkening, which is discussed in greater detail in this paper, is also reduced due to the usage of importance resampling of points on light-emitting surfaces. While the standard algorithms, such as bidirectional path tracing, cannot be run in real-time with satisfactory quality, a set of novel approaches have emerged to fill this gap. These algorithms are capable of running in real-time, although they do suffer from certain limitations. This paper describes the combination of bidirectional path tracing, DDGI and ReSTIR. This rectifies the drawbacks of missing indirect reflection, darkening and missing caustics of these algorithms. Ultimately, all the results of these algorithms are compared by verifying real-time rendering time and comparing quality to reference images. The quality is evaluated by using comparions for darkening and the similarity of the real-time rendered result to an offline path-traced result. The results of this paper demonstrate that the algorithm presented improves upon previous algorithms in terms of quality, while still maintaining real-time rendering constraints.

Keywords

real-time rendering, global illumination, ray tracing

1 INTRODUCTION

Over the past decade, ray tracing has moved from pre-rendered media to real-time applications. While hardware advances have made real-time ray tracing possible on consumer-grade hardware, real-time path tracing remains out of reach. However, path tracing offers several significant advantages over traditional ray tracing.

Path tracing produces more realistic images by incorporating indirect lighting and refraction without sampling the light sources themselves. As scenes get larger, the probability of selecting a visible light decreases, leading to darkness without any adjustment to brightness. Path tracing, on the other hand, is unaffected by darkening in larger scenes and maintains consistent illumination.

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.

Ideally, there would be an algorithm that supports direct illumination, indirect illumination and refraction, while still being able to run in real-time (i.e. $1/30$ seconds of render time per frame). This paper compares several approaches and presents a combined algorithm that supports both indirect illumination and refraction, providing a solution for darkening larger scenes while improving realism.

This paper uses terms such as ray tracing, path tracing and bidirectional path tracing. Both path tracing and bidirectional path tracing are implemented as described in [PJH16] using uniform direction sampling. Ray tracing is used as a short term for the algorithm presented in [Whi80].

2 RELATED WORK

This paper discusses three main techniques: Firstly, the bidirectional path tracer. Second, the improvements to ray tracing by Talbot *et al.* [Tal05] with reservoir-based spatio-temporal importance resampling (ReSTIR) by Bitterli *et al.* [BWP$^+$20]. And thirdly, improvements in global illumination by McGuire *et al.* [MMNL17] improved by Majercik *et al.* [MGNM19, MMK$^+$21, MMK$^+$22] with Dynamic Diffuse Global Illumination (DDGI). All these algorithms are the basis of the algorithm discussed in this paper and are compared with each other and the result of this paper.

ReSTIR is a relatively new algorithm developed in 2020. It is a method based on importance sampling that uses reservoirs to speed up the sampling process. It is based on the idea that importance sampling becomes more accurate the closer the light distribution is to the distribution from which samples are taken [Vea98]. The accuracy of the light distribution match can be improved by reusing information across space and time, also known as spatio-temporal reuse. This has been applied to bidirectional path tracing [GKDS12, Kel97, LW98, VG95, VG97], to path guidance [DK18, Jen95, LW95, MGN17, VKŠ$^+$14], and to resampling strategies [HKD14, KSKAC02, Tal05, VG97, KMA$^+$15]. While resampling is a good strategy, it causes significant overhead when implemented without an optimisation approach. For this reason, reservoir-based resampling has been used, [CHA82]. Based on this, ReSTIR was formulated by Bitterli *et al.* [BWP$^+$20].

DDGI is an algorithm that improves global illumination by adapting ray tracing with irradiance caching. Irradiance caching is a method of improving illumination by caching illumination data. It was introduced by Ward *et al.* [WRC88], and has been further developed by [A$^+$86, Hec90]. These implementations are statically rendered, i.e. they do not react to changing or moving geometry in the scene. In contrast to static lighting is dynamic lighting, which incorporates changes in the scene. Partially dynamic approaches have been implemented [GS12, RZD14, SL17, SSS$^+$20, SNRS12, SJJ12, VPG14]. DDGI by Majercik *et al.* [MGNM19, MMK$^+$21, MMK$^+$22] further develops these techniques based on McGuire *et al.* [MMNL17] to introduce fully dynamic rendering.

This paper builds on the Vulkan API to access the render time acceleration of graphics cards. The render-time acceleration is achieved through acceleration structures built by the graphics drivers themselves, which limits the use of improved techniques such as Multi-Level Hashed Grids [CPJ10], which can only be easily implemented by the vendor. Also, this work does not use any denoising or machine learning in the actual rendering process, such as CNN-enhanced Poisson Interpolation [ELM23], which could be used to further improve the resulting images.

3 DARKENING PROBLEM

Contrary to expectation, some rendering algorithms produce darker scenes when more light sources are added. This darkening effect particularly occurs in larger scenes where light sources are scattered amongst obstructive geometry. Algorithms that select a light source are prone to darkening, as they often encounter interrupted light paths in bigger scenes, casting shadows on objects. Larger scenes with more light sources increase the likelihood of sampling a light that doesn't contribute to the scene's brightness. This darkening effect is addressed by ReSTIR [BWP$^+$20], though not explicitly defined as such in their research. Notably, algorithms like path tracing are unaffected by darkening since they do not directly sample light.

Basic ray tracing is impacted by darkening. It traces rays from the camera, hitting geometry and connecting to light-emitting surfaces to determine brightness. While efficient for direct lighting, sampling light-emitting surfaces becomes problematic in larger scenes, as the probability of connecting to the relevant light sources decreases (see Figure 1).

Reservoir-based spatio-temporal importance resampling (ReSTIR) [BWP$^+$20] addresses the darkening issue in real-time rendering time. By prioritizing points on light-emitting surfaces based on hit probability, it ensures that non-contributing points are disregarded while important ones are more likely to be selected (see Figure 2). It also uses a step in which it disregards non connecting paths. This results in a uniformly lit scene.

To evaluate the darkening produced by each algorithm, two scenes are compared: one with a single Cornell box and another with five non-intersecting Cornell boxes. By positioning the camera to capture the same angle in both scenes, a pixel-by-pixel brightness comparison yields a percentage value indicating the degree of darkening. Values range from 20% (no darkening countermeasures) to 100% (completely darkening-free). Re-

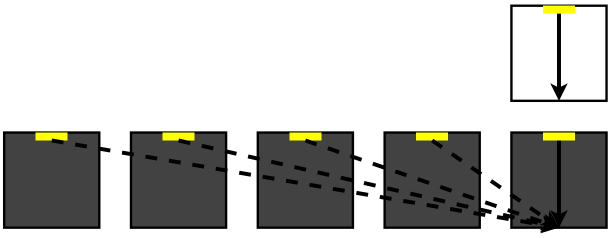


Figure 1: This illustration visualizes the darkening effect. In the first row, a single box with a light source is lit normally since all light points are unobstructed. In the second row, with five boxes, selecting a light from a different box in four out of five cases results in an 80% darker scene. Algorithms like ray tracing and bidirectional path tracing are impacted since they require selecting a scene light source.

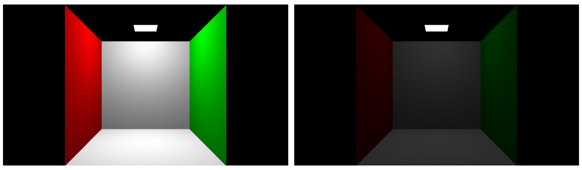


Figure 2: Comparison between two seemingly identical Cornell boxes. Both are rendered using ray tracing, but the right one is in a scene with five other Cornell boxes, resulting in a darker image.

sults and their implications are thoroughly analyzed in the Results section.

4 REFRACTION PROBLEM

Refraction is the bending of light when transported throu transparent materials. Depending on the refractive index of the material, the direction of the light ray is changed. For this to work, the entry and exit of a light ray must be processed. Just trying to connect two points in space will not work, because the object with the transparent material will be interpreted as an obstruction. Simple ray tracing only traces a ray from the camera to a point, and from that point to a point on a light-emitting surface. A transparent object will therefore cast a shadow. This is visualised, along with an example of its absence, in figure 3. This requires the light rays to be focused on a specific point. Rendering algorithms such as path tracing have a highter amount of paths traced from one pixel to a light source because of refraction, resulting in caustics. Both methods are computationally expensive, leading to a variety of algorithms that trade caustics for performance.

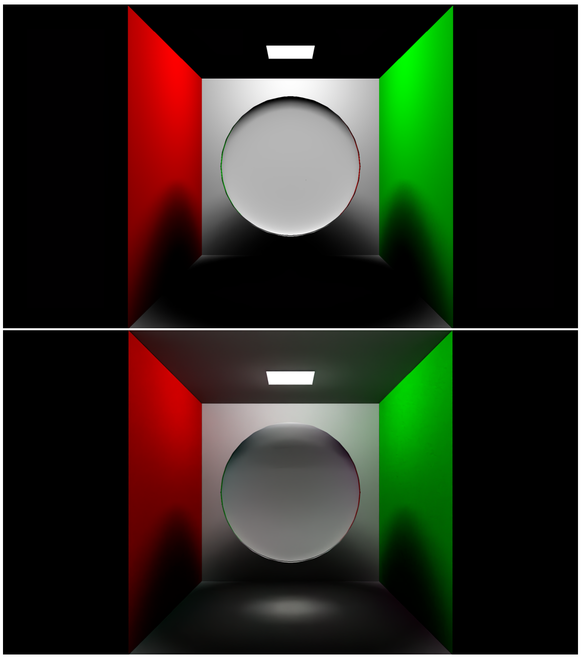


Figure 3: Both images contain a glass sphere that should refract the light coming from the lamp above. However, only the lower image has a visible focal point of light, while the upper image is completely dark below the sphere. The glass sphere ideally bends the light coming from the lamp and focuses it on the ground, creating caustics. The upper image is rendered using ray tracing, which does not simulate refraction, resulting in the dark shadow below the sphere. The lower image is rendered using bidirectional path tracing (without the constraints of real-time) and shows the light being focused by simulated refraction, creating a bright spot below the sphere.

In ray tracing, transparent materials and caustics cannot be simulated by either ReSTIR, DDGI or a combination of the two. ReSTIR only improves the sampling of points on light emitting surfaces; it does not change how rays are actually traced. As a result, hitting transparent geometry will still cast a shadow. Similarly, DDGI only adds an estimate for global illumination. It also does not change the way rays are traced, so transparent materials will still cast shadows. But these shadows are less dark, thanks to the global illumination estimate. There are still no caustics at the focal points.

Both path tracing and bidirectional path tracing support transparency and refraction, and thus simulate caustics. Their main problem is that they do not run in real-time for good quality, even on dedicated hardware. The ideal would be to combine the speed of ReSTIR and DDGI with the quality of (bidirectional) path tracing. This would ideally preserve their speed while providing results such as caustics.

5 COMBINING DIFFRENT ALGORITHMS

As discussed earlier, ReSTIR, DDGI and bidirectional path tracing all have specific advantages. ReSTIR improves the sampling of points on light emitting surfaces, resulting in improved light levels in larger scenes. DDGI provides a way to simulate global illumination

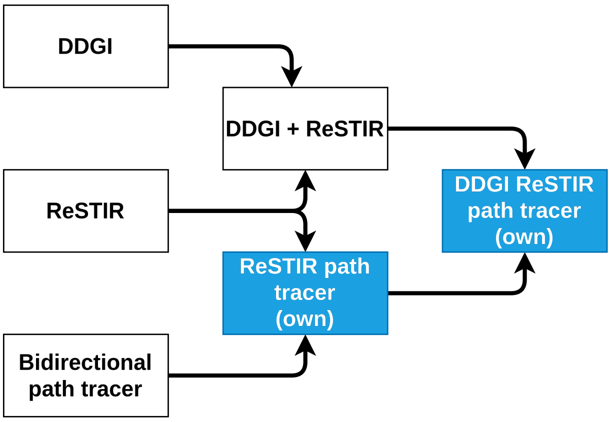


Figure 4: These are all the algorithms discussed in this paper and how they relate to each other. Two algorithms that point to a third show that they are combined into that algorithm. DDGI is from [MGNM19], ReSTIR is from [BWP+20] and the combination of the two is from [MMK+22]. Parts marked "own" with a blue background are new ideas from this paper.

without the performance penalty of path tracing. Bidirectional path tracing itself is a high quality, reliable algorithm that balances accuracy with speed. There are several ways to combine these algorithms. This paper is mainly concerned with the combination of all three algorithms, but also discusses their predictors. The relationship between the different algorithms can be seen in the figure 4.

ReSTIR improves ray tracing by preventing darkening (see section 3). This is done by introducing a sampling bias into the sampling of points on light emitting surfaces, giving each possible starting point for a ray a weight. DDGI adds the ability to estimate indirect lighting by building on top of ray tracing. It collects different light information by simply tracing rays, as is done when generating the actual image. Comparing the two, it can be seen that ReSTIR changes the start of the ray tracing operation, while DDGI changes the end. This allows them to be combined, which has already been done by Majercik et al. [MMK+22] As a result, DDGI, which previously suffered from darkening, now has a way to mitigate this.

Bidirectional path tracing is very similar to ray tracing. Both sample a point on a light-emitting surface, and both trace initial rays from the camera. As a result, both suffer from similar problems. Bidirectional path tracing, like ray tracing, has a problem with darkening. However, this is the result of using the same sampling algorithm for both algorithms. This makes it possible to use ReSTIR sampling in bidirectional path tracing, which reduces darkening. However, this adds noise to the image due to the still high processing cost, but inherits the ability to simulate transparency.

Finally, DDGI's Global Illumination Estimation can be added. DDGI's Global Illumination Estimation calculates an estimate based on ray tracing information previously collected by probes. These probes can then be used to calculate a specific background light level at any point in space. This is used in DDGI at the point where the traced ray hits the geometry, and the estimate is collected at the hit point. The estimate and the result of the ray tracing operation are added together to produce the final result. Because it is based on ray tracing, it inherits its limitations when it comes to transparency. Interestingly, ray tracing in DDGI can be replaced by bidirectional path tracing because it does not modify the ray itself. The only exception is the addition of the estimate.

Bidirectional path tracing starts from both a point on a light-emitting surface and a vision ray. Because the ray of sight is always the same for the same pixel, it can be reused, reducing the amount of computation required. The results of this ray are averaged. Coincidentally, the first hit of the vision ray is also the point at which DDGI injects its estimate. As a result, the estimate can simply be added to the result of the bidirectional path tracer at the first hit point of the vision ray. Adding the estimation adds more colour data to the overall image, smoothing out noise while preserving transparency and refraction. Adding ReSTIR also gives it the ability to mitigate darkening through improved sampling of points on light-emitting surfaces.

This combination of all three algorithms - bidirectional path tracing as the base, ReSTIR as an improved sampling of points on light emitting surfaces, and DDGI as an additional estimation of global illumination - is the central algorithm of this paper. The combination can be expressed as $L = L_{DDGI} + L_{BPT}(P_{ReSTIR})$. L describes the illumination, where L_{DDGI} describes the illumination given by the DDGI estimates. P_{ReSTIR} are the points on light emitting surfaces selected by ReSTIR and used by L_{BPT} as points from which to trace light rays. L_{BPT} stands for Bidirectional path tracing and results in a light value. Both light values are then added together for the final result, the same way DDGI handles the combination of direct illumination by ReSTIR and its own illumination estimation. This can lead to certain radiance contributens being accounted for multiple times, reasulting in a brighter image. The combination of all three algorithms still is an improvement in real-time rendering, as it is comparable to regular ray tracing in terms of performance, while providing caustics with reduced darkening and noise compared to pure bidirectional path tracing. A better breakdown of the results can be found in the next section.

6 RESULTS

The results are compared in four categories. Firstly, rendering times are discussed. These are the baseline to put all algorithms on an equal footing in the comparison by configuring them to be at least 30 FPS fast. Next, we discuss the darkening of the different algorithms. It shows which algorithms suffer more from darkening than others. The actual quality is then compared by comparing each image with a reference image from the same camera position and angle. The calculation compares each pixel on how different it is from the offline rendered reference. This results in values that can be compared across algorithms. Finally, some of the missing details are shown by providing images of certain details of the scene.

Rendering time is a critical test case for real-time rendering. Speeds of 30 or even 60 FPS are desirable. Therefore there is only a time window of 33.3 to 16.6 milliseconds for each frame. Figure 5 shows the rendering times. It shows lines for each algorithm and its rendering time per scene. Marker lines are also inserted to show the 30 and 60 FPS markers.

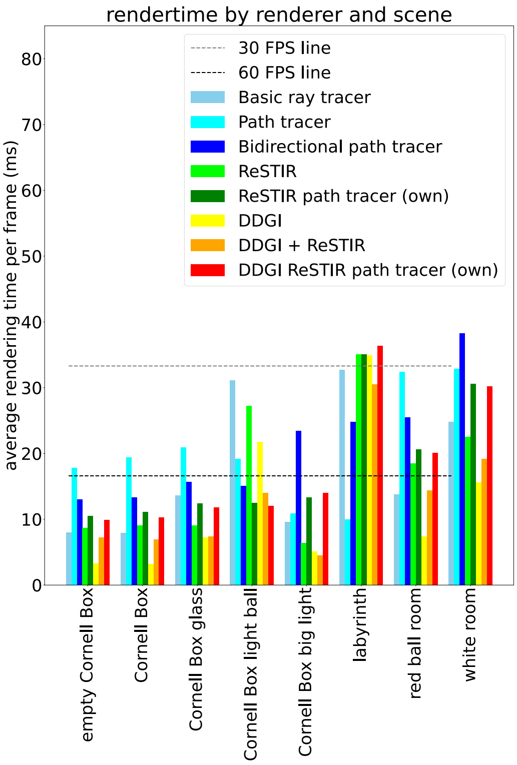


Figure 5: All algorithms and scenes and their respective render times. The two dotted lines also show the render time limits for 30 and 60 FPS.

Rendering algorithms have a number of parameters that represent a trade-off between performance and quality. Real-time rendering is constrained in terms of performance by the 30 FPS minimum. Consequently, the parameters of the algorithms have been chosen so that they do not fall below this minimum. This makes the algorithms comparable, as no algorithm can take advantage of excessive rendering times. Each algorithm has to meet the requirement to run in real-time. It is interesting to note that the most demanding scenes are the Cornell box with the light sphere, the maze and the white room. However, it is worth noting that some algorithms are better suited to certain scenes than others.

Darkening is the next problem to be discussed. As noted in section 3, scenes with more obstructed light-emitting surfaces are darker for some algorithms. The one and five Cornell box test is used to visualise this. This test is an original work of this paper. It tests darkening by comparing an image rendered in a scene with a single Cornell box with an image generated in a scene with five Cornell boxes. A correct result is an identical image or images with the same brightness when rendered with the same renderer. Both images are compared based on the average brightness preserved from the single to the five Cornell boxes as a relative value, as shown in figure 2. The brightness is calculated pixel by pixel by dividing the pixel brightness of the five Cornell box scene image by the pixel brightness of the single Cornell box scene image. All resulting values are averaged to give a value between $[0,1]$. Pixels where the brightness of the single Cornell box scene image is zero are ignored due to division by zero. A value of 100% means identical brightness, while a value of 20% would mean a reduction in brightness of 80%. 100% is ideal, i.e. no darkening. The calculated results are shown in the figure 6.

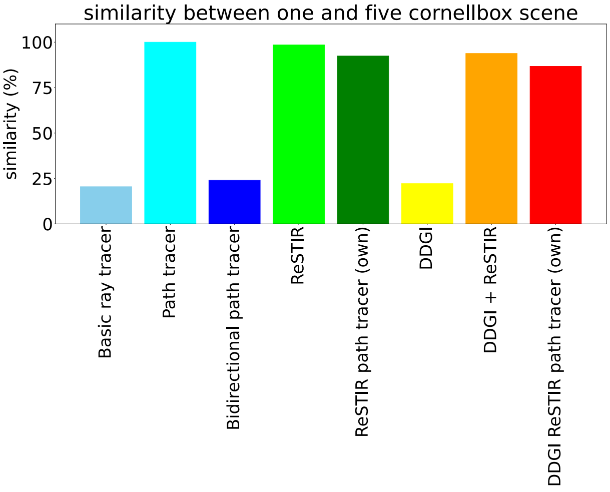


Figure 6: Brightness is maintained when comparing the rendering result between a single and five Cornell boxes. 100% is optimal, while in this case 20% is the worst case.

Both ray tracing and bidirectional path tracing have a darkening problem, whereas path tracing does not. This is because both ray tracing and bidirectional path tracing use a step where a light is sampled, whereas path tracing does not. If the scene contains multiple light-emitting surfaces, it becomes more likely that a sampled light will be obstructed, affecting the hit point of the vision ray, a problem that cannot be solved by increasing the number of samples taken because the probability, and hence the distribution of the resulting hits, does not change. path tracing, on the other hand, works perfectly because it does not have to worry about sampling points on light-emitting surfaces. It follows the ray until it hits a light-emitting surface or a specified maximum depth at which the ray stops. As a result, it is not influenced by other light-emitting surfaces in the scene. The results are identical because the rays hit the same geometry.

Of the other algorithms, only pure DDGI has a darkening problem. ReSTIR is a ray tracing algorithm with resolved darkening, so it is expected to have almost no darkening. As a result, all algorithms that use ReSTIR also have very low levels of darkening. It is important to note that as more algorithms are combined, the darkening increases slightly. The DDGI ReSTIR path tracer does not have as extreme a darkening as pure ray tracing, but it is still noticeable at around 10% in reduced image brightness. This concludes the darkening comparisons.

The quality is compared with reference images. A bidirectional path tracer without real-time rendering constraints is used to generate the reference images. The comparison between the result and the reference is made using the mean squre error method. The higher the value, the worse the resulting overall difference.

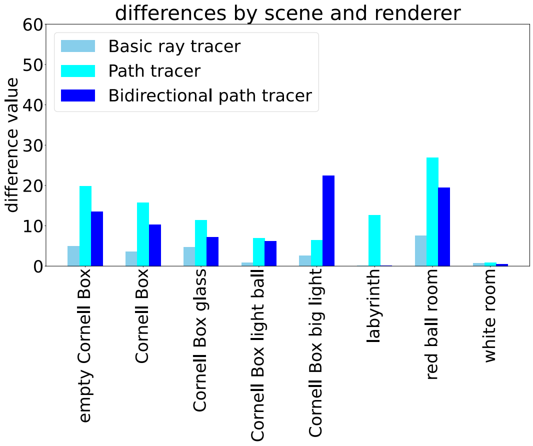


Figure 7: This figure compares the simple ray tracer, the regular path tracer and the bidirectional path tracer with mean square error. It can be seen that the ray tracer is the best of the three renderers in all cases. And the path tracer is worse than the bidirectional path tracer in almost all cases, which is to be expected.

The first comparison is made between the ray tracer, simple and bidirectional path tracers in the figure 7, which is done to show an example where the bidirectional path tracer has a higher quality than the simple path tracer in most cases. The improvement of the bidirectional path tracer is expected because the algorithm is faster than the regular path tracer, which means it can give a better result in real-time, which is the case in every scene except the Cornell Box big light. The immense size of the light-emitting surface favours simple path tracing. On the other hand, while the improvement in all the other scenes is quite significant, the difference in the Labyrinth scene is as good as the raytracer's results. The ray tracer, which is faster than both path tracers, can achieve better results than both because of the speed improvement. The simplification of the traced path results in a speed improvement because the path contains only one bounce. These three compared algorithms are just to show that the comparison method works. It shows the expected results by showing that the bidirectional path tracer has a higher quality than the regular path tracer.

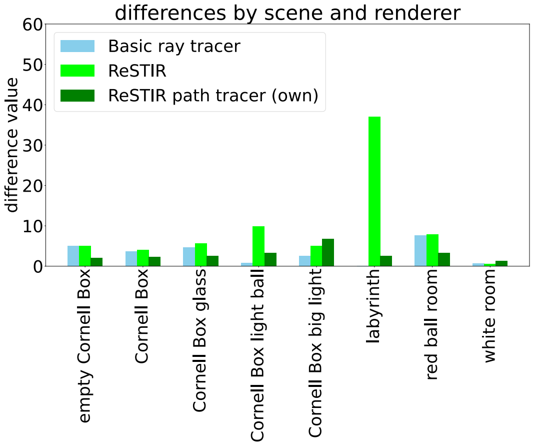


Figure 8: Comparison of the basic raytracer with ReSTIR and the bidirectional path tracer combined with ReSTIR, called the ReSTIR path tracer. It can be seen that the raytracer is better than the other two algorithms, while the other two are better or worse than each other depending on the scene.

The ray tracer, ReSTIR and the ReSTIR path tracer are compared with the chart in figure 8. The simple ray tracer beats the other algorithms in terms of quality, sometimes by a small margin, as in the case of the empty Cornell Box scene, or by a large margin, as in the case of the Cornell Box scene with large lights. The other two algorithms alternate in quality. As always, the Maze scene is the standout, with ReSTIR looking the worst by a wide margin. To sum up, the basic raytracer has the highest quality of the three algorithms when only looking at this statistic. If you take into account the darkening seen in the previous section, you can make an argument for the ReSTIR raytracer. It has almost no darkening, like standard ReSTIR, but has better quality in half the cases, but never really worse compared to ReSTIR. The basic raytracer has a darkening problem, which makes it generally worse. But more algorithms can be compared.

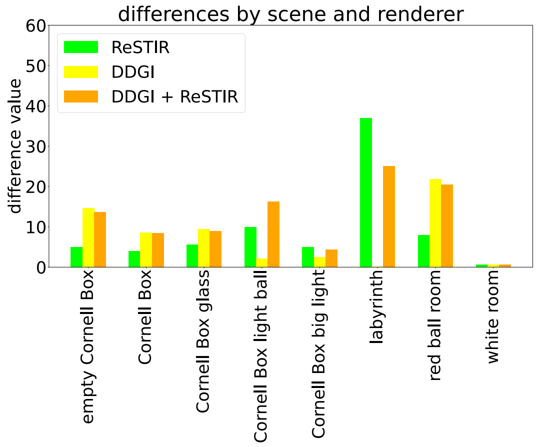


Figure 9: The comparative results for the ReSTIR and DDGI renderers and their combination. It is interesting to note that some of these algorithms alternate in quality depending on the scene. The quality changes with the scenes, but the overall quality is similar.

Next is a comparison between ReSTIR, DDGI and ReSTIR with DDGI, as shown in the figure 9. It depends very much on the scene to say which algorithm is the best and which is the worst. For example, the ReSTIR algorithm is the best for the empty Cornell box scene, but the worst for the maze scene. DDGI is the worst for the empty Cornell Box scene, but the best for the maze, by an impressively wide margin. You would expect ReSTIR and DDGI together to be in the middle, but in half of the scenes it is the worst algorithm. Sometimes this is a small margin, as in the Cornell Box scene, but other times it is a significant margin, as in the Cornell Box big light. The result is a heterogeneous mix in which no algorithm is truly the best.

In the final comparison, DDGI, DDGI with ReSTIR and the bidirectional path tracer with DDGI and ReSTIR are discussed. The results are shown in Figure 10. DDGI and DDGI with ReSTIR are either very similar, with DDGI being slightly worse in some cases, or DDGI is massively better. While this is interesting, the critical part is the DDGI ReSTIR path tracer, which is the best algorithm in half of the scenes, and the worst by a small margin in two scenes.

Figure 11 is included to show the full comparison between all algorithms. It shows all the algorithms side by side, with the final algorithm of this paper, the DDGI ReSTIR raytracer, being one of the better algorithms, being the best in two cases and otherwise having a low error value. The ray tracer and DDGI are both better in some cases, but have a darkening problem which makes them generally worse. In conclusion, the results show that the combination of the bidirectional raytracer with DDGI and ReSTIR gives the best overall quality.

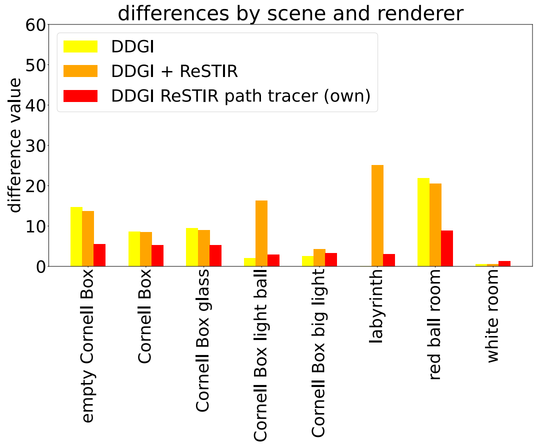


Figure 10: All three different DDGI-based algorithms were compared. The DDGI ReSTIR path tracer is an original algorithm that combines DDGI, ReSTIR and bidirectional path tracing. It is one of the best algorithms in this comparison for most scenes.

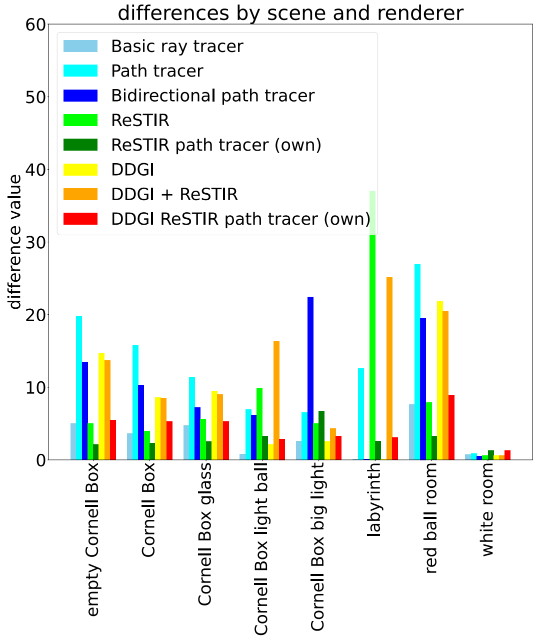


Figure 11: All quality comparison results are shown in the figure.

Some details are not quantified in this paper, but should be shown for completeness, mainly concerning caustics, for which the scene Cornell Box glass is used. The light is concentrated at the bottom, resulting in a bright spot. A visual comparison between different algorithms can be seen in figure 12. The images were denoised with the Intel ODIN denoiser, as were the other images in this paper. The visual comparison focuses on the Cornell Box Glass scene and shows both the indirect illumination and the caustics for comparison.

The reference image is bright but casts dark shadows. Especially the ceiling and under the glass sphere are very dark. There are also caustics on the floor, resulting in a bright spot. These caustics are not visible in the next two renderers, the raytracer and the DDGI with ReSTIR. Both render dark shadows. But because of its focus on global illumination, DDGI with ReSTIR has a non-black ceiling. It is slightly brighter than the reference and has similar color values. Finally, the final result of this paper, the DDGI ReSTIR path tracer, has caustics. It has a slight but noticeable bright spot on the floor. The ceiling is also illuminated, but brighter than in the other images. As a result of the visual inspection, the solution of this paper still needs improvement, but it produces good results. It shows potential in the area of caustics and global illumination, but is noisy in its raw form. This noise is present on both the ceiling and the floor.

7 CONCLUSION

In conclusion, this paper presents a novel algorithm usable in real-time rendering that simulates both indirect illumination and refraction. This algorithm is a combination of three algorithms: Bidirectional path tracing, DDGI and ReSTIR. All of these algorithms have their advantages and disadvantages, and thus their combination rectifies these to some extent. They have been compared in both quality and speed to see their advantages and disadvantages. But this paper also showed that the new algorithm of this paper is the best overall algorithm when compared in quality and details like caustics.

While the results of this new algorithm improve on its predecessors, it also inherits their limitations. The use of bidirectional path tracing in real-time rendering has always resulted in noisy results. This algorithm is no exception, even if the noise is reduced by the other algorithms. While ReSTIR solves the darkening problem, there is still some darkening when combined with other algorithms, not much, but still measurable. DDGI is an approximation algorithm, and as such does not give correct results, but better results than the other algorithms discussed in this paper, which also affects the final algorithm.

These algorithms mostly rely on bidirectional path tracing to do the heavy lifting of the algorithm. As a result, the rendering result is noisy. While DDGI compensates for this to some extent, highlights such as caustics are particularly noisy, leaving room for optimization.

Neural denoising and neural sampling Neural denoising and neural sampling were developed after the advent of deep learning. Both have been applied to real-time rendering applications [HMS+20]. These denoising and sampling techniques could improve the results by removing noise or improving ReSTIR based sampling.

Caching of paths The algorithm described in this paper is based on bidirectional path tracing. As such, multiple paths are constructed through the scene. These paths could be reused in later frames to improve rendering quality, which has been done based on ReSTIR with regular path tracing [OLK+21]. The application of ReSTIR-based reuse in world space [Boi21] could also be applied to this algorithm for further improvement.

8 REFERENCES

[A+86] James Arvo et al. Backward ray tracing. *Developments in ray tracing*, 12:259–263, 1986.

[Boi21] Guillaume Boissé. World-space spatiotemporal reservoir reuse for ray-traced global illumination. In *SIGGRAPH Asia 2021 Technical Communications*, SA '21 Technical Communications, New York, NY, USA, 2021. Association for Computing Machinery.

[BWP+20] Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph.*, 39(4), July 2020.

[CHA82] M. T. CHAO. A general purpose unequal probability sampling plan. *Biometrika*, 69(3):653–656, 12 1982.

[CPJ10] Vasco Costa, Joao Pereira, and Joaquim Jorge. Multi-level hashed grids for ray tracing. *WSCG*, 2010.

[DK18] Ken Dahm and Alexander Keller. Learning light transport the reinforced way. In Art B. Owen and Peter W. Glynn, editors, *Monte Carlo and Quasi-Monte Carlo Methods*, pages 181–195, Cham, 2018. Springer International Publishing.

[ELM23] Blaž Erzar, Žiga Lesar, and Matija Marolt. Fast incremental image reconstruction with cnn-enhanced poisson interpolation. *Václav Skala-UNION Agency*, 2023.

[GKDS12] Iliyan Georgiev, Jaroslav Křivánek, Tomáš Davidovič, and Philipp Slusallek. Light transport simulation with vertex connection and merging. *ACM Trans. Graph.*, 31(6), nov 2012.

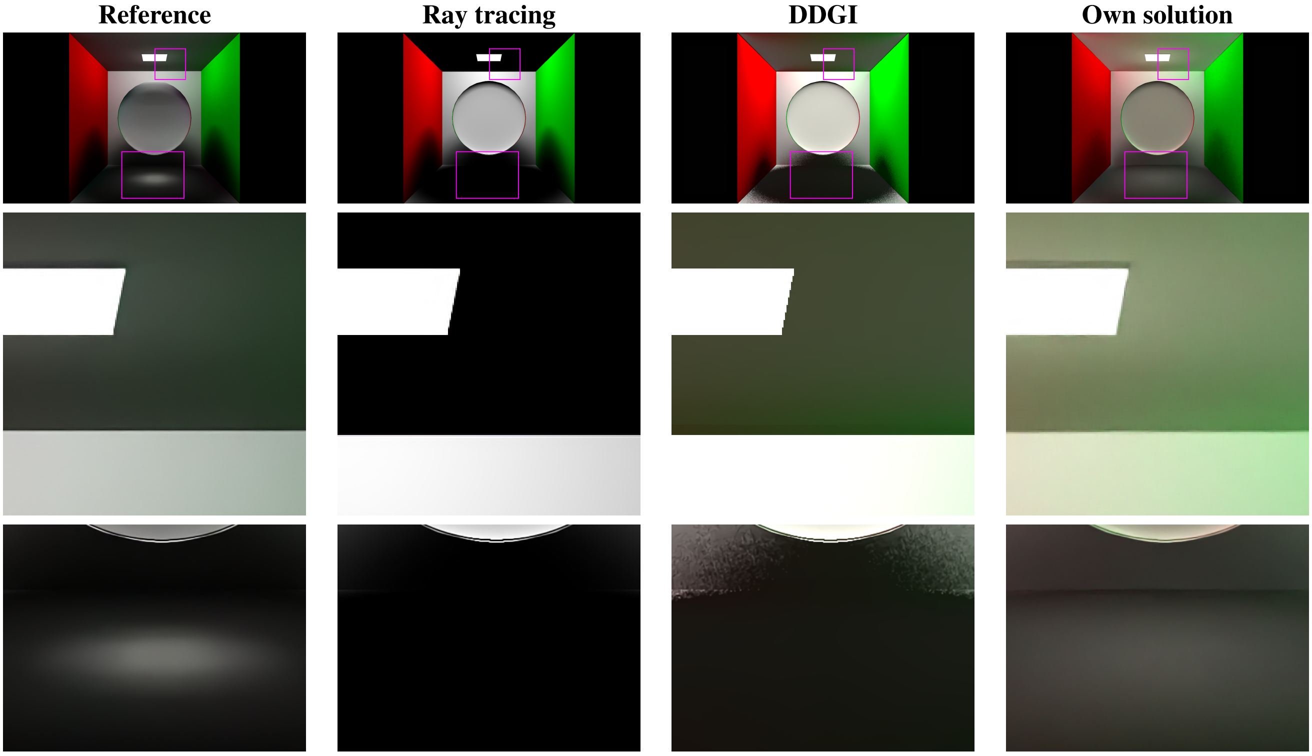


Figure 12: The results of rendering the Cornell Box Glass scene in different renderers for visual comparison. From left to right: Bidirectional path tracing offline reference, the raytracer, DDGI with ReSTIR, and the final result. The magenta boxes are enlarged in the bottom two images below the top images. These images show (the lack of) indirect illumination in the middle image and caustics in the bottom image.

[GS12] NS Mickael Gilabert and Nikolay Stefanov. Deferred radiance transfer volumes, global illumination in far cry 3. In *Game Developers Conference*, volume 2, 2012.

[Hec90] Paul S Heckbert. Adaptive radiosity textures for bidirectional ray tracing. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 145–154, 1990.

[HKD14] Toshiya Hachisuka, Anton S. Kaplanyan, and Carsten Dachsbacher. Multiplexed metropolis light transport. *ACM Trans. Graph.*, 33(4), jul 2014.

[HMS+20] J. Hasselgren, J. Munkberg, M. Salvi, A. Patney, and A. Lefohn. Neural temporal adaptive sampling and denoising. *Computer Graphics Forum*, 39(2):147–155, 2020.

[Jen95] Henrik Wann Jensen. Importance driven path tracing using the photon map. In Patrick M. Hanrahan and Werner Purgathofer, editors, *Rendering Techniques '95*, pages 326–335, Vienna, 1995. Springer Vienna.

[Kel97] A Keller. Instant radiosity, siggraph 97 proceedings, 1997.

[KMA+15] Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. Gradient-domain path tracing. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, 2015.

[KSKAC02] Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum*, volume 21, pages 531–540. Wiley Online Library, 2002.

[LW95] Eric P Lafortune and Yves D Willems. A 5d tree to reduce the variance of monte carlo ray tracing. In *Rendering Techniques' 95: Proceedings of the Eurographics Workshop in Dublin, Ireland, June 12–14, 1995 6*, pages 11–20. Springer, 1995.

[LW98] Eric Lafortune and Yves Willems. Bidirectional path tracing. *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics'*, 93, 01 1998.

[MGN17] Thomas Müller, Markus Gross, and Jan Novák. Practical path guiding for efficient light-transport simulation. In *Computer Graphics Forum*, volume 36, pages 91–100. Wiley Online Library, 2017.

[MGNM19] Zander Majercik, Jean-Philippe Guertin, Derek Nowrouzezahrai, and Morgan McGuire. Dynamic diffuse global illumination with ray-traced irradiance fields. *Journal of Computer Graphics Techniques (JCGT)*, 8(2):1–30, June 2019.

[MMK+21] Zander Majercik, Thomas Müller, Alex Keller, Derek Nowrouzezahrai, and Morgan McGuire. Dynamic diffuse global illumination resampling. *Computer Graphics Forum*, page 13, Dec 2021.

[MMK+22] Zander Majercik, Thomas Müller, Alexander Keller, Derek Nowrouzezahrai, and Morgan McGuire. Dynamic diffuse global illumination resampling. *Computer Graphics Forum*, 41(1):158–171, 2022.

[MMNL17] Morgan McGuire, Michael Mara, Derek Nowrouzezahrai, and David Luebke. Real-time global illumination using precomputed light field probes. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, page 11, February 2017. I3D 2017.

[OLK+21] Y. Ouyang, S. Liu, M. Kettunen, M. Pharr, and J. Pantaleoni. Restir gi: Path resampling for real-time path tracing. *Computer Graphics Forum*, 40(8):17–29, 2021.

[PJH16] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation 3rd Edition*. Elsevier, November 2016.

[RZD14] Hauke Rehfeld, Tobias Zirr, and Carsten Dachsbacher. Clustered pre-convolved radiance caching. In *EGPGV EuroVis*, pages 25–32, 2014.

[SJJ12] Jorge Schwarzhaupt, Henrik Wann Jensen, and Wojciech Jarosz. Practical hessian-based error control for irradiance caching. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

[SL17] Ari Silvennoinen and Jaakko Lehtinen. Real-time global illumination by precomputed local reconstruction from sparse radiance probes. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017.

[SNRS12] Daniel Scherzer, Chuong H Nguyen, Tobias Ritschel, and Hans-Peter Seidel. Preconvolved radiance caching. In *Computer Graphics Forum*, volume 31, pages 1391–1397. Wiley Online Library, 2012.

[SSS+20] Dario Seyb, Peter-Pike Sloan, Ari Silvennoinen, Michał Iwanicki, and Wojciech Jarosz. The design and evolution of the uberbake light baking system. *ACM Transactions on Graphics (TOG)*, 39(4):150–1, 2020.

[Tal05] Justin F Talbot. *Importance resampling for global illumination*. Brigham Young University, 2005.

[Vea98] Eric Veach. *Robust Monte Carlo methods for light transport simulation*. Stanford University, 1998.

[VG95] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques*, pages 145–167. Springer, 1995.

[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, pages 65–76, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[VKŠ+14] Jiří Vorba, Ondřej Karlík, Martin Šik, Tobias Ritschel, and Jaroslav Křivánek. On-line learning of parametric mixture models for light transport simulation. *ACM Transactions on Graphics (TOG)*, 33(4):1–11, 2014.

[VPG14] K Vardis, G Papaioannou, and A Gkaravelis. Real-time radiance caching using chrominance compression. *Journal of Computer Graphics Techniques Vol*, 3(4), 2014.

[Whi80] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, jun 1980.

[WRC88] Gregory J Ward, Francis M Rubinstein, and Robert D Clear. A ray tracing solution for diffuse interreflection. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 85–92, 1988.

Empathy Training using Virtual Environments

Ron Jackson
University of Colorado
Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80907
rjackso7@uccs.edu

Sudhanshu Kumar Semwal
University of Colorado
Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80907
ssemwal@uccs.edu

ABSTRACT

We developed a virtual environment (VE) for nursing students so that they can experience what a person living with schizophrenia constantly hears. In our implementation, Non Player Character (NPC) Eva interacts with the player by recognizing the facial expressions of the players wearing Oculus Rift-S head mounted display (HMD). We use the Unity game development platform, and implement machine learning (ML) algorithms using deep learning (DL) models to provide such simulated experiences. In our implementation, the NPC Eva recognizes the player's facial expressions and reacts with a variety of facial, body, and verbal animated responses. Our empathy training virtual environment is developed for the nursing students. Our colleagues at the College of Nursing have also undertaken an approved IRB (Internal Review Board) user study. This paper focuses on technical details of our algorithms, their implementations. Main results of our research are summarized, including a positive reception of our empathy training virtual environment.

Keywords

Computer vision, facial expression recognition, neural networks, virtual reality, visual interaction, Python, Unity3D(TM)

1 INTRODUCTION

Our goal is to create a first person Virtual Environment (VE) providing a first person perspective, enabling a participant to experience auditory hallucinations [Deegan2022] a person living with schizophrenia might hear. Our VE provides an ability to move through, explore, and interact with places, objects and non-player characters (NPCs) to accomplish tasks that are consistent with the designed theme of the Nursing Curriculum. Our VE asks players (i.e. Nursings students in our case) to locate objects and interact with NPCs while visiting common VE locations such as a street-side coffee kiosk. NPCs interact with players using scripted behavioral responses.

Empathy is an essential skill in the nurse client relationship which can be trained and practiced [Ward2016, Ward2012, Ward2009] Even though empathy has been recognized as an important skill set, [Ward2016, Ward2012] report a decline in empathy in undergraduate nursing students. This may be attributed to time constraints in developing quality relationships with patients emphasizing the critical nature of empathetic interactions [Ward2009]. Our motivation to develop a VE is to provide an immersive VR training environment which can be an effective means of developing nursing students' empathy towards persons with schizophrenia as nursing students can also also experience the same auditory hallucination using our VE at any time of their convenience. Our implementation also addresses the partial occlusion of the user's face by the Oculus Rift S headmounted display (HMD) by providing a facial expression recognition (FER) DL model trained initially on full facial image, then refined by using only lower half facial images during final training by adding lower half of images avoiding upper half of these images which are covered by a VR headset in our case. We use Python-based FER application server to continuously communicate with the Unity VR platform client. The server uses an efficient residual network (ResNet) convolutional neural network (CNN) design which is trained to recognize five facial expressions. We used the Facial Expression Recognition Challenge 2013 (FER2013) data set [Carrier2013], Real-world Affective Faces (RAF) database, and some supplemental user images to train and test the CNN model [Jackson2022]. The resulting trained CNN provides 81% recognition accuracy across five different facial expressions on images with an occluded upper half face, providing

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.

https://www.doi.org/10.24132/CSRN.3401.10

us a basis to show that our basic hypothesis works. In future, this recognition rate could be improved by adding more training data set. Figure 1 shows our basic setup. Additionally, the server-to-client inter-application communication average response time measured from client request to client receipt of server facial expression response is less than 0.25 seconds based on experimental observations (Figure 2). This responsiveness helps ensure timely and smooth initiation of NPC animation actions in our VE. The next sections of this paper discuss related work, design and implementation of our techniques. Eyes and eyebrows and folding features of our face are not visible due to VR-Headset (Figure 1) so the main technical challenge which we have addressed in this paper is how facial expressions can be computationally recognized by an NPC using only the bottom half of player's facial features as the player (nursing students) with headset would interacts with NPC. We implemented a VR scavenger hunt which replaces a real-life scavenger hunt experience designed in Nursing curriculum for all Nursing Students. Our system, which has undergone limited University approved IRB user study, is mainly geared towards recognizing player's facial expressions by an NPC to facilitate empathetic interactions. This may serve as part of online curriculum option for Nursing Students at our university, with the hope of developing empathy towards a person experiencing constantly denigrating and disturbing voices.

2 RELATED WORK AND MOTIVATION

We define empathy in the context of the nurse-patient relationship as predominantly a cognitive, rather than emotional, attribute that involves the ability to understand, rather than feel, experiences, concerns, and perspectives of the patient, combined with a capacity to communicate this understanding [Jackson2022]. Based on interactions with Nursing Faculty, it was thought that Nursing students would develop empathetic behaviors using our Virtual Environement where we can design storyline and scenarios where Nursing student player can experiences relatable experiences, such as hearing voices which a person with schizophrenia constantly experiences [Ward2012]. As clinical experiences are limited and may not present opportunities for real-life interactions with a person living with schizophrenia, our virtual environment can provide an online option for training Nursing students to experience relatable experiences. Schizophrenia is a serious mental wellness issue that interferes with a person's ability to think clearly, manage emotions, make decisions, and relate to others [Jackson2022]. It is a complex, long-term mental wellness issue, affecting about one percent of the population. Although schizophrenia can occur at any age, the average age of a person diagnosed with schizophrenia, and its onset, tends to be in the late teens to the early twenties for men, and the late twenties to early thirties for women. It is uncommon that schizophrenia is diagnosed younger than 12 or older than 40 years of age. The symptoms of schizophrenia fall into three categories: positive, negative, and cognitive [Jackson2022]. Positive symptoms are psychotic behaviors not generally seen in general population. People with positive symptoms may lose touch with some aspects of reality. Symptoms include hallucinations, delusions, thought disorders (unusual or dysfunctional ways of thinking), movement disorders (agitated body movements) [Jackson2022]. Negative symptoms are those that are associated with disruptions of normal emotions and behaviors, and include flattening affect such as reduced expression of emotions via facial expression or voice tone; reduced feelings of pleasure in everyday life, difficulty beginning and sustaining activities, and reduced speaking. Cognitive symptoms of schizophrenia can be subtle for some, but are more severe for others, and patients may notice changes in their memory or other aspects of thinking. Symptoms include possible poor decision making due to changes in the ability to understand information and use it to make decisions, trouble focusing or paying attention, and problems with working memory such as the ability to use the information immediately after learning it.

Virtual Reality based environments can be particularly valuable to develop critical skills and enhance the cognitive understanding when the same firsthand real world experiences are not available to experience, or are not safe [Jackson2022] as some nursing students may have apprehension towards working directly with a person with schizophrenia lacking training and skill set. Our Virtual environment may help a nursing student develop cognitive maps and experiences while training in a safe Virtual environment. Virtual environments can enhance such experiences by developing cognitive maps so that real-life experience of others can be experiences by augmenting a VE with hearing voices which a player can experience. As part of their course curriculum, the College of Nursing uses an audio recording of Hearing Voices that are Distressing [Deegan2022], and conduct a workshop to train students while they are asked to perform real world in-person tasks at different stations. The nursing students are our intended target as players in our Virtual Environment. Existing College of Nursing curriculum involvs listening to the hearing voices recording through audio earphones throughout the training session by the nursing students. During Scavenger hunt training, nursing students visiting different stations to finding objects, reading an article, taking a quiz, and solving crossword puzzles, all the time listening to the distressing hearing voices [Deegan2022].

Our immersive VR training environment provides similar training experiences of a scavenger hunt, for example, while immersing the user in the hearing voices audio recording. Cognitive maps [Tolman1948] are a series of psychological transformations that allow an individual to acquire, code, store, recall, and decode information about their spatial environments by developing an understanding of relative locations and attributes of the phenomena in their spatial environment [Tolman1948, Jackson2022]. [Deegan2022] hearing voices simulation is key component of our Virtual Environment and is used to provide an opportunity to develop a sense of empathy for those who suffer with schizophrenia. Recent VR research indicates the most important user experience mechanisms are illusion of virtual body ownership (user perception that the virtual body is their own), and agency (user attribution of actions in the VE as their own) [Barbot2020].

3 METHODOLOGY AND IMPLEMENTATION DETAILS

We added a visual interaction mode between the user and primary NPC (Eva) in our VE enabling the possibility of new user experiences. Figure 2 shows overall flow in implemented Virtual Environment. NPC-Eva's has the computational ability to see and recognize the approaching user's facial expression in one of five categories– angry, disgust, sad, happy, surprise– and then react with unique scripted animated facial, body and verbal responses. This allows us to simulate both storylines as planned – (i) user-initiated interactions, such as approaching or speaking, with Eva while getting in line at a Cofeee-kiosk with other NPCs, and (ii) NPC-Eva initiated interactions as NPC-Eva can see and respond to participant's facial expressions as the player approaches the coffee kiosk in our planned interactions (Figure 2). Eva's Faecial Expression Recognizer (FER) is a Python-based facial expression (FE) prediction server operating on player's (i.e. nursing student or one of the co-authors) facial image frames captured by a web camera. The server runs continuously and provides Facial Expression (FE) prediction replies to the Unity SE application client requests as the user initially approaches Eva at the coffee kiosk (See Figure 1). The visual interaction mode was the result of three main developments and provides the following functions:

- A custom frontal aspect detector trained to find and segment the user's HMD-occluded face is activated and defines a box bounding the user's face. The detector is developed using the DLib toolkit for image processing and ML tools [Dlib2021]. The custom detector model is a Support Vector Machine (SVM) using Histogram of Oriented Gradients (HOG) features trained on images collected of the author wearing the HMD.
- A ResNet CNN FER prediction model trained on five facial expressions (angry, disgust, sad, happy, surprise) given inputs that are HMD-occluded facial image frames predicts the user's expression (See Figure 2). The prediction model is a ResNet CNN developed using combined training images (16,329 total) from the Real-world Affective Faces Database (RAF) and the Facial Expression Recognition Challenge 2013 (FER2013) data sets [Carrier2013], plus a small set of supplemental training facial images (989 total) of the author. This prediction model performs recognition using only the lower-half of facial images and achieves 81% accuracy.
- A real-time message interface between the Python FE prediction model and the Unity SE application receives requests and provides predicted facial expression replies. The message interface uses ZeroMQ, which is an open-source messaging library supported in Python and by through NetMQ, which is a native port of the ZeroMQ [ZeroMQ2021] library. The messaging pattern used is Request/Reply, where the Unity SE application is the requesting client and the Python FE prediction script is the server providing string type facial expression replies whenever it receives a request. Timing data observed during experimentation shows that the total time between FER server replies when the SE application is making continuous requests typically ranges between 0.20 and 0.25 seconds.

The VE automatically sets Eva's response mode– positive or negative – based on NPC Eva's perception of the user's expression simulating relatable experiences [Jackson2022, Deegan2022]. Angry and disgust generate negative reactions, while sad, happy and surprise generate positive reactions. Figure 2 shows the overall functional flow of our Virtual Environment.

4 DEVELOPMENT AND COMPUTING PLATFORMS

The task pipelines for these capabilities, which are all implemented in Python version 3.7, are shown in Figure 3. The Unity and C development environments for this project are Unity version 2019.2.19f1 and Visual Studio Community 2019 version 16.9.3, respectively. The Oculus Application is version 28.0.0.222.469, using an Oculus Rift S HMD (Firmware version 2.2.0) and Touch Controllers (Firmware version 1.14.2).

4.1 HMD-occluded Facial Detection

The Python integrated development environment (IDE) for this project is Spyder 4.2.1 configured for Tensor-

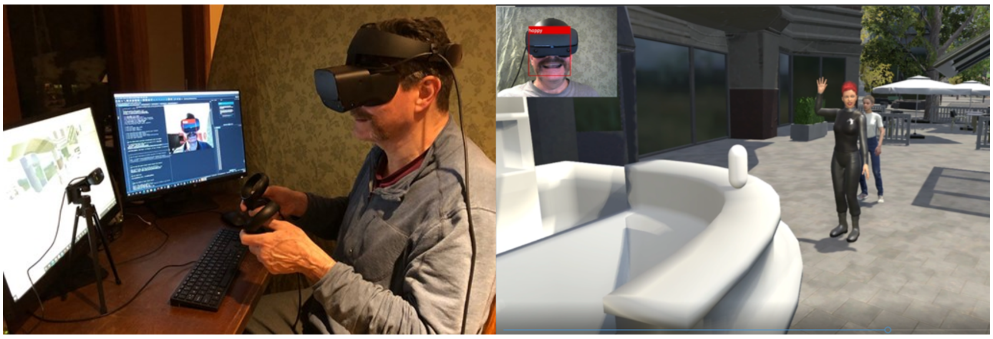


Figure 1: Interaction set up and Eva's recognition of user facial expression and response

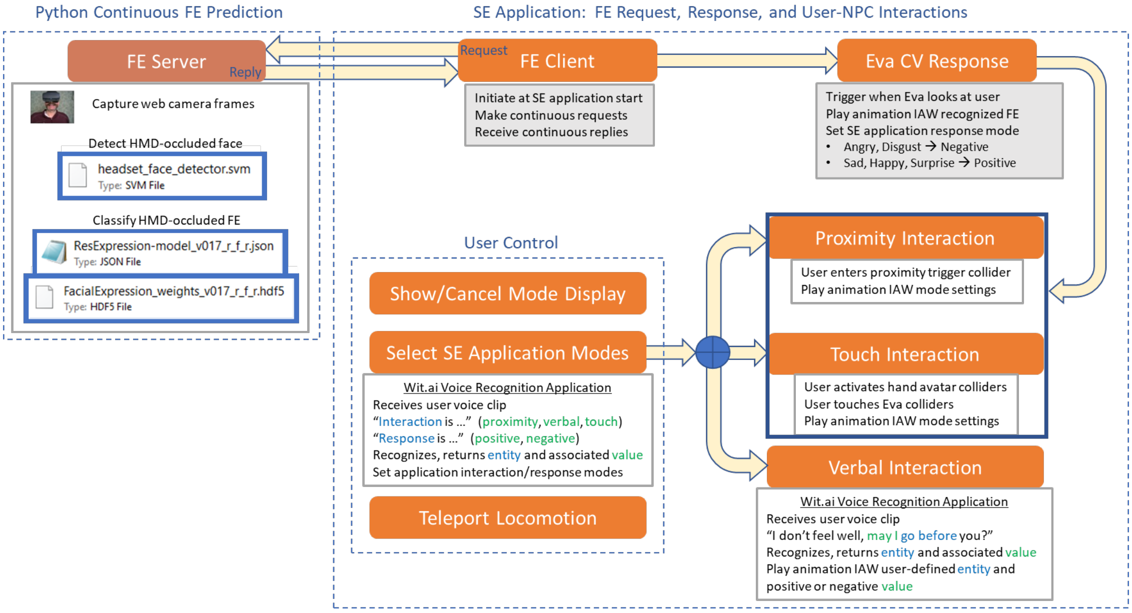

Python Continuous FE Prediction
SE Application: FE Request, Response, and User-NPC Interactions
FE Server
Reply
Request
FE Client
Eva CV Response
Capture web camera frames
Detect HMD-occluded face
headset_face_detector.svm
Type: SVM File
Classify HMD-occluded FE
ResExpression-model_v017_r_f_r.json
Type: JSON File
FacialExpression_weights_v017_r_f_r.hdf5
Type: HDF5 File
Initiate at SE application start
Make continuous requests
Receive continuous replies
Trigger when Eva looks at user
Play animation IAW recognized FE
Set SE application response mode
Angry, Disgust → Negative
Sad, Happy, Surprise → Positive
User Control
Show/Cancel Mode Display
Select SE Application Modes
Wit.ai Voice Recognition Application
Receives user voice clip
"Interaction is …" (proximity, verbal, touch)
"Response is …" (positive, negative)
Recognizes, returns entity and associated value
Set application interaction/response modes
Teleport Locomotion
Proximity Interaction
User enters proximity trigger collider
Play animation IAW mode settings
Touch Interaction
User activates hand avatar colliders
User touches Eva colliders
Play animation IAW mode settings
Verbal Interaction
Wit.ai Voice Recognition Application
Receives user voice clip
"I don't feel well, may I go before you?"
Recognizes, returns entity and associated value
Play animation IAW user-defined entity and
positive or negative value

Figure 2: Functional flow in Implemented Virtual Environment (VE)

flow GPU support. The primary software library versions are: OpenCV - 2.4.1, DLib - 19.21.1, Tensorflow - 3.4.1, and PyZMQ - 20.0.0 (which supports the ZMQ library version 4.3.3). Hardware environment is a HP Omen Obelisk 875-0084 desktop computer with the following specifications: Intel® Core™ i7-9700 processor, 16 GB SDRAM memory, and NVIDIA® GeForce® RTX 2070 SUPER™ graphics card with 8 GB dedicated memory. The web camera used for image frame capture to implement computer vision is a Logitech ® C270 HD operating at 30 frames per second (FPS), with a frame width and height of 640 and 480 pixels, respectively.

The occluded facial detector developed in this project use Histogram of Oriented Gradients (HOG) feature descriptors and a support vector machine (SVM) model to perform detection classification. Both OpenCV and DLib provide SVM techniques for performing detection and classification using HOG and Haar Cascade feature descriptors. All feature descriptors seek to represent images, or grids of image patches, using a concise set of information that describes and distinguishes the content of the image. The resulting feature vector for the entire image will be much smaller in size than the total number of pixels and color channels of that image. A HOG descriptor is the distribution of two dimensional direction gradients of the image pixel intensities. The gradients are calculated for a dense, and overlapping, grid of pixel patches across the width and height of the image. Gradients values are larger where there are abrupt changes in pixel values, so these features are useful for detecting distinct parts of images such as corners and edges [Jackson2022]. The SVM [Mallick2018, Jackson2022] first attempts to transform the dataset under consideration so the classes within it can be separated linearly. Then, it solves a constrained optimization problem to find the best possible separating line distinguishing the classes. The best possible line is one that provides the largest margin between the line and the closest examples in either class, while minimizing the decision errors that are made [Jackson2022]. A data point that lies exactly on the margin boundary is a support vector [Mallick2018]. The description above

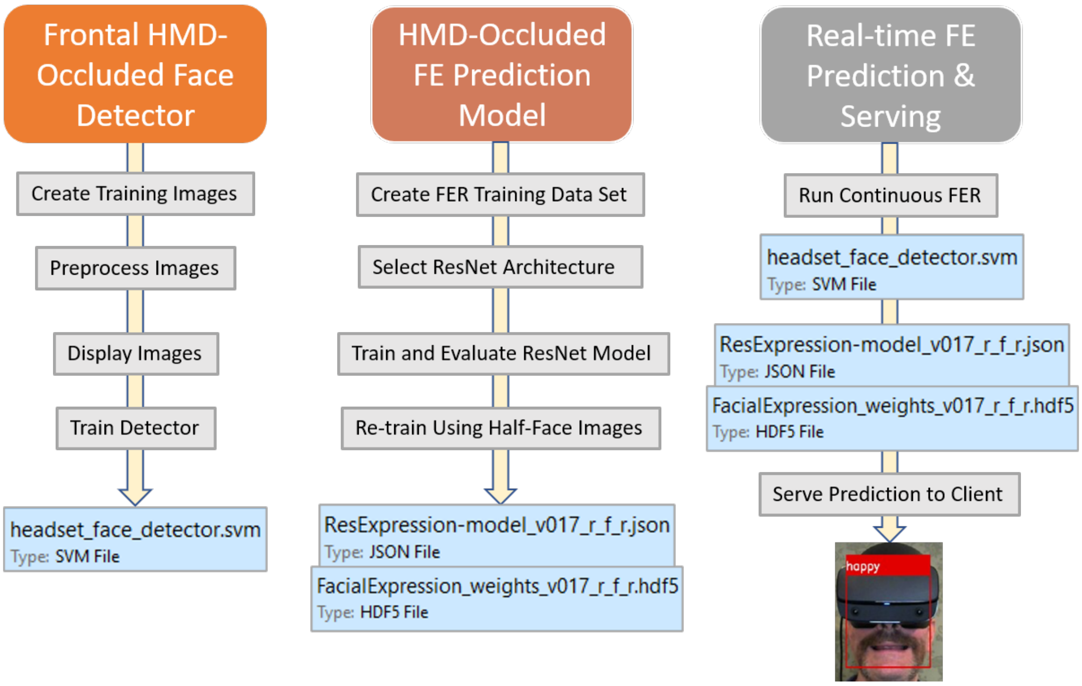

Frontal HMD-Occluded Face Detector
Create Training Images
Preprocess Images
Display Images
Train Detector
headset_face_detector.svm
Type: SVM File
HMD-Occluded FE Prediction Model
Create FER Training Data Set
Select ResNet Architecture
Train and Evaluate ResNet Model
Re-train Using Half-Face Images
ResExpression-model_v017_r_f_r.json
Type: JSON File
FacialExpression_weights_v017_r_f_r.hdf5
Type: HDF5 File
Real-time FE Prediction & Serving
Run Continuous FER
headset_face_detector.svm
Type: SVM File
ResExpression-model_v017_r_f_r.json
Type: JSON File
FacialExpression_weights_v017_r_f_r.hdf5
Type: HDF5 File
Serve Prediction to Client
happy

Figure 3: CV and FER Task Pipelines

is a 2D example. For higher dimensions, the SVM finds the best possible separating plane, or hyper-plane.

One of the challenges we faced was that images of the user's face occluded by the HMD were different that those not occluded in the data which we were using. So we needed to add faced with HMD to the data. These images were created using a web camera (Figure 4). A sliding image collection window captures user facial images at different locations across the web camera field of view. The window is outlined by a box when viewed in the web cam display. The user keeps their HMD-occluded face centered in the window as it moves. This is important because the coordinates defining the extent of the box sides will be used to specify the object to be detected during training of the HOG SVM. Figure 4 shows an example of training images displayed. Finally, the detection model is trained. The images are divided into 80/20% split training and test sets. The dlib.train_simple_object_detector ( ) function trains and evaluates the model performance.

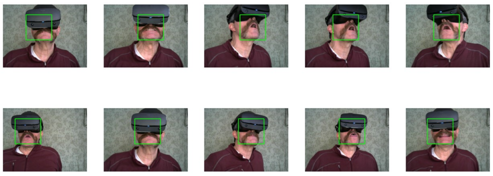


Figure 4: Training image samples displayed for verification

4.2 HMD-occluded FE Prediction Dataset

We decided at the outset of the project to implement a reduced set of facial expressions to demonstrate CV interaction in our implementation. Many of the data sets available for facial expression recognition have at least seven expression categories (for example, anger, disgust, sad, fear, happy, surprise, and neutral). Selecting only data corresponding to a subset of the full expression classes improves the overall recognition accuracy by eliminating a class that is hard to distinguish, for example, images labelled fear. Although not a neutral emotion, fear is often confused with several other expression classes in this implementation, and we felt that players (Nursing students) will welcome our story lines with excitement and curiosity and not with fear, as those were similar to regular training which nursing students undergo in their curriculum. Eliminating the neutral class also made sense because our story line handled both positive (smile) and negative (rude tone and statements) interactions as we did not plan any interaction for a neutral expression. As a result, we decided upon a facial expression set consisting of five – anger, disgust, sad, happy, and surprise – classification of player's lower half facial image. The remainder of this section explains the workflow tasks as shown in the middle column of Figure 3.

For the FER training, we chose existing data sets containing diverse images (many different subjects under varying conditions) as the core of the training set for this project. After researching the existing data sets, the RAF and the FER2013 data sets were selected. The RAF data set consists of color and monochrome images at a 100×100 pixel resolution. The FER2013 data set has monochrome images at a 48×48 pixel resolution. Some of the facial images have other objects (hands, sunglasses, hats, watermarks, for example) that partially occlude facial features. Some images have varying aspect and rotation angles. We noticed that the existing data set composition across the five expression classes is unbalanced and is a common occurrence in these types of data sets. Finally, the FER2013 48×48 image pixel resolution is too small for the multiple ResNet convolution stages. These challenges are addressed by performing in-place data augmentation and class weighting during training of the CNN FER model. Both techniques help address the class imbalance inherent in the data sets. The combined data set uses RAF images from all five classes, with FER2013 images only from selected classes (disgust and surprise). This approach also helped to increase examples in the underrepresented classes, as that was our concern. Finally, the FER2013 image sizes were expanded to 100×100 pixels using interpolation and the number of channels was expanded to three for implementation.

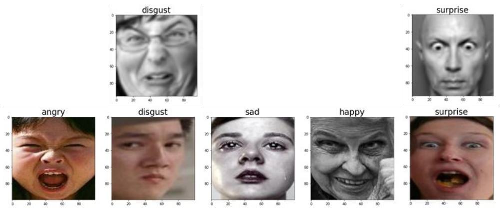


Figure 5: Example RAF (bottom) and FER2013 (top) images [Carrier2013]

4.3 Occluded Facial Expression (FE) Prediction Model using ResNet Architecture

We used Keras with a Tensorflow backend to develop the ResNet CNN model. The code for the CNN models is implemented, trained and evaluated on Google Colab using available GPUs. The number of training epochs is explored experimentally (ranging from 30 to 120). GPUs were necessary to ensure reasonable model training times. After training and evaluating different model/training hyper-parameter configurations on Google Colab to identify the best performer, the selected model and weight files were moved to the Spyder development environment on the desktop computer for use in the real-time FE prediction server.

As explained in [Jackson2022], ResNet architecture provides an initial convolution layer (7×7, and 2×2 strides), up to three residual stage layers, average pooling and flatten layers, and a 5 node dense output layer. The convolution depth dimensions for the three ResNet stages are (64, 64, 256), (128, 128, 512) and (256, 256, 1024), respectively. Each residual stage contains three bottleneck convolution blocks where the first block uses a projection shortcut to match input and output dimensions. The remaining blocks use identity shortcuts. The total depth of this model (counting the initial 7×7 convolution and the final dense layer) is 29 (ResNet-29) if all three residual stages are used, and 20 (ResNet-20) if only the first two residual stages are used.

4.3.1 Training and Evaluating ResNet Model as facial expression classifier

The best performing model was designated ResExpression-model_v014_r_f.json (which also has an associated model weights file) with an average prediction test accuracy of 88%. Experimentation showed that training this model for more than 90 epochs did not significantly improve validation accuracy, but showed signs indicating the onset of over fitting (training loss continues to decrease, but validation loss stays the same, or increases).

Expre- -ssion	Prec- -ision	Recall	F1	Sup- -port.
angry	0.70	0.78	0.74	131
disgust	0.65	0.78	0.71	221
sad	0.76	0.86	0.81	383
happy	0.96	0.89	0.92	925
surprise	0.97	0.92	0.94	798
Accuracy			0.88	2450
Macro Avg.	0.81	0.84	0.82	2450
Weighted Avg.	0.89	0.88	0.88	2450

Table 1: Full face model classification report

4.3.2 Retrain Selected Model Using Half Face Images

The final step to develop the HMD-occluded FE prediction model is adaptation of the full face model to a half face expression prediction model. This is done by retraining the full face model on the lower half of the images in the Reduced RAF+FER2013 data set. All images in the Reduced RAF+FER2013 were processed to remove the top 50 rows of pixels and then the images were resized using interpolation back to 100×100 pixels. The model was retrained using the same architecture, characteristics and training hyper-parameters as described in the previous section. This allowed us to train the model using the examples we collected from those wearing the HMD. The resulting model provides 80% prediction accuracy on the half-face test images, losing only 8% performance relative to the full face model. However, during experimentation using continuous image capture and FE prediction, we felt this model did not provide stable FE prediction over the amount of time necessary to provide a consistent input to the our empathy application module, and will need to be improved in future. The implemented solution to increase Facial Epression (FE) prediction stability was to introduce a relatively small (989 images, < 6% of the dataset total) set of user face supplemental images into the Reduced RAF+FER2013 dataset. User facial images presenting all five expressions were captured and classified into the five different expression classes. An example of the user face supplemental images is shown in Figure 6.

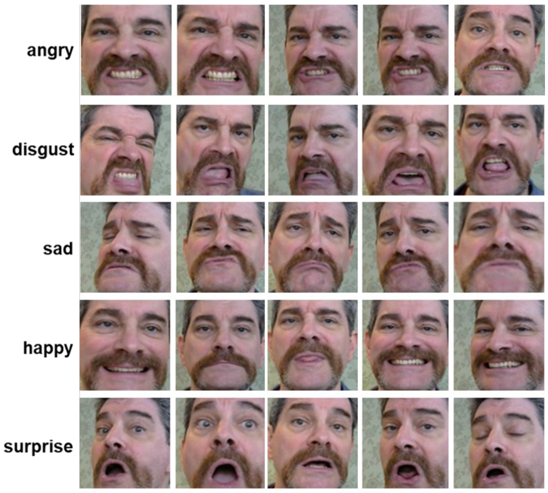


Figure 6: User face supplemental image examples

The FE prediction model was retrained on the half-face version of this new data set (Reduced RAF+FER2013+RKJ, using the second author's initials to represent the supplemental images). A sample of the half-face images from this dataset is shown in Figure 7. The resulting model provides 81% prediction accuracy. This is only a 1% increase in prediction accuracy compared to the data set without the supplemental images, however, experimental observation indicates that the stability and consistency of FE prediction is significantly improved in real-time continuous image capture with the web camera. This half-face prediction model is designated ResExpressionModel_v017_r_f_r.json (and also has an associated model weights file, FacialExpression_weights_v017_r_f_r.hdf5). This is the model used to move forward in the project and implement the real time FE prediction server. Table 2 shows the overall classification report for half face images.

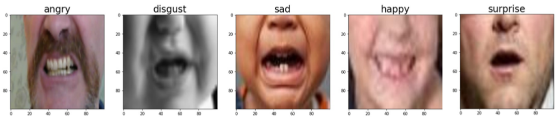


Figure 7: Reduced RAF+FER2013+RKJ half-face images

4.4 Real Time FE Prediction and Serving

Implementation of the Computer Vision based interaction requires the continuous capture of facial images while the user is wearing a VR-headset or HMD. The web camera is positioned directly in front of the user and at the height of the upper chest, or lower neck, as shown in Figure 1. The camera positioning ensures a full and clear view of the lower half face and provides the best performance as determined by experimentation.

Expre- -ssion	Prec- -ision	Recall Recall	F1 F1	Sup. Sup.
angry	0.56	0.72	0.63	157
disgust	0.56	0.68	0.61	271
sad	0.66	0.68	0.67	368
happy	0.90	0.89	0.89	945
surprise	0.93	0.82	0.88	857
Accuracy			0.81	2598
Macro Avg.	0.72	0.76	0.74	2598
Weighted Avg.	0.82	0.81	0.80	2598

Table 2: Half face model classification report

4.4.1 *Run Continuous FER*

Initialization tasks are first accomplished to prepare for continuous frame capture and FE processing. These tasks include: using OpenCV to assign the web camera as a video frame capture source and create a window for display of the frames, loading the DLib headset_face_detector.svm model, loading the Keras ResExpression-model_v017_r_f_r.json model and FacialExpression_weights_v017_r_f_r.hdf5 weights files, and setting up the ZeroMQ message socket/address to bind (connect) as the reply server. After performing the above initialization, the FER server enters into a continuous loop which can be terminated by the user through a keyboard command. Each pass through the loop: (a) Captures a web camera frame, (b) Detects HMD occluded faces using headset_face_detectoršvm, (c) Extracts the portion of the frame corresponding to the HMD occluded face, (d) Creates a lower half image of the detected HMD-occluded face, (e) Resizes the image to 100×100 pixels, (f) Normalizes the image pixel values between 0 and 1, (g) Reshapes the image as a tensor, (h) Sends the image to ResExpression-model_v017_r_f_r.json for FE prediction.

The results of the prediction are displayed as an annotated box around the user's HMD occluded face in the display window upon completion of each pass through the loop. See Figure 8 for an annotated example of FE as surprise.

4.4.2 *Serve Face Expression (FE) Prediction to the Client*

FE prediction serving also occurs within the continuous loop described above. During each pass through the loop, the following actions are performed: (a) The reply server socket tries to check for the receipt of a request from the Unity request client, (b) If there is no request, an exception is thrown which is handled by printing a message to console, (c) Execution flow is returned to the outer continuous FE prediction loop. (d) Else there is a request, so print the request to console and do the next four steps, (e) Append the most recent FE prediction to a current expressions list of maximum length of 10, based on the user selected value, (f) Determine the most common prediction in this list, (g) Reply to the Unity request client with the most common prediction, (h) Print the current expression list and reply to console, (i)Execution flow is returned to the outer continuous FE prediction loop.

Experimental observation of the time required to complete a request-reply cycle shows that the majority of these transactions occur within 0.20 - 0.25 seconds. Most of this time is used by the FE detection and prediction workflow. The additional time added by the message interface for requests and replies is within the range of 0.001 - 0.01 seconds.

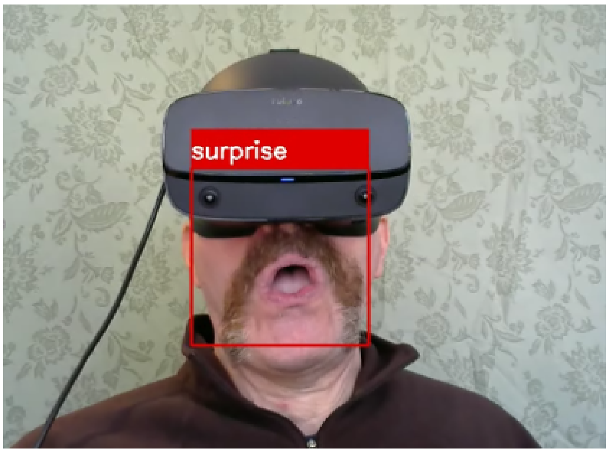


Figure 8: HMD occluded FE prediction display window

4.5 SE Application FE Request Client

The SE application also required modification of the primary NPC controller, the addition of a threaded request client class, and new visual interaction animations for the primary NPC, Eva. The primary NPC controller is modified to implement both the continuous FE requests and to provide the animation triggers upon receipt of the FE predictions. The controller instantiates and activates the requester class and provides animation triggers upon receipt of FE predictions. The NPC controller initiates the visual interaction animation when the user is within range (a specified distance and angle) that also corresponds to the conditions necessary to trigger Unity Inverse Kinematics animation that causes Eva to “look” at the user. Based on the expression message received from the FE prediction server, one of five Eva animation responses are triggered. Additionally, the application response mode is set to positive if the the received expression is sad, happy, or surprised. The response mode is set to negative if the received expression is angry or disgust.

4.5.1 Facial Animation and Audio Clips used for NPC-Eva

We obtained body pose animation clips from Mixamo.com, and used these as NPC body animation building blocks for Eva. However, these clips do not provide facial animation. We added these expressions by first composing individual verbal responses for Eva to deliver when observing each user facial expression. The length of each spoken response was designed to match the duration of the corresponding body pose animation sequence. Then, facial expression animations were added to each body pose animation using the blend shapes available in the Eva Skinned Mesh Renderer. The Eva model provides many facial blend shapes that can be used to create any type of facial expression. Multiple facial blend shape characteristics were selected to create the desired effect. Key frames were created in the Unity Mecanim Curves display to position each desired facial expression at the correct time during animation. The key frames were connected using curve interpolation available in the Mecanim Curves tool. During development of a facial animation, the sequence can be run on the Eva model in Unity editor mode to monitor the effect of each change. Finally, the verbal response for each visual interaction was recorded as audio clip. The audio clip is played automatically upon start of each animation sequence by embedding an event at time = 0 of the animation sequence.

4.5.2 Lip Sync Animation for NPC-Eva

A final animation was incorporated which uses Eva’s Viseme blend shapes to synchronize her mouth and lip movements to the playback of each recorded verbal response. Visemes are the patterns of lip and mouth movements that correspond to the basic phoneme sound utterances during speech. Oculus provides a Lip Sync plug in to Unity that maps a model’s available visemes to a standard set of visemes determined by Oculus. Oculus also provides a utility that computes and stores the visemes for each audio clip to make the playback during animation more responsive. The Lip Sync animations are triggered by a method incorporated into the primary NPC controller.

5 RESULTS AND INTERACTIONS

Primary interactive NPC, Eva, provides many different reactions to the user based upon a variety of simulated senses that trigger interactions. Additionally, Eva can also provide both initial first impression reactions of the user approaching the coffee kiosk, and additional reactions as the user attempts to break in line.

Eva's initial reaction is based upon her recognition of the user's facial expression in one of five classes: anger, disgust, sad, happy, or surprise. This visual interaction mode is activated whenever the user approaches Eva within a distance and angular range specified in the SE application. As the user continues to move forward and breaks line in front of Eva, a final reaction by Eva is presented based on the current setting of the application's interaction mode and Eva's perception of the user's mood based on her recognition of the user's facial expression. There is at least one animated response for each combination of three interaction modes, such as proximity, spoken, and touch and Eva's two facial expression recognition based response modes (positive and negative) as explained earlier [Jackson2022, Deegan2022].

The current SE application version provides the user a full set of control mechanisms. The user controls the modified application interaction mode setting with voice commands. The voice commands and the user's spoken interaction utterances are enabled via buttons on the Touch Controllers. Once selected, the touch interaction mode can be employed when the user depresses either of the Touch Controller palm triggers and "touches" Eva to push her away with the user hand avatar. The player can also display the application mode setting, and then clear this display, with Touch Controller buttons. The user can also moves through the VE with a teleportation mode of locomotion. Teleportation is accomplished with the thumb joystick on the right Touch Controller. Rotation of the user's facing direction is accomplished with the thumb joystick on the left Touch Controller.

6 SCENARIO, STORY LINE, AND EMPATHY INTERACTIONS

We implemented the following scenario and story line: (a) A room setting pops up. (b) When the player happen to reach to a fountain where the person starts to experience very disturbing voice like "Go away from here, you are not supposed to come here" etc. and he will try to navigate back to map and this time he meets with a dog on his away. (c) Dog interaction is provided as the user experiences general disturbing voice/noise like "Go out of my way", "you are not meant to be here." These voices are from [Deegan2022]. (d) Player tries to scavenger hunt to a coffee kiosk and meets EVA. Eva would offer the player her place in line, and provide empathetic interactions, or not, based on player's facial expressions. (e) The player buys a coffee and communicates with coffee seller, (f) Finally, from coffee kiosk he tries to head towards clinic.

6.1 Voice based Empathy Interactions

The modified application uses Wit.ai cloud-based voice recognition to enable voice commands and interaction with Eva. The voice recognition is based on entities and values. Move aside, go away, get lost, and get out will generate a negative response from Eva when the application is in the spoken mode. The voice recognition application is trained to recognize various sentences or phrases that contain the above mentioned entities and values. Below are only a sample of examples of sentence and phrase variations that may be recognized are provided below: (a) Get out of my way, I ... (want to, need to) ... go first. (b) "May I go first, please?", (c) (I'm tired, I don't feel well) "... please allow me to go first." These verbal statement are understood by NPC-Eva and generate negative and positive empathetic response from NPC-Eva.

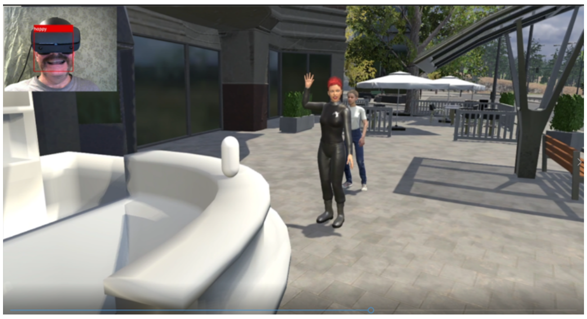


Figure 9: Eva response based on recognition of user's facial expression

7 CONCLUSION

Our implementation demonstrates an efficient messaging interface between the Python FE prediction server and the Unity C#.NET client. The ZeroMQ library provides a lightweight approach for creating a communication interface that was easier to implement. The C# port of ZeroMQ, is easy to implement in a separate thread so continuous client requests can be made without interrupting Unity graphic rendering tasks.

In summary, we implemented our primary goal of creating a 3D Unity based Virtual Environment (VE) so that a player can experience the hallucinatory sounds which person with schizophrenia constantly hears. Six new NPC body, facial expression, and voice animations were integrated into the primary NPC, Eva, to support the new visual interaction mode. We showed that NPCs, such as Eva, can provide provide empathetic verbal interactions based on positive and negative interactions by the player wearing a VR headset.

An IRB study has been undertaken by College of Nursing Faculty at our university over several semesters, and early results are promising, and bode well for our future efforts.

8 FUTURE WORK

In future, we imagine extending the virtual environment to an Augmented Reality (AR) space enabling a multi-person immersion experience with integrated interactive presentations; potentially using actors to add depth to the simulation by playing the roles of concerned family and friends of the person with schizophrenia. In addition, we would like to include neutral face recognition by NPC to provide better transition, feedback, and experiences to the player by NPC. In addition, eyes provide expressive emotive features in human-faces. As see through VR-headset become more available, we anticipate future incarnations of our work to incorporate see through headset facial images for training and better accuracy.

9 ACKNOWLEDGMENTS

We are thankful for College of Nursing Faculty Drs. Deborah Pina-Thomas and Lynn Philips for their input, undertaking the University IRB case study, and providing a simulated recording [Deegan2022] of what a person with Schizophrenia hears.

10 REFERENCES

[Barbot2020] James C. Barbot, Baptiste; Kaufman. 2020. What makes immersive virtual reality the ultimate empathy machine? Discerning the underlying mechanisms of change. Computers in Human Behavior 111 (2020), 106431. https://doi.org/10.1016/j.chb.2020.106431

[Carrier2013] Pierre-Luc Carrier and Aaron Courville. 2013. Challenges in Representation Learning: Facial Expression Recognition Challenge. https://www.kaggle.com/c/challenges- in-representation- learning- facial- expression-recognition- challenge/data

[Cortes1995] C. Cortes and V. Vapnik. 1995. Support vector networks. Machine Learning 20 (1995), 273-297.

[Dalal2005] N. Dalal and B. Triggs. 2005. Histograms of oriented gradients for human detection. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 05),Vol.1.886-893

[McAllister2020] Jodi; McAllister Margaret; Lazenby Mark Dean, Sue; Halpern. 2020. Nursing education, virtual reality and empathy? Nursing Open 7, 6 (2020), 2056-2059.

[Deegan2022] Patricia Deegan, Hearing voices that are distressing: A training simulation experience. National Empowerment Center, NARPA.org, Accessed 2022.

[Claudia2020] Claudia; Riva Giuseppe; Villani Daniela Di Natale, Anna Flavia; Repetto. 2020. Immersive virtual reality in K-12 and higher education: A 10-year systematic review of empirical research. British Journal of Educational Technology 51, 6 (2020), 2006-2033.

[Ward2016] Julia Ward. 2016. The empathy enigma: Does it still exist? Comparison of empathy using students and standardized actors. Nurse Educator 41, 3 (2016), 134-138.

[Ward2012] Julia Ward, Julianne Cody, Mary Schaal, and Mohammadreza Hojat. 2012. The Empathy Enigma: An Empirical Study of Decline in Empathy Among Undergraduate Nursing Students. Journal of Professional Nursing 28, 1 (2012), 34-40.

[Ward2009] Julia Ward, Mary Schaal, Jacqueline Sullivan, Mary E. Bowen, James B. Erdmann, and Mohammadreza Hojat. 2009. Reliability and Validity of the Jefferson Scale of Empathy in Undergraduate Nursing Students. Journal of Nursing Measurement 17, 1 (2009), 73-88.

[Johnson2014] Keith Johnson and Sudhanshu K Semwal. 2014. Shapes: A Multi-Sensory Environment for the B/VI and hearing-impaired community. In 2nd International Workshop on Virtual and Augmented Assistive Technology (VAAT) at IEEE Virtual Reality 2014. 1-6.

[Mallick2018] Satya Mallick. 2018. Support Vector Machines (SVM). https://learnopencv.com/support-vector-machines-svm/

[Jackson2022] Ron Jackson, Using the Unity Game Development platform to build Virtual Reality Schizophrenia Empathy training applications, pp1-107, Advisor: SK Semwal, MS Thesis, University of Colorado Colorado Springs, 2022.

[ZeroMQ2021] Zeromq.org. 2021. ZeroMQ: An open-source universal messaging library. April 27, 2021 from https://zeromq.org/

[Tolman1948] Edward Tolman, In Pschological Review, Cognitive Maps in Ants and Man, 1948, vol 55, pages 189-208.

[Dlib2021] Dlib.net. 2021. DLib C++ Library. Retrieved April 10, 2021 from http://dlib.net/

Texture-based Global Illumination for Physics-Based Light Propagation in Interactive Web Applications

Adrian Roth
University of Applied Sciences and Arts
Northwestern Switzerland
Hochschule für Technik
Bahnhofstrasse 6
5210 Windisch
adrian.roth.ar@proton.me

Hilko Cords
University of Applied Sciences and Arts
Northwestern Switzerland
Hochschule für Technik
Bahnhofstrasse 6
5210 Windisch
hilko.cords@fhnw.ch

ABSTRACT

Lamps are difficult to market via websites. Renderings and photographs of showrooms rarely fit consumer preferences and demands and the technical specifications for lighting are unintuitive and difficult to understand for non-experts. This poses a challenge for lighting system manufacturers and suppliers, as customers increasingly make purchases online. Giving customers the option to see how a lamp affects the environment would provide them with an intuitive way to find a suitable product. Unfortunately, expensive global illumination rendering methods are required to accurately simulate the effect of a specific lamp on the environment. However, contemporary methods are not suited for low-end consumer hardware or interactive web-applications. Therefore, we present a new texture-based global illumination method that simulates the effect of specific lamps on complex, polygonal 3D geometries on the GPU. Our method employs iterative light propagation to pre-compute the illumination in texture space, leveraging GPU-based bounding volume hierarchies. Our method can provide pre-calculated physics-based lighting on demand in under a second for interactive scenarios in browsers using WebGl - outperforming state-of-the-art offline renderers significantly.

Keywords

physics-based simulation, real-world lights, pre-calculated global illumination, light propagation, texture-based, real-time web-applications, polygonal objects

1 INTRODUCTION

The characteristics of lights are abstract and difficult for non-experts to grasp intuitively. Thus, manufacturers and retailers of lamps and lighting systems find it challenging to market their products through a website. Images of showrooms, such as seen in Figure 1, are commonly used to showcase lighting systems in the real-world. However, this may not meet consumer preferences and needs and cannot demonstrate the lighting system in the user's actual environment. Furthermore, the visual perception of lighting systems is heavily determined by the concrete room and furniture in which they are installed. Especially if there is a large amount of indirect lighting.

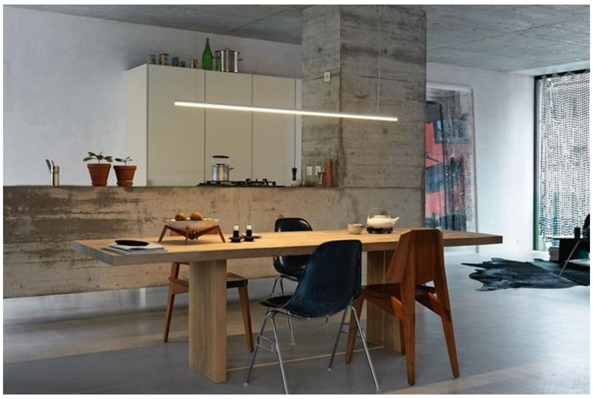


Figure 1: *Example of a lamp showcase from the company Ribag AG [Rib24].*

Previously, we developed a web configuration tool that determines physical light distribution properties based on physical lamp data (IES-data [Ill91]) using global illumination (GI) [Rie+23]. By applying an analytical solution for concave, rectangular, cuboidal rooms to solve the rendering equation, we achieved high performance on GPUs and mobile devices and determined accurate physical light distributions within milliseconds (Figure 2). This lets us provide pre-calculated GI on

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.

demand in under a second, allowing users to re-render scenes for a quick and iterative design process in a web browser.

However, without furniture, the rooms are lifeless and cannot accurately represent a customer's unique environment. Furthermore, the empty rooms struggle to convey their size or scale, making it difficult to understand the effect of lighting systems in real-world settings.

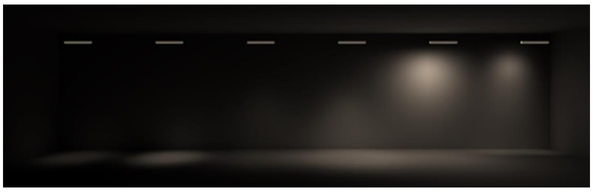


Figure 2: *An empty room with several lamps rendered physically accurate with analytical GI [Rie+23].*

To address this issue, we propose a GI approach for creating physically accurate illumination of IES-Data in furnitured scenes. As a result, we suggest implementing spatial acceleration methods on the GPU while facilitating only the most basic OpenGL features in order to significantly increase light calculation and path tracing performance and allow for an implementation on the web - with WebGL. Thereby, all static lighting information is pre-calculated and stored in textures, enabling interactive rendering performance during exploration. Our method is suitable for determining accurate illumination and interactive exploration on the Web using WebGL (Figure 3). Our technique accurately replicates shadows, indirect lighting, and diffuse surface reflections while executing extensive ray casting operations across the entire geometry for the path tracing algorithm.

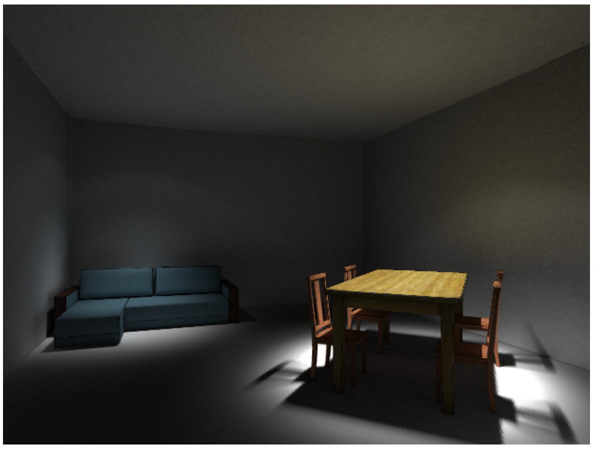


Figure 3: *A furnished scene of a polygonal living room rendered physically accurate on the GPU using our proposed rendering method.*

2 RELATED WORK

Real time global illumination is an active topic of research. Recent advances in hardware [Bur20] and rendering techniques [WP22; Lin+22] enable real time path tracing in complex scenarios. While these techniques still rely on sophisticated hardware, they reflect a broader shift in consumer expectations toward realistic rendering.

The number of ways to achieve global illumination is considerable. [DBB18] provides a comprehensive overview of global illumination algorithms. Of these, path tracing [Kaj86; LW93] and radiosity [Gor+84] are the most relevant to our work. Path tracing determines global lighting using stochastic ray casting, whereas radiosity uses an iterative radiance transfer technique to exchange light between a scene's surfaces.

The development of real time global illumination for web applications has additional hurdles as it must cater to a wide range of platforms. E.g., WebGL enables web applications to access graphics hardware from a browser while being platform-independent. However, this platform-agnostic design comes at the cost of lower performance and a limited range of features compared to desktop solutions like OpenGL, Vulkan or DirectX.

Early attempts to perform global illumination with WebGL 1.0 were highly effective, but limited in scale and ability [Con+11; Hac15]. Recent approaches using the upgraded framework, WebGL 2.0, can display complicated scenes with excellent precision and detail. Lesar et al. produced real-time representations of volumetric medical data utilizing path tracing in web environments [LBM18]. They achieve accurate renderings by iteratively and progressively using Monte Carlo ray tracing. However, their approach is limited to volumetric data. In contrast, Nilsson et al. obtained relevant results for mesh-based scenes [NO18]. Using path tracing, they were able to render scenes with thousands of triangles at about 10Hz. By accumulating samples they are able to render high quality images of simple scenes. Most recently Vitsas et al. [Vit+21] have developed a general purpose ray tracing framework on top of WebGL able to produce photorealistic 3D graphics on the web.

Light maps, on the other hand, have long been the norm for global illumination effects in real-time applications [ÖA17; Ras+10]. Because of the static nature of light maps, recent advances in real-time global illumination make them increasingly insignificant. However, so far, they still remain the standard for games, web, and mobile applications due to their high visual quality for static scenes and their low impact on runtime performance. Nevertheless, the above-mentioned, recent advances in real-time global illumination have also benefited the fast construction of light maps [Luk+13; CL21].

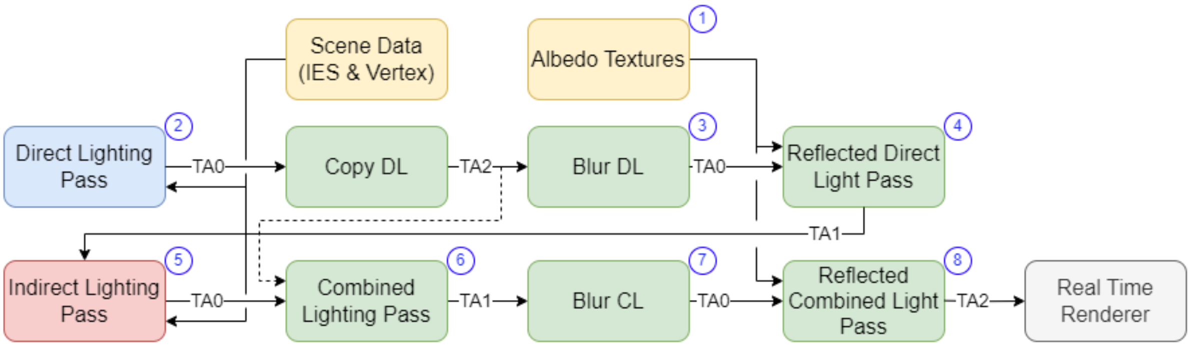


Figure 4: *Our global illumination render pipeline: Direct light shader executions (blue), indirect light shader executions (red) and effects shader executions (green). Yellow indicates static resources. Each step's output indicates the texture array (TAs) that are rendered into. The numbers at the top right of each step correspond to the numbered textures in Figure 5.*

Typically, efficient path-tracing and hence, efficient light map creation, necessitate specific data structures for accelerating ray casting. Bounding volume hierarchies (BVHs) are well established methods that considerably reduce the number of ray-triangle intersection tests [LBM18; STØ05]. However, implementing BVHs on the GPU using WebGL remains challenging. WebGL does not support stacks, hence stack-less algorithms are required. Threaded bounding volume hierarchy (TBVH) uses a fixed traversal order with pre-computed hit and miss pointers [STØ05]. Hachisuka proposed a variation to TBVHs as Multiple-threaded bounding volume Hierarchy (MTBVH), to lessen the negative performance impact from the fixed traversal [Hac15]. Thereby, MTBVH pre-computes several versions of the hit and miss pointers that are optimised based on the general direction of a ray.

3 OUR APPROACH

Our rendering method is related to path tracing and radiosity, and it operates with similar concepts. For each object we generate a series of light maps in a preprocess we call the global illumination pipeline. To enable interactive framerates, the results are given to a real-time rasterisation renderer.

As is often the case, our light maps are generated using an iterative Monte Carlo ray-tracing algorithm. In the first iteration, the direct light from each light source is determined and stored in a light map for each object. In a second iteration, the indirect light is calculated for each surface using a reverse path tracing algorithm. To determine possible sources of reflected light, we cast a number of rays in random directions for each pixel in the light map. Therefore, we transform the uv-coordinate into the world space. The amount of incoming light is then determined using the direct light maps.

Given enough rays, we can expect the results to converge to a reasonable estimation of the reflected light based on the idea of Monte Carlo global illumination by Lafortune [Laf96]. The indirect lighting pass can be repeated for as many iterations as desired, but in practice, the cost of a second reflection outweighs it's contribution to the overall illumination effect. Finally, the computed direct and indirect light maps are accumulated and fed into the real-time renderer.

In the following, we will elaborate on the details of the global illumination pipeline, specifically in terms of the improvements we implemented to render polygonal models and apply the BVH to accelerate the ray casting procedure. We will also briefly present our solution to circumvent the texture limitation of WebGL 2.0.

3.1 Global Illumination Pipeline

The global illumination pipeline assumes a static scene. There are no moving objects or lights that could cause the lighting to change. This fits in very well with our use case in the configuration of professional lighting systems.

Our global illumination pipeline consists of two basic steps: direct and indirect lighting passes. Figure 4 depicts the complete process, which we shall detail below. In Figure 5, the resulting textures of the model of a chair for each render call in the process are shown. Whereas Figure 6 shows the final result.

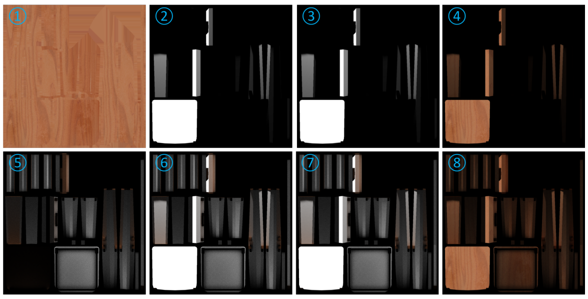


Figure 5: *Evolution of the texture of a chair: (1) Albedo Texture, (2) Direct Light (3) Blurred Direct Light, (4) Reflected Direct Light, (5) Indirect Light, (6) Combined Light, (7) Blurred Combined Light, (8) Reflected Combined Light.*

To compute the direct light (2), we walk over each point on each surface and determine how much light we receive from each light source. This requires both the scene geometry to check for obstructions and the determination of the incoming light intensities in the IES data format. The results are stored in textures for each object. For our purposes, the texture mapping for the polygon models is expected to be unique, ensuring that no two surfaces share pixels. If they did, the light data from one surface would override another's.

Afterwards, the results of direct light texture are copied for later use. To reduce artefacts from sampling the IES file, we apply a Gaussian blur to the direct light texture (3). The direct light is then multiplied by the object's albedo texture to get the reflected direct light (4). The Gaussian blur must be applied prior to multiplying with the albedo texture. If we would apply the blur after multiplying the albedo texture and direct light, we would lose the albedo texture's details.

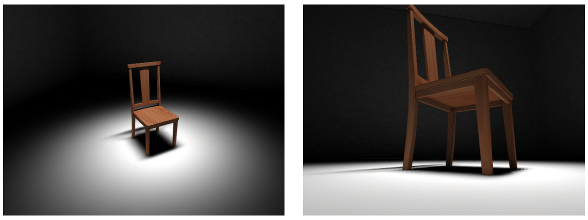


Figure 6: *A chair, rendered with our render method.*

To determine the indirect light (5), we step through each point on each surface and cast numerous rays in random directions. Wherever the rays hit a surface, we extract the associated light intensity from the reflected direct light texture. We then average the received light intensities from all rays to determine how much light is received at the sampling spot. Thus, we gather global light information from the scene onto the sampled point (see Figure 7).

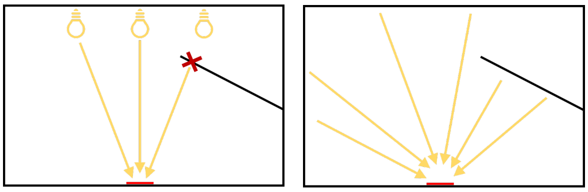


Figure 7: *Diagrams of the direct (left) and indirect (right) lighting passes. The sampled surface is in red. The cast rays are indicated in yellow and point in the direction the light travels, and therefore against the direction the ray is cast.*

Afterwards, the results of the indirect lighting pass and the original texture of the direct light are summed up (6). The end result is the combined light intensity that reaches each surface of the object. We blur it again to remove noise from the direct and indirect passes (7), then multiply it by the albedo texture of the object (8). Finally, we are left with a texture that depicts the total amount of light reflected on the surface of the object.

3.2 BVH

During the direct lighting and indirect lighting passes, we perform millions of ray-triangle intersections. To accelerate this procedure, we employ BVHs. We create the BVH using a simple top-down algorithm and the cost function surface area heuristic (SAH) as described by Jefferey Goldsmith and John Salmon [GS87]. The traversal algorithm on the GPU is based on the MTBVH algorithm described by Hachisuka et al. [Hac15]. Wherein only the traversal order of the BVH is required, represented as hit and miss links for each node, and the axis-aligned bounding box (AABB). Leaf nodes also store a reference to the set of vertices they contain.

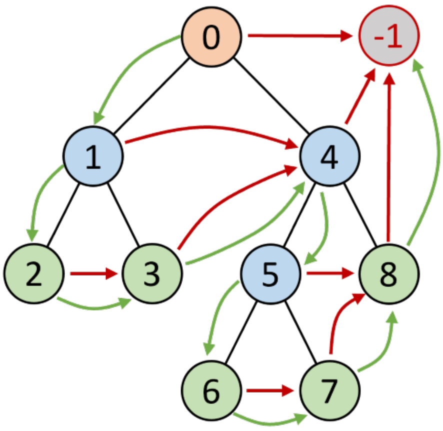


Figure 8: *Example of TBVH: the red node is the root node, the blue nodes are internal nodes and the green nodes are leaf nodes. The nodes are numbered in depth-first order. Green arrows show the hit links and red arrows show the miss links.*

The hit and miss links are shown in Figure 8. The green hit links follow the depth-first numbering of the nodes. The miss links always point to the next unvisited, unrejected node. For leaf nodes, they are always equal to the hit links. The miss link of internal left nodes (e.g. 1 and 5) points to their sibling node. Internal right nodes (e.g. 4) point to the closest unvisited right sibling of a preceding node. If there are no more viable nodes, the hit and miss links point to -1 as an indication that the traversal has ended.

With TBVH the traversal algorithm remains straightforward. It merely needs to track the current minimal distance when stepping through the BVH. At each level, we check whether the bounding volume is hit or missed, and then proceed to the node provided in the hit or miss link respectively. Extending the traversal to MTBVH merely requires more hit and miss link lists for the optimised traversal in each major direction.

Ordering the hit and miss links, AABBs, and vertex reference indices in the same order in respective lists eliminates the need to construct a data structure on the GPU. The node is represented by an index that reads relevant data from one of four lists (hit links, miss links, AABBs, and vertex indices).

To accommodate the transfer of the data from the CPU to the GPU, the traversal data of the BVH has to be encoded into a texture. The four aforementioned lists are therefore wrapped into a two dimensional array. A node is then no longer represented by an index but a 2D coordinate (see Figure 9). The hit and miss links can therefore also be stored as 2D coordinates pointing to the next node in the traversal order. The AABBs are represented through two three dimensional vectors for the lower and upper bound of the AABB. The vertex indices can remain as they are. This results in a total of 30 floating point values and two integer values per node.

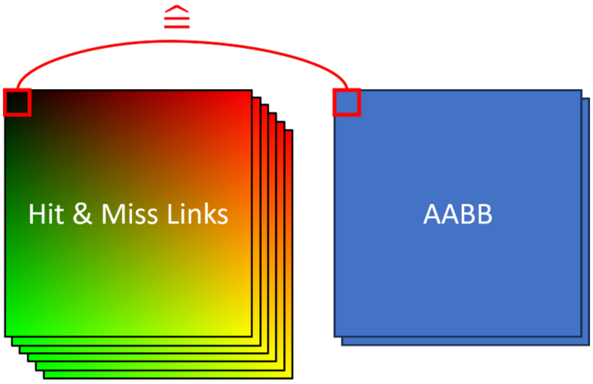


Figure 9: *Encoding of BVH traversal data in two data texture arrays. The data for a node in the BVH is dispersed across all textures but always at the same position in each texture. Each node can be represented through the 2D coordinate of that position.*

We can store all 32 values in 8 four channel floating point pixels. The hit and miss links per direction can be stored in one 4 channel pixel, each link has 2 components. The lower bound of the AABB and the starting index of the vertices can be stored as one pixel as well as the upper bound and the end index (Figure 10). To maintain the simplicity of representing a node as a 2D coordinate, each of these eight pixels are stored in separate textures at the appropriate position.

For our implementation we've opted to store the hit and miss links in a six layer texture array and the AABBs in a two layer texture array (Figure 9). We've separated the two out of convenience and clarity in the code base and for easier debugging. However, there is no technical reason to separate the two and a single texture array could hold all eight textures.

3.3 Texture Limitations

WebGL 2.0 imposes a hard limit for active textures during a single pass. The limit depends on the hardware and currently is usually either 16 or 32 textures. WebGL is designed for forward rendering applications

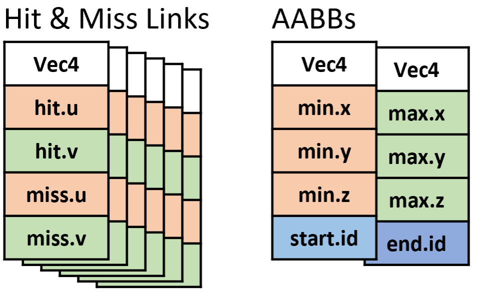


Figure 10: *Encoding of a single BVH node as eight 4 channel pixels (vec4). 6 pixels to hold the hit and miss links for each direction in the MTBVH and 2 pixels to define the AABB and the start and end indices of the associated vertices.*

which rarely exceed this limitation. However, our method needs the direct light maps of every model in the scene during indirect lighting passes. Additionally, the scene geometry and acceleration structures are also stored and passed to the GPU through data textures. As the number of models in a scene grows, it becomes increasingly difficult to accommodate all of its textures with the limit imposed by WebGL.

To circumvent the texture limit, we can use texture arrays. They count as one texture against the limit but are able to hold several textures in a three dimensional space. All textures in a texture array have the same resolution, which might pose problems when reading or writing to them on the GPU. There are two options for creating a texture array for textures with different resolutions. Either the textures are scaled so that they all have the same resolution, or the texture array is dimensioned for the largest texture, with smaller textures occupying only a portion of the layer.

Both options have substantial drawbacks. Scaling textures changes the distribution of pixels per surface. Because we use the texture as our sampling raster, the distribution of sampled points also changes. Alternatively, if the textures don't fill the full resolution, built-in WebGL functions such as texture wrapping and pixel interpolation will fail for edge cases.

In our implementation, we opted for the latter option. Achieving a uniform sample density was more critical than avoiding texture wrapping or pixel interpolation issues, which can be explicitly handled in shaders. Additionally, edge cases are infrequent when the texture mapping is designed to stay within boundaries.

4 DISCUSSION

We implemented our proposed rendering method and different BVHs within a web environment. Thereby, our prototype uses WebGL and all performance and quality metrics were measured in the Microsoft Edge browser. We performed tests on a laptop NVIDIA GTX 1650 with an Intel Core i7-9750H and a desktop NVIDIA GTX 1080 Ti with an AMD Ryzen 9

3950X. In the following, we will examine how introducing GPU-based TBVH and MTBVH improves performance and discuss and compare visual quality with a reference implementation in Unity.

4.1 Performance

To evaluate the performance of our prototype, we measured the time needed to pre-calculate the global illumination and render the light maps. In particular, we ignore the time needed to load all assets and to generate the BVH, as these procedures are not related to the actual rendering method itself.

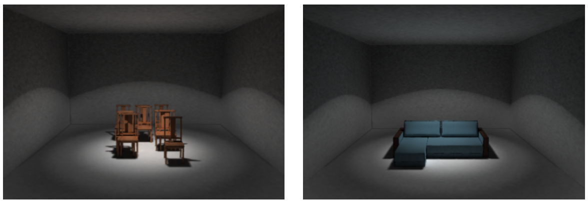


Figure 11: *Test cases chairs_05 (left), with 5 chairs, and sofa_01 (right), with one sofa, rendered with our render method under a single point light.*

We defined a set of scenes and settings to measure the performance improvements of BVHs for different numbers of polygons and rendered pixels (Table 1). The test cases contained a number of chairs (1, 3, 5, or 10) or a single sofa (Figure 11). Each scene was tested on the two aforementioned GPUs with no BVH, TBVH or MTBVH, resulting in 30 test cases. For each test case we recorded ten measurements of the render times.

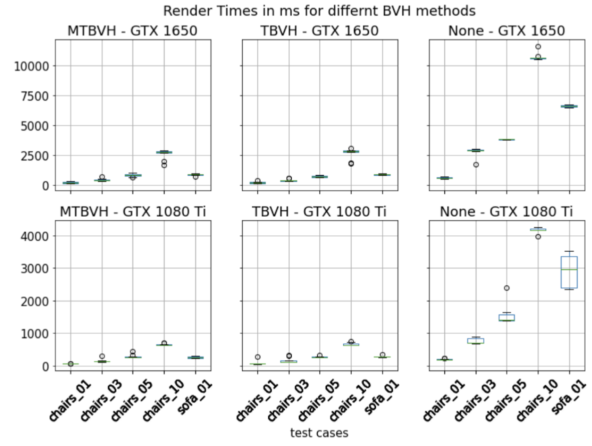


Figure 12: *Render Times in ms for different BVH methods. The number of chair models between each test case is increased, from 1 to 3 to 5 to 10. The sofa_01 test case is the exception, it contains only a sofa (see also Figure 11).*

The box-plot in Figure 12 shows the render times for the different BVH methods across our test cases. BVHs reduce the render times substantially. The increase in render times across test cases cannot be attributed to the increase in polygons alone. Each additional object requires an additional set of light maps to be rendered, increasing the total number of rendered pixels (Table 1). For example, *chairs_5* and *sofa_01* have a similar number of rendered pixels but *sofa_01* consists of twice the number of polygons. Comparing these two test cases provides the best estimation of the effect BVHs have on their own.

Our measurements reveal that the difference between TBVH and MTBVH is minor in our test cases (Table 1). MTBVH frequently leads by a few *ms*, especially for larger scenes, but the difference is negligible. The determination of the overall direction of the ray in our implementation of MTBVH probably causes branching issues on the GPU execution, leading to a longer render time.

4.2 Graphics

In order to evaluate the correctness of our rendering method, we created an equivalent scene in Unity. We then used Unity's built-in light baking pipeline to generate light maps with similar constraints as our own rendering method. Specifically, we allowed only one single reflection per sampled ray. Notably, we used Unity in our case as an offline renderer - the baking of the light maps took Unity's engine between one and three minutes for each of these scenes. The results are shown in 13, 14 and 15.

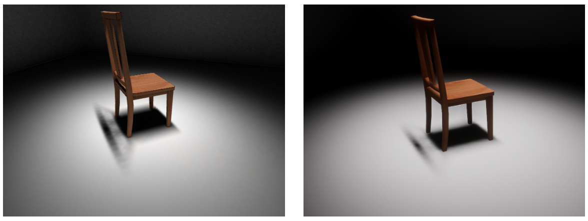


Figure 13: *Visual comparison of shadows between our render method (left) and Unity (right).*

We can confirm the presence of many of the expected global illumination effects. The walls and ceiling are indirectly lit by the area on the floor that is illuminated by the lights. The shadow of the chair is soft due to the influence of multiple lights (Figure 14). Also, there are coloured reflections based on the surface colour of the cube in Figure 15.

Our method produces more detailed soft shadows in Figure 13 than Unity's light maps, which have lesser resolution in principle. Thus, our renderer can represent the shadow gradient at a more granular level, resulting in improved quality.

The employment of a Gaussian filter causes colour bleeding on the edges. In Figure 14, the edges of the chair have small black seams. During both, direct and indirect render passes, the gaps between the texture's faces remain blank. During the blur pass, these gaps bleed into the visible area of the texture. However, these artefacts can easily be prevented by using a more

			Average Render Times per Test Case in ms					
			GTX 1650			GTX 1080 TI		
Test Case	# Pixels	# Polygons	MTBVH	TBVH	No BVH	MTBVH	TBVH	No BVH
chairs_01	459k	252	226	**222**	618	**61**	81	206
chairs_03	590k	732	446	**412**	2810	**154**	168	760
chairs_05	721k	1212	835	**732**	3833	289	**272**	1553
chairs_10	1049k	2412	**2616**	2675	10708	**644**	662	4172
sofa_01	655k	2036	**887**	908	6617	**263**	279	2909

Table 1: *Performance and metrics for each of our test cases in Figure 12. The fastest render times for each test case are indicated in bold.*

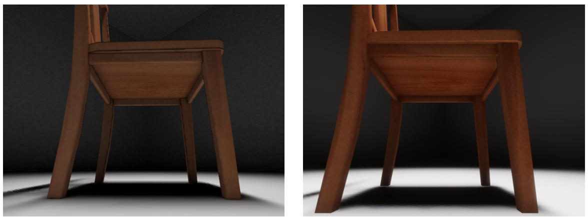


Figure 14: *Visual comparison of indirect lighting between our render method (left) and Unity (right).*

sophisticated filter, such as a bilateral filter. Additionally, the wall textures in Figure 14 are noisier than their Unity equivalents. This is most likely because the Unity light map method utilises more samples in this case and a more advanced denoising technique.

Using a Gaussian filter is not physically accurate. However it impacts the quality of diffuse lighting minimally, especially in low frequency areas and reduces the number of required rays, increasing performance. Nonetheless, accurate simulations are possible by disabling the Gaussian filter and increasing the ray count.

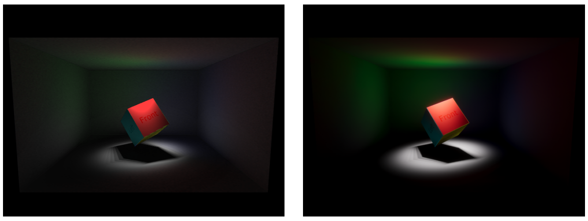


Figure 15: *Visual comparison of coloured reflections between our render method (left) and Unity (right).*

Figure 14 demonstrates the effects of the indirect pass on the underside of the chairs. The chair rendered with our method is more matted and less colourful, but the overall shading is fairly similar to the chair generated with Unity. In Figure 15, we rendered a separate scene containing a test cube only. This image depicts the impact of diffuse coloured surface reflections. Again, the version rendered with Unity is more colourful and less noisy, but the effect is similar in both versions.

One flaw that is visible in Figure 16 is that the floor is completely dark outside the directly lit area. As the walls and the ceiling are not directly lit, there is no lit surface from the direct light pass that could reflect onto the floor. By allowing up to two bounces, the floor is also illuminated. However, with our rendering method, this results in nearly twice the render time for the same scene.

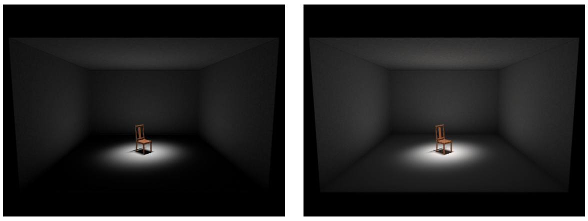


Figure 16: *Comparison of our results rendered with a single indirect light pass (left) and two indirect light passes (right).*

Overall, we have shown that our rendering method delivers similar results to a modern state-of-the-art renderer, while our performance is significantly higher. The baking of the light map took Unity's engine between one and three minutes for each of these scenes. Nevertheless, our method still suffers some minor visual artefacts inherent to our approach, which are the price for the higher performance.

5 CONCLUSION & FUTURE WORK

We presented a rendering approach for fast global illumination on the GPU on the web using WebGL. Our approach enables the simulation of physically accurate light propagation of real lamp data (i.e., IES-data). We achieved high performance by implementing BVHs on the GPU to quickly find ray-triangle intersections. Thereby, static light information is stored in light maps. In this way, a scene can be adjusted and modified before being re-rendered within a second to enable easy design and interactive navigation in web environments, such as configuration tools in the light manufacturing industry.

In this way, we achieve visual results comparable to state-of-the-art offline-renderers with only minor visual shortcomings. However, our approach is significantly faster (seconds vs. minutes).

Future work includes the investigation of more advanced denoising methods such as Non-Local Means with Joint Filtering (JNLM), RadeonPro, Open Image

Denoiser (OIDN) or OptiX [Uni23; God23] to improve visual results. This would probably also allow a reduction in the resolution of light maps, which would improve performance even further. Adapting the rendering method to progressive rendering of light maps, similarly to the works by Lesar et al. and Nilsson et al. [LBM18; NO18], would also allow for intermediate results while further details are added successively as the user navigates the scene. Finally, we plan to incorporate technologies such as Deep-Learning Super Sampling (DLSS) for improved anti-aliasing and importance sampling to increase quality and performance even further.

6 ACKNOWLEDGEMENTS

This work was supported by the project LiSi, funded by the Forschungsfonds Aargau and the company Ribag AG. We would like to thank Dominik Hausherr and Andreas Richner from Ribag AG for the valuable collaboration.

REFERENCES

[Bur20] Andrew Burnes. *Whitepaper: NVIDIA Ampere GA102 GPU Architecture*. Tech. rep. NVIDIA, 2020.

[CL21] Ma Cheng and Wang Lu. “A Fast Light Baking System for Mobile VR Game Based on Edge Computing Framework”. In: *Proceedings of the 2021 ACM International Conference on Intelligent Computing and its Emerging Applications*. 2021, pp. 176–181.

[Con+11] John Congote, Alvaro Segura, Luis Kabongo, Aitor Moreno, Jorge Posada, and Oscar Ruiz. “Interactive visualization of volumetric data with WebGL in real-time”. In: *Proceedings of the 16th International Conference on 3D Web Technology*. Web3D ’11. Paris, France: Association for Computing Machinery, 2011, pp. 137–146. ISBN: 9781450307741. DOI: `10.1145/2010425.2010449`.

[DBB18] Philip Dutre, Philippe Bekaert, and Kavita Bala. *Advanced global illumination*. CRC Press, 2018.

[God23] Godot. *Using Lightmap global illumination*. Nov. 30, 2023. URL: `https://docs.godotengine.org/en/stable/tutorials/3d/global_illumination/using_lightmap_gi.html` (visited on 02/08/2024).

[Gor+84] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. “Modeling the Interaction of Light between Diffuse Surfaces”. In: *SIGGRAPH Comput. Graph.* 18.3 (Jan. 1984), pp. 213–222. ISSN: 0097-8930. DOI: `10.1145/964965.808601`.

[GS87] Jeffrey Goldsmith and John Salmon. “Automatic Creation of Object Hierarchies for Ray Tracing”. In: *IEEE Computer Graphics and Applications* 7.5 (1987), pp. 14–20. DOI: `10.1109/MCG.1987.276983`.

[Hac15] Toshiya Hachisuka. *Implementing a Photorealistic Rendering System using GLSL*. 2015. arXiv: `1505.06022 [cs.GR]`.

[Ill91] Illuminating Engineering Society. *IES Standard File Format for Electronic Transfer of Photometric Data and Related Information*. Tech. rep. IES Computer Committee and others, 1991.

[Kaj86] James T. Kajiya. “The rendering equation”. eng. In: *Computer graphics (New York, N.Y.)* 20.4 (1986), pp. 143–150. ISSN: 0097-8930.

[Laf96] Eric Lafortune. “Mathematical models and Monte Carlo algorithms for physically based rendering”. In: *Department of Computer Science, Faculty of Engineering, Katholieke Universiteit Leuven* 20.74-79 (1996), p. 4.

[LBM18] Žiga Lesar, Ciril Bohak, and Matija Marolt. “Real-time interactive platform-agnostic volumetric path tracing in webGL 2.0”. In: *Proceedings of the 23rd International ACM Conference on 3D Web Technology*. Web3D ’18. Poznań, Poland: Association for Computing Machinery, 2018. ISBN: 9781450358002. DOI: `10.1145/3208806.3208814`.

[Lin+22] Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. “Generalized resampled importance sampling: foundations of ReSTIR”. In: *ACM Trans. Graph.* 41.4 (July 2022). ISSN: 0730-0301. DOI: `10.1145/3528223.3530158`.

[Luk+13] Christian Luksch, Robert F Tobler, Ralf Habel, Michael Schwärzler, and Michael Wimmer. “Fast light-map computation with virtual polygon lights”. In: *Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games*. 2013, pp. 87–94.

[LW93] Eric P Lafortune and Yves D Willems. "Bidirectional path tracing". In: (1993).

[NO18] Martin Nilsson and Alma Ottedag. "Realtime path tracing of small scenes using WebGL". In: (2018).

[ÖA17] Bekir Öztürk and Ahmet Oğuz Akyüz. "Semi-dynamic light maps". In: *ACM SIGGRAPH 2017 Posters*. 2017, pp. 1–2.

[Ras+10] Jim Rasmusson, Jacob Ström, Per Wennersten, Michael Doggett, and Tomas Akenine-Möller. "Texture compression of light maps using smooth profile functions". In: *Proceedings of the Conference on High Performance Graphics*. 2010, pp. 143–152.

[Rib24] Ribag AG. *Collections*. Feb. 5, 2024. URL: `https://www.ribag.ch/en/products/collections` (visited on 02/05/2024).

[Rie+23] Manuel Riedi, Alexander Legath, Luca Fluri, and Hilko Cords. *Configuration and Simulation Tool for Lighting Systems*. Online-Tool. Aug. 2023. URL: `https://iit.cs.technik.fhnw.ch/lichtsystem-konfigurator/` (visited on 02/29/2024).

[STØ05] Lars Ole Simonsen, Niels Thrane, and P Ørbæk. "A comparison of acceleration structures for GPU assisted ray tracing". In: *Master's thesis, University of Aarhus* (2005).

[Uni23] Unity. *The Progressive Lightmapper - Unity Documentation*. 2023. URL: `https://docs.unity3d.com/Manual/progressive-lightmapper.html` (visited on 02/08/2024).

[Vit+21] Nick Vitsas, Anastasios Gkaravelis, Andreas A Vasilakis, and Georgios Papaioannou. "WebRays: Ray tracing on the web". In: *Ray Tracing Gems II: Next Generation Real-Time Rendering with DXR, Vulkan, and OptiX* (2021), pp. 281–299.

[WP22] Chris Wyman and Alexey Panteleev. "Rearchitecting spatiotemporal resampling for production". In: *Proceedings of the Conference on High-Performance Graphics*. HPG '21. Goslar, DEU: Eurographics Association, 2022, pp. 23–41. DOI: `10.2312/hpg.20211281`.

Improving Image Reconstruction using Incremental PCA-Embedded Convolutional Variational Auto-Encoder

Amir Azizi
CYENS CoE
Nicosia
CYPRUS
a.azizi@cyens.org.cy

Panayiotis Charalambous
CYENS CoE
Nicosia
CYPRUS
p.charalambous@cyens.org.cy

Yiorgos Chrysanthou
CYENS CoE
Nicosia
CYPRUS
y.chrysanthou@cyens.org.cy

ABSTRACT

Traditional image reconstruction methods often face challenges like noise, artifacts, and blurriness, requiring handcrafted algorithms for effective resolution. In contrast, deep learning techniques, notably Convolutional Neural Networks (CNNs) and Variational Autoencoders (VAEs), present more robust alternatives. This paper presents a novel and efficient approach for image reconstruction employing Convolutional Variational Autoencoders (CVAEs). We use Incremental Principal Component Analysis (IPCA) to enhance efficiency by discerning and capturing significant features within the latent space. This model is integrated into both the encoder and sampling stages of CVAEs, refining their capability to generate high-fidelity images. Our incremental strategy mitigates scalability issues associated with traditional PCA while preserving the model's aptitude for identifying crucial image features. Experimental validation utilizing the MNIST dataset showcases noteworthy reductions in processing time and enhancements in image quality, underscoring the efficacy and potential applicability of our model for large-scale image generation tasks.

Keywords

Image Processing, Image Reconstruction, Principal Component Analysis, Convolutional Variational Auto-Encoders

1 INTRODUCTION

Generative models like Variational Autoencoder (VAE) are a significant advancement in deep learning, using probabilistic approaches to generate new data from existing datasets.VAEs,consisting of an encoder and decoder pair, capture key properties [CGD$^+$20, BTLLW21].VAEs have a wide range of applications, from image [LSM$^+$21, YYSL16, WX21] ,video [YZAS21, WRO21, ZLS$^+$21, DLW$^+$22], text [LPL21, YDML21, SWL$^+$21, ZDYC21] and music generation [JY23, WT22, WY21] to anomaly detection [NYW20, LCB$^+$20], and they are easier to train than their competitors, such as Generative Adversarial Networks (GANs) [BTLLW21].They ensure training stability by offering a greater range of realistic and varied facts [DB21].CVAEs integrates the functionalities of VAEs with those of a CNNs, constituting a specialized form of deep generative model [YKK21].CVAEs possess an architecture that leverages CNNs as both encoders and decoders, harnessing the spatial relation-capturing capabilities of CNNs and the fidelity in data generation characteristic of VAEs [SDRM21].This method demonstrates exceptional proficiency in processing grid-based data, especially images, leveraging the spatial understanding capabilities inherent in CNNs [BLD22]. CVAEs stand as foundational components across a diverse range of applications [KSZ$^+$21, WML21, LPC22, CQWZ21], spanning image synthesis, reconstruction, and anomaly detection. Their robustness consistently delivers stable, realistic, and semantically coherent outcomes [JJ21].In Figure 1, the structure of CVAEs and the interaction between their components are illustrated.This paper is dedicated to enhancing both the efficiency and quality of image reconstruction through the utilization of CVAEs.
Our contributions are summarized as follows:

1. Introducing a novel method that combines IPCA with CVAEs, enhancing the sampling and encoding stages.
2. considerably reduces processing time, thus boosting the algorithm's overall efficiency.
3. Demonstrating substantial enhancements in image quality by implementing our proposed algorithm on MNIST datasets.

The paper is structured as follows: Section 2 delves into the related works; in Section 3, we outline our novel approach and methodology. Section 4 is our Experimental results. Finally Section 5 , discusses the research challenges, limitations, and potential future directions.

https://www.doi.org/10.24132/CSRN.3401.12

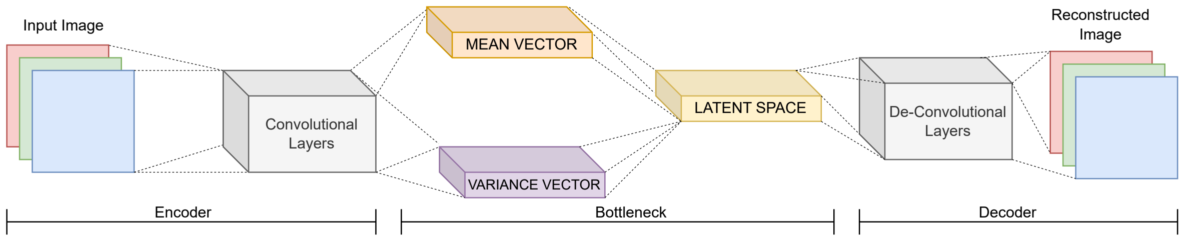


Figure 1: CVAEs utilize convolutional layers to encode input images, compress information into latent space, and decode samples, enabling efficient data representation and generation of new samples.

2 RELATED WORKS

The VAE, introduced by Kingma et al. [KW13].They applied variational Bayesian inference principles for image generation.It consists of two neural networks,Figure 2 shows an inference network for generating latent variable distributions, and a generation network for approximation.Deep learning and Artificial Intelligence have revolutionized image reconstruction and image generation from diverse data sources, with Variational Auto-encoders emerging as key methodologies for high-quality image generation [EEAMT22, LSC20, IB23, WCQ23]. The Conditional Variational Autoencoder (CVAE),which is shown in Figure 3, improves the unsupervised model by incorporating category information labels, transforming it into a semi-supervised mode within the CVAE framework [SLY15, HNW21].The Very Deep Variational Autoencoder (VDVAE) is an extension of the standard Variational Autoencoder (VAE) with increased depth in its neural network architecture [Chi20]. This depth allows for more intricate hierarchical representation of latent variables, enabling high-fidelity and nuanced reconstructions while navigating large-scale dataset complexities. VDVAE-SR [CHW$^+$23] uses transfer learning on pretrained VDVAEs to improve image super-resolution. Fusion-VAE [DVZN22] is a novel deep hierarchical variational autoencoder for generative image fusion, outperforming traditional methods. VAEL, a neuro-symbolic generative model, combines VAEs and probabilistic Logic Programming strengths, enabling the generation of new data points satisfying logical constraints while capturing complex relationships VAEL architecture illustrated in Figure 4 VDVAEs, while powerful for image generation, face computational complexity due to high-dimensional data, but ongoing advancements aim to improve efficiency and scalability [ASY$^+$22, VK20].

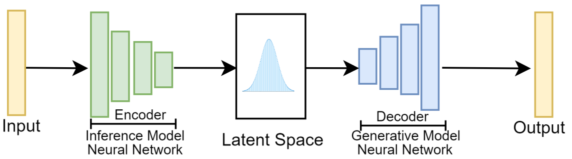


Figure 2: The model employs variational Bayesian inference principles for image generation, utilizing two neural networks for inference and generation to capture data structure and variability.

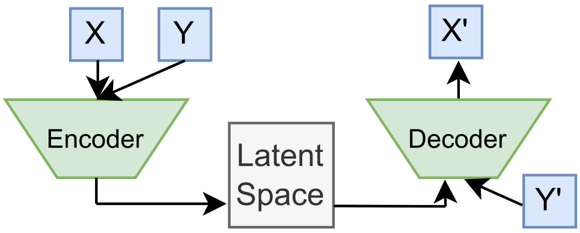


Figure 3: CVAE, similar to VAE,adds category information to input data, maximizing logarithmic marginal likelihood and lower bound function of variation, but not solving image blur or high synthetic data accuracy.

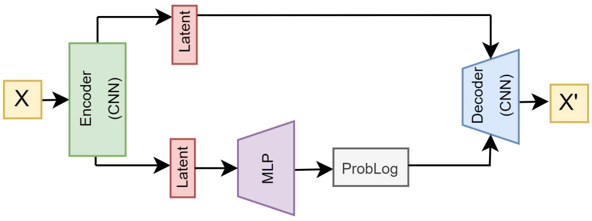


Figure 4: The VAEL model consists of three components: an encoder, a Prob-Log program, and a decoder, which compute latent variables, parameterize programs, and reconstruct images.

3 PROPOSED METHOD

The study uses IPCA and CVAEs to enhance image reconstruction. CVAEs extract hierarchical features from images using convolutional layers, preserving spatial information. They encode input images into a latent space, capturing meaningful representations of digits in a lower-dimensional space. Our novel approach lies in integrating IPCA within the training paradigm of CVAEs for image reconstruction. IPCA is used after encoding to reduce dimensionality in data obtained from the CVAE. It refines the latent space representation before decoding, enhancing efficiency and retaining meaningful information for the decoding step,

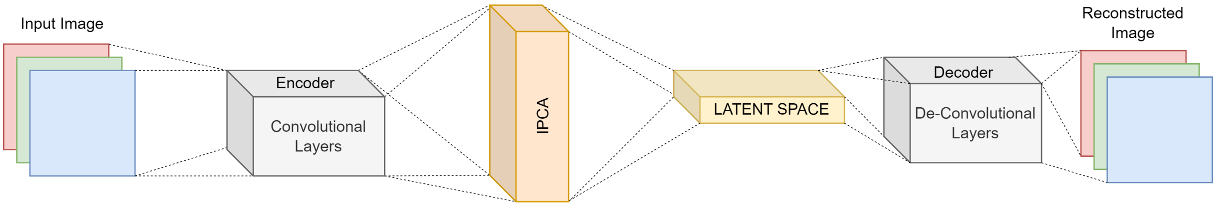


Figure 5: IPCA-CVAE: integrates incremental PCA and Convolutional Variational Auto-encoders to improve image reconstruction, enhancing efficiency and retaining meaningful information for decoding.

thereby optimizing the handling of latent representations The workflow of our proposed method , as illustrated in Figure 5 is structured as follows:

1. CVAEs are used to improve image reconstruction by extracting hierarchical features from images using convolutional layers. These layers preserve spatial information, ensuring image structural integrity. By encoding input images into a latent space, CVAEs capture meaningful representations of digits, allowing the model to learn essential features while reducing data dimensionality.
2. IPCA is a technique used in the training paradigm of CVAEs to optimize image reconstruction by refining the latent space representation obtained from the encoding step, thereby enhancing efficiency and retaining crucial information for the decoding step.
3. The decoder uses deconvolutional layers, also known as transposed convolutional layers or upsampling layers, to transform the refined latent representation from IPCA into a high-dimensional feature map. This process optimizes the handling of latent representations, enhancing the efficiency and accuracy of image reconstruction tasks. The integrated approach ensures meaningful information is retained and utilized throughout the decoding process.

The integration of IPCA enhances the CVAE model's efficiency by refining latent space representation before decoding, reducing data dimensionality while retaining crucial information for accurate image reconstruction. This approach enhances overall image reconstruction task performance and provides a novel solution for efficient dimensionality reduction and preservation of meaningful information. It is applicable across different domains and datasets, offering versatility and effectiveness in handling latent representations. Integrating IPCA into CVAEs training provides a robust and efficient solution for image reconstruction, contributing to image processing and deep learning techniques.

4 RESULTS

The study utilizes a PC with a Core i7-8700K CPU, 16GB RAM, and Google Colab for computational resources. The results of this work are presented in two parts:
i) Analyzing the impact of the IPCA on processing time.
ii) Exploring the relationship between re-constructed image quality and the proposed method.
We use two main datasets: MNIST [LC10] and Fashion-MNIST [XRV17].MNIST has 70,000 gray-scale images of ten handwritten digit classes. Fashion-MNIST is similar but covers ten categories of clothing, including items like t-shirts, trousers, and pullovers. To enhance compression, we utilized three models: CVAE, PCA-CVAE, and IPCA-CVAE. The architecture of the encoder and decoder involved a consistent structure. The encoder started with an input layer for gray-scale images, followed by two Conv2D layers implementing 32 and 64 filters. A flattening layer condensed the output, leading to a dense layer with 16 neurons activated by ReLU. Two dense layers generated parameters for shaping the variational distribution within the latent space. The decoder played a crucial role in the reconstruction process, starting with an input layer for 20-dimensional latent space representation. The output was rearranged into a (7, 7, 64) structure, enabling two Conv2DTranspose layers to progressively upscale the encoded latent space. The output layer, a Conv2DTranspose, reconstructed the original image dimensions, concluding the decoding phase.

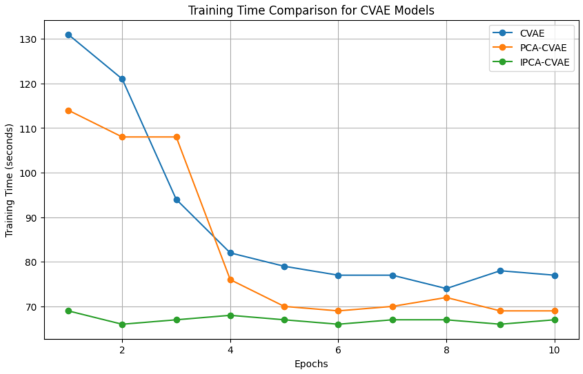


Figure 6: Training Time Comparison for CVAE Models on MNIST digits dataset.

In Figure 6, a comparative analysis of various methodologies is presented, focusing on the training duration. The visual representation of PCA-CVAE and CVAE

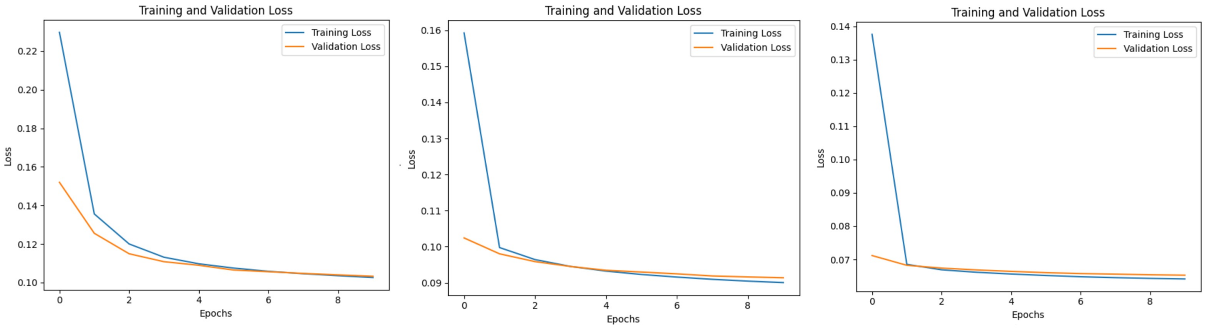


Figure 7: Training and Validation Loss for CVAE(left),PCA-CVAE(middle) ,IPCA-CVAE(right) on MNIST digits dataset

methodologies reveals significant differences. PCA-CVAE offers sustained functionality beyond the fourth epoch and reduces processing time, while IPCA-CVAE demonstrates accelerated training process completion, which is beneficial for large image databases. The proposed IPCA-CVAE method shows heightened efficiency, especially in scenarios with voluminous image collections, demonstrating its superior performance.

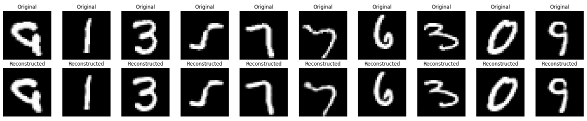


Figure 8: Reconstructed Images based on IPCA-CVAE on MNIST digits dataset.

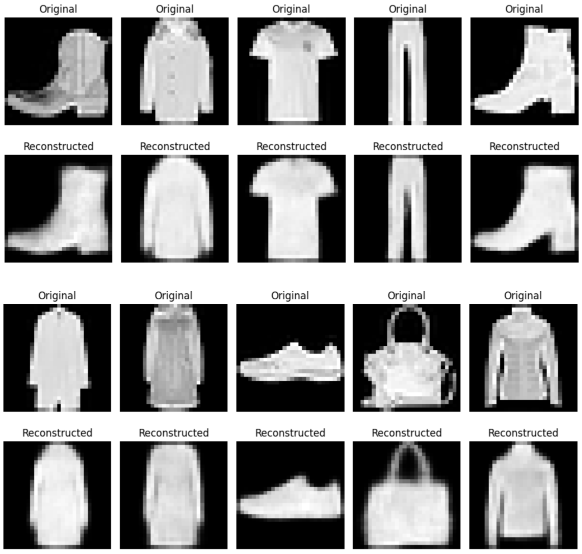


Figure 9: Reconstructed Images based on IPCA-CVAE on MNIST fashion dataset.

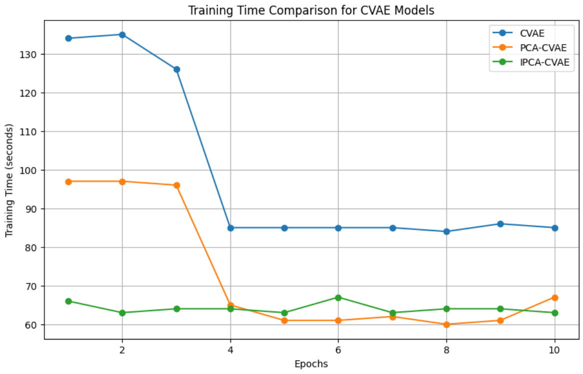


Figure 10: Training Time Comparison for CVAE Models on MNIST fashion dataset.

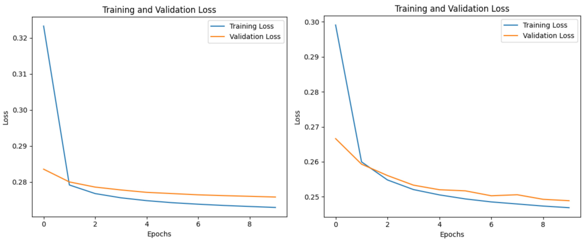


Figure 11: Training and Validation Loss for CVAE(left),IPCA- CVAE(right) on MNIST fashion dataset.

Figure 7 provides a comparison of Training and Validation Loss across various CVAE methods. Figure 8 and Figure 9 display reconstructed images generated using the IPCA-CVAE method applied to the MNIST datasets. Furthermore, Figure 10 highlights a noticeable discrepancy in training duration on the MNIST Fashion dataset, emphasizing the tangible advantages of employing the proposed algorithm. Significantly, the discernible reduction in processing time attests to the algorithm's efficacy in optimizing training efficiency. Additionally, Figure 11 presents a comparative analysis between CVAE and IPCA-CVAE methods. The left image corresponds to the CVAE method, the middle image represents PCA-CVAE, and the right image pertains to IPCA-CVAE. Notably, the Figure distinctly reveals a more pronounced reduction in loss during the training process in the proposed method compared to alternative approaches.

4.1 Metrics Evaluation

In this section, we quantitatively assess the performance of our proposed method, CVAE-IPCA, on the

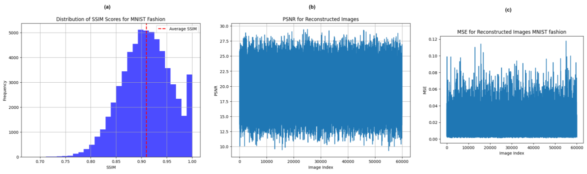


Figure 12: Metrics Evaluation after epoch 10 :(a) SSIM, (b) PSNR, and (c)MSE. The images demonstrate an average SSIM of 0.9097, PSNR of 21.0432, and an MSE of 0.0133. These values reflect the effectiveness of the proposed method.

MNIST fashion dataset for image reconstruction tasks. We employ three widely used image quality metrics: Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), and Structural Similarity Index (SSIM). PSNR measures the ratio between the maximum possible power of a signal and the power of corrupting noise, defined as:

$$PSNR = 10 * log_{10}(peakval^2)/MSE$$

, where peakval is the maximum possible pixel value of the image.MSE calculates the average squared difference between the original and reconstructed images, which is defined for two images such as $\hat{g}(n,m)$ and $g(n,m)$as below :

$$MSE = 1/MN * (\sum_{n=0}^{M} \sum_{m=1}^{N}) [\hat{g}(n,m) - g(n,m)]^2$$

Lastly, SSIM evaluates the similarity between two images based on luminance, contrast, and structure, and is defined as

$$SSIM(x,y) = [l(x,y)]^{\alpha} * [c(x,y)]^{\beta} * [s(x,y)]^{\gamma}$$

Here, L denotes luminance, C represents contrast, and S signifies structure. These parameters help gauge brightness, intensity range differences, and local pattern similarities between images, respectively.α ,β, and γ are constants used for computation stability. results of metrics evaluation illustrated in Figure 12.

4.2 Ablation Study

In our ablation study, we systematically varied three key parameters: learning rate, batch size, and latent space dimensionality, to understand their impact on the performance of our proposed method.

1. Learning Rate Variation: We tested learning rates of 0.001, 0.0001, and 0.01 to analyze their effect on model convergence and performance metrics such as PSNR, SSIM, and MSE. This experiment provided insights into how different learning rates influence training dynamics and model effectiveness.
2. Batch Size Variation We explored batch sizes of 64, 128, and 256 to assess their impact on training stability and computational efficiency. By observing training speed and model accuracy under different batch size settings, we gained an understanding of their trade-offs and implications.
3. Latent Space Dimensionality Variation : We varied latent space dimensions between 20, 50, and 100 to examine how they affect the model's ability to capture and represent input data features. Analyzing reconstruction quality across different latent space sizes provided insights into the dimensionality's role in feature representation.

The default parameter values for our method are learning rate = 0.001, batch size = 128, and latent space dimension = 20. These values serve as reference points for comparison against the variations tested in our ablation study.

4.2.1 Findings

1. Our investigation found that employing a learning rate of 0.0001 failed to yield any notable improvements. Although a rate of 0.01 showed some enhancements compared to 0.0001, the most consistent and optimal outcomes were achieved with a learning rate of 0.001.
2. Changing the batch size did not lead to any improvement in the results. However, the alterations in batch size yielded better outcomes compared to those from modifying the learning rate. Notably, the optimal results were observed with a batch size of 128. As anticipated, increasing the batch size to 64 resulted in longer processing times, while decreasing it to 256 reduced processing times.
3. Increasing the latent space dimensions to 100 yielded improved results. However, this enhancement came at the cost of significantly increased processing time. The results of the ablation study are shown in Table 1.

Figure 13 also presents the outcomes of PSNR.Figure 14 shows the reconstructed images with different parameters and finally, Figure 15 illustrates the results of SSIM, MSE, and training curves.

5 CONCLUSION AND FUTURE WORKS

This paper introduces a novel approach to image reconstruction using CVAEs.The efficiency of the model is optimized through IPCA, a technique used to identify and capture significant features in the latent space. This

Parameter	PSNR	SSIM	MSE	TT	Remarks
lr=0.01	20.439	0.8907	0.0167	1031	ld=20,bs=128
lr=0.001	21.0432	0.9097	0.0133	974	ld=20,bs=128
lr=0.0001	19.3043	0.8898	0.0193	1021	ld=20,bs=128
ld=20	21.0432	0.9097	0.01505	974	lr=0.001,bs=128
ld=50	20.8135	0.8992	0.0193	1255	lr=0.001,bs=128
ld=100	22.002	0.9101	0.0123	1245	lr=0.001,bs=128
bs=64	20.754	0.8998	0.0163	1182	lr=0.001,ld=20
bs=128	21.0432	0.9097	0.0133	1021	lr=0.001,ld=20
bs=256	20.9097	0.9001	0.0140	1084	lr=0.001,ld=20

Table 1: Summary of the Ablation Study(lr=Learning Rate , ld=Latent Dimension ,bs=Batch Size ,TT=Training Time(s))

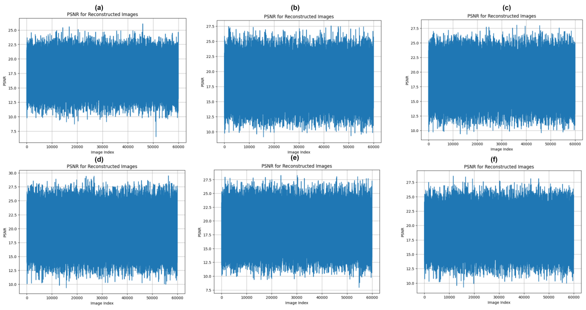


Figure 13: PSNR results (a)learning rate=0.0001, (b)learning rate=0.01 (c)=latent dimension=50 (d)latent dimension=100 (e)batch size=64 (f)batch size=256

Figure 14: Reconstructed Images (a)learning rate=0.0001, (b)learning rate=0.01 (c)=latent dimension=50 (d)latent dimension=100 (e)batch size=64 (f)batch size=256

incremental strategy addresses scalability concerns associated with traditional PCA while preserving the model's proficiency in identifying essential image features. Experimental results on the MNIST dataset show :

1. Reduces processing time and improves image quality.

2. Applicable in large-scale image generation tasks.

3. CVAEs' sensitivity to input data distribution variations.

Our proposed method demonstrates significant potential across various image-related tasks, extending beyond the datasets examined thus far. In object detection, denoising, and other applications, such as image classification using extensive datasets like CIFAR-10 and ImageNet, integrating IPCA-CVAE could enhance feature extraction and dimensionality reduction. This enhancement may lead to improved classification accuracy and resilience against variations in image content. Moreover, in tasks like style transfer and image synthesis, where generating realistic and diverse images is crucial, IPCA-CVAE's refined latent representations could offer finer control over visual attributes, enabling the creation of more compelling images. Additionally, in medical imaging tasks like tumor detection and segmentation, where handling high-dimensional images is common, IPCA-CVAE integration could help capture meaningful anatomical structures while reducing computational complexity and memory usage. Furthermore, in satellite imagery analysis for environmental monitoring and disaster management, IPCA-CVAE could facilitate efficient feature extraction and anomaly detection, enabling timely identification of critical changes in land cover and environmental conditions.

Moving forward, future research directions should focus on adapting the model to diverse and more complex datasets, exploring real-time optimization strategies, and improving the robustness of the CVAE framework. Additionally, incorporating domain-specific knowledge or constraints into the model may further refine its applicability for specific image reconstruction tasks.

6 ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 739578 and the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital

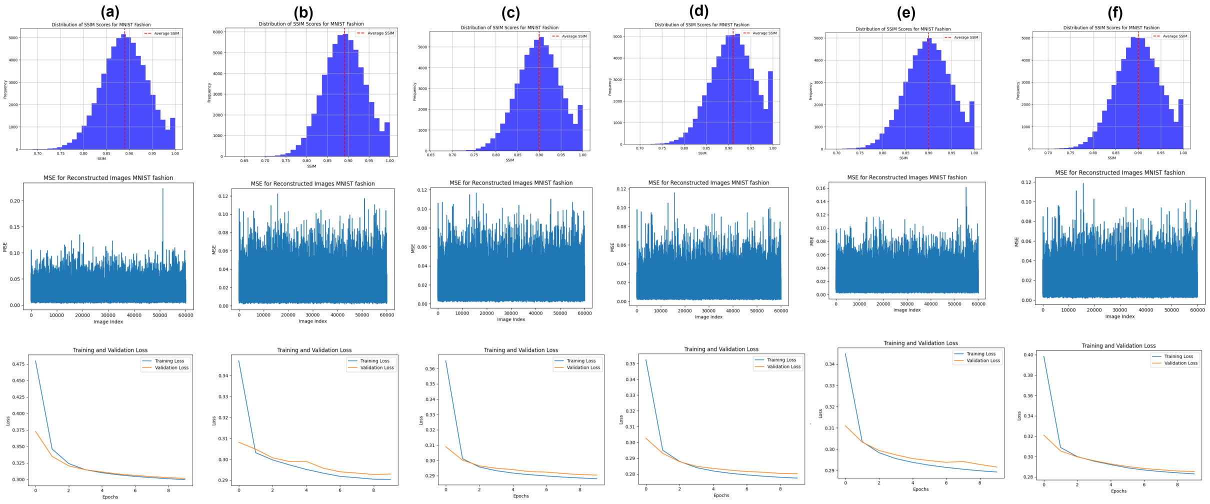


Figure 15: SSIM results(top) MSE results(middle) and Training curves(bottom) (a)learning rate=0.0001, (b)learning rate=0.01 (c)=latent dimension=50 (d)latent dimension=100 (e)batch size=64 (f)batch size=256

Policy. This work has received funding from the European Union's research and innovation program Horizon Europe under grant agreement No 10192889. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

7 REFERENCES

[ASY+22] Bilal Ahmad, Jun Sun, Qi You, Vasile Palade, and Zhongjie Mao. Brain tumor classification using a combination of variational autoencoders and generative adversarial networks. *Biomedicines*, 10(2):223, 2022.

[BLD22] Jichao Bao, Liangping Li, and Arden Davis. Variational autoencoder or generative adversarial networks? a comparison of two deep learning methods for flow and transport data assimilation. *Mathematical Geosciences*, 54(6):1017–1042, 2022.

[BTLLW21] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[CGD+20] Taylan Cemgil, Sumedh Ghaisas, Krishnamurthy Dvijotham, Sven Gowal, and Pushmeet Kohli. The autoencoding variational autoencoder. *Advances in Neural Information Processing Systems*, 33:15077–15087, 2020.

[Chi20] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

[CHW+23] Darius Chira, Ilian Haralampiev, Ole Winther, Andrea Dittadi, and Valentin Liévin. Image super-resolution with deep variational autoencoders. In Leonid Karlinsky, Tomer Michaeli, and Ko Nishino, editors, *Computer Vision – ECCV 2022 Workshops*, pages 395–411, Cham, 2023. Springer Nature Switzerland.

[CQWZ21] Dingliang Chen, Yi Qin, Yi Wang, and Jianghong Zhou. Health indicator construction by quadratic function-based deep convolutional auto-encoder and its application into bearing rul prediction. *ISA transactions*, 114:44–56, 2021.

[DB21] David Dehaene and Rémy Brossard. Re-parameterizing vaes for stability. *arXiv preprint arXiv:2106.13739*, 2021.

[DLW+22] Yiping Duan, Mingzhe Li, Lijia Wen, Qianqian Yang, and Xiaoming Tao. From object-attribute-relation semantic representation to video generation: A multiple variational autoencoder approach. In *2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2022.

[DVZN22] Fabian Duffhauss, Ngo Anh Vien, Hanna Ziesche, and Gerhard Neumann. Fusionvae: A deep hierarchical variational autoencoder for rgb image fusion. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision*

– *ECCV 2022*, pages 674–691, Cham, 2022. Springer Nature Switzerland.

[EEAMT22] Mohamed Elasri, Omar Elharrouss, Somaya Al-Maadeed, and Hamid Tairi. Image generation: A review. *Neural Processing Letters*, 54(5):4609–4646, 2022.

[HNW21] William Harvey, Saeid Naderiparizi, and Frank Wood. Conditional image generation by conditioning variational auto-encoders. *arXiv preprint arXiv:2102.12037*, 2021.

[IB23] Ashhadul Islam and Samir Brahim Belhaouari. Fast and efficient image generation using variational autoencoders and k-nearest neighbor oversampling approach. *IEEE Access*, 11:28416–28426, 2023.

[JJ21] Anyue Jiang and Behnam Jafarpour. Deep convolutional autoencoders for robust flow model calibration under uncertainty in geologic continuity. *Water Resources Research*, 57(11):e2021WR029754, 2021.

[JY23] Shulei Ji and Xinyu Yang. Emomusictv: Emotion-conditioned symbolic music generation with hierarchical transformer vae. *IEEE Transactions on Multimedia*, 2023.

[KSZ$^+$21] Adam R Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Sona Mokrá, and Danilo Jimenez Rezende. Nerf-vae: A geometry aware 3d scene generative model. In *International Conference on Machine Learning*, pages 5742–5752. PMLR, 2021.

[KW13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[LC10] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[LCB$^+$20] Shuyu Lin, Ronald Clark, Robert Birke, Sandro Schönborn, Niki Trigoni, and Stephen Roberts. Anomaly detection for time series using vae-lstm hybrid model. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4322–4326. Ieee, 2020.

[LPC22] Sang Min Lee, Sang-Youn Park, and Byoung-Ho Choi. Application of domain-adaptive convolutional variational autoencoder for stress-state prediction. *Knowledge-Based Systems*, 248:108827, 2022.

[LPL21] Ruizhe Li, Xutan Peng, and Chenghua Lin. On the latent holes of vaes for text generation. *arXiv preprint arXiv:2110.03318*, 2021.

[LSC20] Zhi-Song Liu, Wan-Chi Siu, and Yui-Lam Chan. Photo-realistic image super-resolution via variational autoencoders. *IEEE Transactions on Circuits and Systems for video Technology*, 31(4):1351–1365, 2020.

[LSM$^+$21] Kaikai Liu, Renjun Shuai, Li Ma, et al. Cells image generation method based on vae-sgan. *Procedia Computer Science*, 183:589–595, 2021.

[NYW20] Zijian Niu, Ke Yu, and Xiaofei Wu. Lstm-based vae-gan for time-series anomaly detection. *Sensors*, 20(13):3738, 2020.

[SDRM21] Kelathodi Kumaran Santhosh, Debi Prosad Dogra, Partha Pratim Roy, and Adway Mitra. Vehicular trajectory classification and traffic anomaly detection in videos using a hybrid cnn-vae architecture. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11891–11902, 2021.

[SLY15] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[SWL$^+$21] Huajie Shao, Jun Wang, Haohong Lin, Xuezhou Zhang, Aston Zhang, Heng Ji, and Tarek Abdelzaher. Controllable and diverse text generation in e-commerce. In *Proceedings of the Web Conference 2021*, pages 2392–2401, 2021.

[VK20] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020.

[WCQ23] Zhangkai Wu, Longbing Cao, and Lei Qi. evae: Evolutionary variational autoencoder. *arXiv preprint arXiv:2301.00011*, 2023.

[WML21] Hongzhuang Wu, Xiaoli Ma, and Songyong Liu. Designing multi-task convolutional variational autoencoder for radio tomographic imaging. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(1):219–223, 2021.

[WRO21] Jacob Walker, Ali Razavi, and Aäron van den Oord. Predicting video with vq-vae. *arXiv preprint arXiv:2103.01950*, 2021.

[WT22] Huiyao Wu and Maryam Tavakol. Musebar: Alleviating posterior collapse in recurrent vaes toward music generation. In *International Symposium on Intelligent Data Analysis*, pages 365–377. Springer, 2022.

[WX21] Yang Wu and Lihong Xu. Image generation of tomato leaf disease identification based on adversarial-vae. *Agriculture*, 11(10):981, 2021.

[WY21] Shih-Lun Wu and Yi-Hsuan Yang. Musemorphose: Full-song and fine-grained piano music style transfer with one transformer vae. *arXiv preprint arXiv:2105.04090*, 2021.

[XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[YDML21] Weijie Yuan, Linyi Ding, Kui Meng, and Gongshen Liu. Text generation with syntax-enhanced variational autoencoder. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[YKK21] Qien Yu, Muthu Subash Kavitha, and Takio Kurita. Mixture of experts with convolutional and variational autoencoders for anomaly detection. *Applied Intelligence*, 51:3241–3254, 2021.

[YYSL16] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 776–791. Springer, 2016.

[YZAS21] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.

[ZDYC21] Kun Zhao, Hongwei Ding, Kai Ye, and Xiaohui Cui. A transformer-based hierarchical variational autoencoder combined hidden markov model for long text generation. *Entropy*, 23(10):1277, 2021.

[ZLS$^+$21] Yizhou Zhou, Chong Luo, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. Vaeˆ 2: Preventing posterior collapse of variational video predictions in the wild. *arXiv preprint arXiv:2101.12050*, 2021.

ISSN 2464-4617 (print)
ISSN 2464-4625 (online)
Computer Science Research Notes - CSRN 3401
http://www.wscg.eu
WSCG 2024 Proceedings

Reevaluation in Rule-Based Graph Transformation Modeling Systems

Maxime Gaide[1] David Marcheix[2] Agnès Arnould[3]
Xavier Skapin[3] Hakim Belhaouari[3] Stéphane Jean[2]
[1]ISAE-ENSMA Poitiers, Université de Poitiers, LIAS, Poitiers, France
[2]Université de Poitiers, ISAE-ENSMA Poitiers, LIAS, Poitiers, France
[3]Université de Poitiers, Univ. Limoges, CNRS, XLIM, Poitiers, France
firstname.lastname@{[1,2]ensma, [3]univ-poitiers}.fr

Abstract

In this paper, we widen the naming problem studies to the rule-based graph 3D transformation modeling systems. We propose a persistent naming method taking advantage of the generalized maps' and graph transformation rules' formalization of simple operations. It enables a unique and homogeneous characterisation of entities in all dimensions. Most existing methods require tracking numerous topological entities and consider the persistent naming problem only from the parameters' modifications of a parametric specification standpoint. With our solution, not only the naming problem is tackled within the usual framework of parameters edition, but we also take the specification edition into account (addition, deletion and displacement of operations). Moreover, our solution makes use of directed acyclic graphs to represent the histories of topological entities and to track only the entities used in the parametric specification and the ones they originate from.

Keywords

Topology-based modeling; Graph transformation rules; Persistent Naming; Reevaluation; Generalized maps;

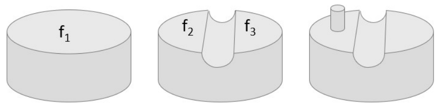
1 INTRODUCTION

The ability to generate multiple variants of an object during a construction process is becoming increasingly frequent in many application areas. Most of the time, tools and operations used to create those variants are dedicated to specific fields and the construction process is often both tedious and time-consuming. For example, in the field of Archaeology, remaining data found on the working field often represent only vestiges of ancients buildings. By means of 3D reproduction, archaeologists painstakingly develop a number of hypotheses they expect to test while being able to quickly model and visualize them [QB15]. CAD uses parametric history-based systems; such systems can be thought as dual structures with, on the one hand, the geometric model corresponding to the modelled object and, on the other hand, the successive operations (and their parameters) recorded during the construction process. This process can then be reevaluated after some slight modifications upon the operations, without starting all over again from the beginning. Nevertheless, creating complex objects always requires a substantial amount of time. Modeling buildings is also of interests for architects. Grammar-based procedural methods are commonly used to generate variations of constructions [HMV09; Mül+06]. But those grammars are based on a specific corpus of information which is difficult to transpose to other case studies.

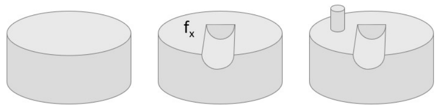
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or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

In this paper, we propose to use a rule-based graph 3D transformation formalism, and more specifically the Jerboa software [Bel+14], to make the development of dedicated modelers for specific applications easier. Rule-based languages for modeling are commonly used in a number of fields such as plant growth with L-systems [Lin74; BTG15], wood's internal structure [Ter+09], or virtual cities 3D models [ESR; Bei+10]. Contrary to other approaches, Jerboa is independent from any specific application field and does not require *ad hoc* operations to be manually coded. Simple operations are formally defined as rules within the Jerboa interface, thereby facilitating their rapid development. Furthermore, it guarantees the topological consistency of the underlying geometric model, regardless of the applied operations [Arn+22]. Jerboa is based on the generalized maps (or G-maps) topological model [Lie91; DL14]. This model represents a specific class of labelled graph and allows the homogeneous modeling of quasi-manifolds in any dimension. Number of applications already make use of Jerboa and/or G-maps in fields such as plant growth

https://www.doi.org/10.24132/CSRN.3401.13



(a) Initial evaluation



(b) Reevaluation

Figure 1: Parametric specification

[BTG15], architecture [Hor+09; ALS15], geology [HH00] or physics-based modeling [Ben+17]. Despite its advantages, Jerboa does not provide the mechanisms required to quickly reevaluate variations from a base model. Conversely, history-based parametric systems take advantage of the construction process recording to make reevaluations as fast and accurate as possible. Thus, our objective is to extend the capabilities of Jerboa by incorporating the mechanisms inherent in parametric systems. Any reevaluation of the parametric specification entails modifying the parameters of the operations. Those parameters are either geometric (such as the length of a groove) or references topological entities (vertices, edges, faces and so on) defined at an earlier stage of the modeling process. Modifying some operation parameters requires ensuring that the subsequent operations are still valid, even if their own parameters have been updated. This issue, known as *persistent naming*, is illustrated in Fig. 1. The initial parametric specification consists of three constructive operations (Fig. 1a): `1-Create-Cylinder(geo_param1); 2-RoundedGroove(f1, geo_param2); 3-CylindricalProtrusion(f2, geo_param3)`. When identifiers or pointers (*i.e.* concrete names) to the topological parameters of a geometric model (*e.g.* the identifier of the face `f2` as a parameter of the `CylindricalProtrusion` operation) are used as topological parameters, the issue of the persistence of these references at reevaluation comes up. For example, in Fig. 1b, the rounded groove's length is reduced. The face `f1` is not split anymore, unlike during the initial evaluation. Thus, neither face `f2` nor face `f3` are created: identifiers and pointers to the entities are obviously different and, therefore, the cylindrical protrusion can no longer be re-applied onto `f2`. Hence the necessity to use persistent identifiers as operations parameters, which make possible to unambiguously characterize entities and find their match at reevaluation. In Fig. 1, using a persistent name to characterize `f2` during the initial evaluation allows matching it with the face `fx` at reevaluation. Although persistent naming has been studied for decades in the CAD's field [Wu+01; MH05; Mar06; Bab10; Xue+16; FH18; Saf+20; CBS23; DZ24], to our knowledge two preliminary approaches have attempted to use graph transformation rules to tackle this issue [Car+19; Gai+23b]. In [Car+19], the authors propose to use *History Records* (HRs) to represent the history of any topological entity designated in parametric specification and *Matching Trees* (MT) to match this entity during reevaluation. This is an interesting initial theoretical approach based on graph transformation, but the history represented in HR is limited. In particular, no distinction is made between the entities at the origin of the designated entity and the evolution of the designated entity itself. Furthermore, some elements are omitted in HRs (such as the history of entities at the origin of the designated entity), which can lead to mismatches during reevaluation. In the concise poster paper [Gai+23b], the authors propose a full persistent naming mechanism based on graph transformation rules. They also propose to complete the histories of topological entities by taking their origins into account and integrate them in a reevaluation mechanism. In this paper, we base our work on [Gai+23b] to integrate the complete history of topological entities. Our contribution is twofold. First, we widen the naming problem studies to the rule-based graph transformation modeling systems. Second, we integrate the mechanisms of reevaluation for parametric systems into Jerboa.

We propose a persistent naming method taking advantage of the rule-based formalization of operations and their ability to precisely describe the history of topological entities, such that these entities are uniquely and homogeneously characterized for all dimensions. Most existing methods require tracking numerous topological entities and consider the persistent naming problem only through the prism of parameters modifications from a parametric specification standpoint[CH95; Wu+01]. Our solution tracks only the entities used in the parametric specification and the ones they originate from. Moreover, not only the naming problem is tackled within the usual framework of parameter edition, but we also take the specification edition (*i.e.* adding, deleting and reordering of operations) into account.

In section 2, we present the necessary concepts to carry out persistent naming mechanisms within the framework making use of graph transformation rules. We focus on G-maps, Jerboa's rules, and on how to automatically detect topological changes (creation, deletion, split, merging, modification) that may occur upon any rule application. Section 3 is dedicated to the data structures used by parametric specifications and persistent naming. Section 4 describes how a parametric specification is evaluated or reevaluated through directed acyclic graphs which track the evolutions of topological entities and the ones they originate from. Section 5 presents the matching process between evaluated and reevaluated entities. We conclude in section 6 and present the main directions of our future works.

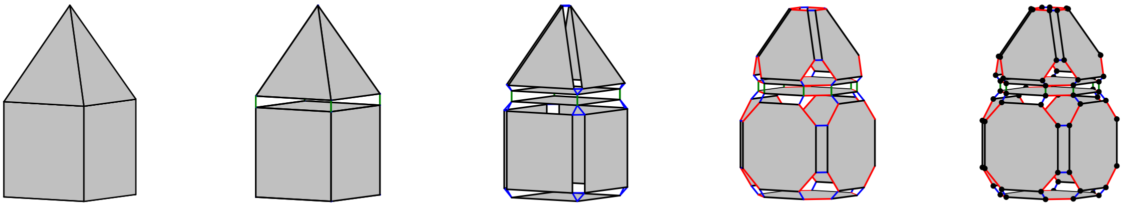
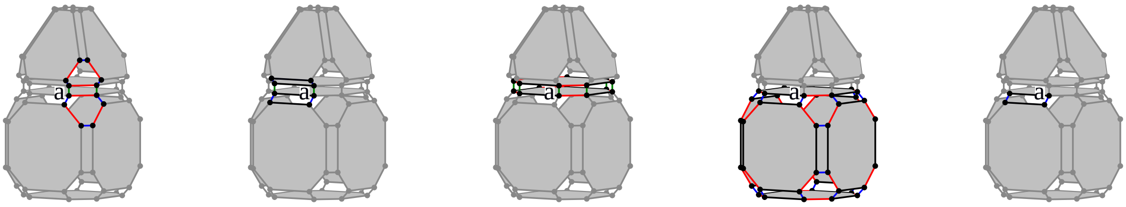

(a) (b) (c) (d) (e)

Figure 2: Cell decomposition of a geometric 3D object



(a) (b) (c) (d) (e)

Figure 3: Orbit decomposition of a geometric 3D object

2 MAIN CONCEPTS

In this section, we present the generalized maps, graph transformation rules and their subsequent concepts which are necessary to the understanding of our contribution.

2.1 Generalized maps

Generalized maps (or G-maps) [Lie91; DL14] allow the representation of manifold geometric objects (with or without boundaries), based on a cellular n-dimensional topological structure. The representation of an object as a G-map comes intuitively from its decomposition into topological cells (vertices, edges, faces, volumes, and so on). For example, the 3D topological object (Fig. 2a) can be decomposed into two volumes (Fig. 2b): a cube and a pyramid. These volumes are *linked* along their common faces with a 3-link, drawn in green. The index "3" means that the link connects two 3-dimensional (possibly a single one) volumes. In the same way, volumes are split into faces connected with blue 2-links (Fig. 2c). Then, faces are split into edges connected with red 1-links (Fig. 2d). Lastly, edges themselves are split into vertices with black 0-links (Fig. 2e) to produce the 3-G-map describing the objects shown in Fig. 2a. A G-map is therefore a graph, the nodes (named *darts*) are vertices of edges of faces of volumes and the arcs are i-links. By convention, border darts have 3 loops which are not represented to make the figures easier to read.

G-maps have conditions guaranteeing objects consistency, for example, two faces are always linked along an edge.

Topological cells are not explicitly represented in G-maps but only implicitly defined as subgraphs named *orbits*. They can be computed using graph traversals defined by an originating dart and by a given set of link labels. For example, the 0-cell (or the object's vertex) incident to some dart a (Fig. 3a) is the subgraph which contains a and all darts reachable from a, using links labelled by 1, 2 or 3 and the links themselves. This subgraph is denoted by $G\langle 1,2,3\rangle(a)$ where $\langle 1,2,3\rangle$ is the *type* of the orbit and models a vertex. The 1-cell (or edge) incident to a (Fig. 3b) is the subgraph $G\langle 0,2,3\rangle(a)$ which contains a and all the reachable darts using links labelled by 0, 2 or 3 and the corresponding links. The 2-cell (or face) incident to a (Fig. 3c) is the orbit $G\langle 0,1,3\rangle(a)$. The 3-cell (or volume) incident to a (Fig. 3d) is the orbit $G\langle 0,1,2\rangle(a)$. Note that orbits are more general than cells. For example, the volume edge $G\langle 0,2\rangle(a)$ (Fig. 3e) is the $\langle 0,2\rangle$-orbit incident to a.

2.2 Graph transformation rules

Jerboas's [Bel+14; Arn+22] graph transformation rules allow the formalization of operations over G-maps. In a few words, a *rule* $r : L \longrightarrow R$ and a *match* $m : L \to G$ to a G-map G, describe the transformation $G \longrightarrow^{r,m} H$ from G to H. The match m allows the replacement of a subgraph of G described by the left-hand side of the rule L with another one described by the right-hand side R, in order to produce H.

Informally, in the extrusion rule (Fig. 4), the left-hand side is made of only one node n_1 (orange) labelled with the $\langle 0,1\rangle$ face type: this way, it can match any face. For the match $m : n1 \mapsto 6$ from L to G (Fig. 5a), the node n_1 matches the whole face $G\langle 0,1\rangle(6)$. On the right side, the node n_1 label remains $\langle 0,1\rangle$. This means that, after applying the rule, the matched face $\langle 0,1\rangle(6)$ has been preserved, in other words $G\langle 0,1\rangle(6) = H\langle 0,1\rangle(6)$ (Fig. 5). In R, the new node n_2 (blue) creates, a copy of the matched face in H. However, n_2's label is $\langle 0,\_\rangle$ meaning that 0-links are preserved and 1-links are deleted. Therefore, n_2 creates face edges $\langle 0\rangle$ from

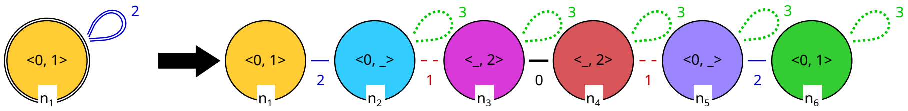
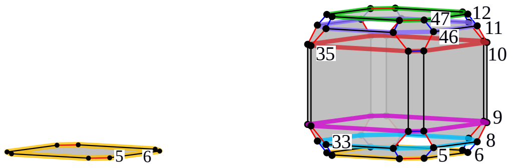


Figure 4: Rule extruding a face into a volume



(a) Graph G: matched by n_1 (b) Graph H : a cube

Figure 5: Extrusion of a face into a cube (rule Fig. 4)

the edges of the matched face. In a similar way, n_3 (pink) creates another copy of the matched face. Because it is labelled $\langle \_, 2\rangle$, 0-links are deleted, 1-links relabelled with 2, n_3 creates edge vertices $\langle 2\rangle$ from the matched face's vertices. Finally, the nodes n_4, n_5, n_6 create the same orbits than nodes n_3, n_2 and n_1, respectively. The nodes' labels, called *implicit arcs*, match the highlighted links (Fig. 5b).

The arc between n_1 and n_2, called *explicit arc*, is 2-labelled in the extrusion rule (Fig.4) and 2-links one-to-one the preserved orange darts and the created blue darts (Fig. 5). Similarly, the explicit arc between n_2 and n_3, 1-links one-to-one the blue and pink darts.

The node n_1 (Fig. 4) is a *preserved node* because it belongs to both the left and right-hand sides of the rule. Nodes n_2 to n_6 are *created nodes* because they belong only to the right-hand side. *Deleted nodes* belong only to the left-hand side. Note that the extrusion rule does not have any deleted node.

The *orbit* notion is extended to patterns of rules.

Jerboa's rules provide syntactic properties which guarantee the preservation of the consistency of G-maps [Arn+22].

2.3 Orbit tracking

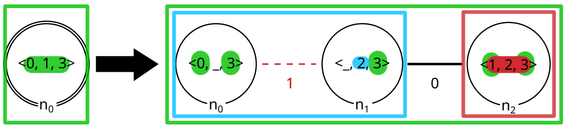
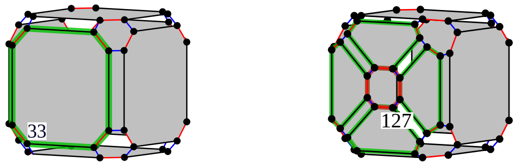


Figure 6: Rule triangulating a face



(a) Graph H: front face matched by n_0 (b) Graph I: triangulated front face

Figure 7: Triangulation of a face (rule Fig. 6)

Rules contain the necessary information characterizing the topological changes affecting an orbit throughout an application [Gai+23a]. Thus, the tracking of an orbit is automatically made without any addition other than the rules' syntactic analysis.

Consider the example of the triangulation (Fig. 6) and its application on H's dart 33 (Fig. 7a). The left-hand side of the rule matches the whole green square face $H\langle 0,1,3\rangle(33)$ while its right-hand side splits it into four green triangles in graph I (Fig. 7b).

Similarly to G-maps' orbits, in the left-hand side of the rule, the $\langle 0,1,3\rangle$-orbit incident to n_0 is the orbit containing the node reachable through arcs labelled in $\langle 0,1,3\rangle$ and those same arcs. This left-hand side orbit is written $L\langle 0,1,3\rangle(n_0)$. Therefore, $L\langle 0,1,3\rangle(n_0)$ is the orbit matching the green face.

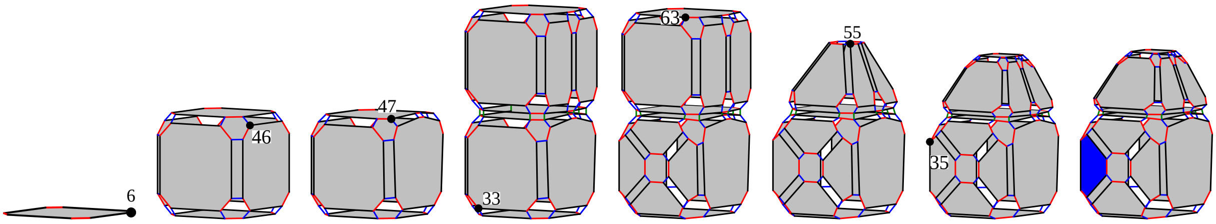
In green (Fig. 6), an 1-arc connects the nodes n_0 and n_1 and a 0-arc connects n_1 and n_2, thus forming the $\langle 0,1,3\rangle$-orbit (face) incident to n_0. This orbit matches the four faces resulting from the application of the rule (Fig. 7b). The syntactic analysis of the rule allows us to deduce that the face orbit is *split* along its implicit 1-arcs because the second implicit 1-arc of n_0 in the left-hand side is relabelled outside of the face $\langle 0,1,3\rangle$-orbit type in the right-hand side. Consequently, the matched face of graph H is split along its vertices' 1-links and into four faces in graph I. Similarly, a rule *merges* two or more $\langle o\rangle$-orbits when a k-th implicit arc is relabelled from i to j, where $i \notin \langle o\rangle$ and $j \in \langle o\rangle$, while there was no such k-th implicit arc in any node of the left-hand side.

In red, the vertex orbit $R\langle 1,2,3\rangle(n_2)$ incident to n_2, matches the vertex $I\langle 1,2,3\rangle(127)$. Since $R\langle 1,2,3\rangle(n_2)$ only contains n_2 which has been created, thus the orbit itself is *created* and the application of the rule creates the vertex $I\langle 1,2,3\rangle(127)$.

In blue, the face vertex $\langle 1,2\rangle$-orbit incident to the preserved node n_0 matches the face vertices of the green face such as the vertex orbit $H\langle 1,2\rangle(33)$. The node n_1 is added to the orbit through a 1-arc, $R\langle 1,2\rangle(n_0)$, thus modifying it. Consequently, the matched face's vertices of graph H are modified in graph I.

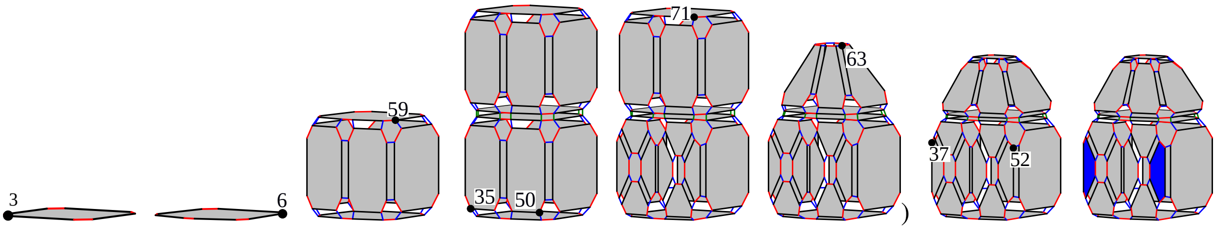
Finally, the face edge $\langle 0\rangle$-orbit incident to the preserved node n_0 matches the green face's edges in H. Since no node is added nor deleted from the orbit nor any arc is relabelled, thus the orbit is *not modified*.

Through this analysis, topological changes can be logged within bulletin boards which are automatically computed without requiring any other intervention.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 8: Evaluation (a) 1-square(pos); (b) 2-extrude(PN_1,vec); (c) 3-insert(PN_2); (d) 4-extrude(PN_3,vec); (e) 5-triangulate(PN_4); (f) 6-collapse(PN_5); (g) 7-chamfer(PN_6); (h) 8-colour(PN_7)



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 9: Reevaluation (a) 1-square(pos); (b) ADD1-insert(3); (c) 2-extrude(PN_1,vec); (d) DELETE 3-insert(PN_2); (e) 4-extrude(PN_3,vec); (f) 5-triangulate(PN_4); (g) 6-collapse(PN_5); (h) 7-chamfer(PN_6); (i) 8-colour(PN_7)

3 PARAMETRIC SPECIFICATION

During an object's construction, a *parametric specification* records both the rules representing the applied operations and their parameters (both topological and geometric) in order to describe the modeling process. Editing a parametric specification means that rules may be added, deleted, moved, and their parameters can be changed. With such changes, the topological parameters may have different concrete names, be deleted and so on. As a result, using the parameters' concrete names eventually lead to unexpected results (at best) or failure at runtime. To this end, persistent names are required to robustly identify topological parameters across an object reevaluation.

This section will follow the evaluation (Fig. 8) of a modeling process as an example in order to illustrate the creation of a parametric specification and its persistent names. Its reevaluation (Fig. 9) illustrates a variant modeling process where a rule is added between applications 1 and 2, and where the application 3 is deleted.

In this paper, all the figures were generated using Jerboa and the software overlay developed to implement the concepts presented in this section and the following ones. An overview of the results can be found here: `http://xlim-sic.labo.univ-poitiers.fr/jerboa/doc/model-reevaluation-based-on-graph-transformation-rules/`

3.1 Persistent name

Since rules use darts as topological parameters, it follows that each persistent name must represent a unique dart. As it happens, rules make it possible to determine unambiguously, which node filters or creates any dart.

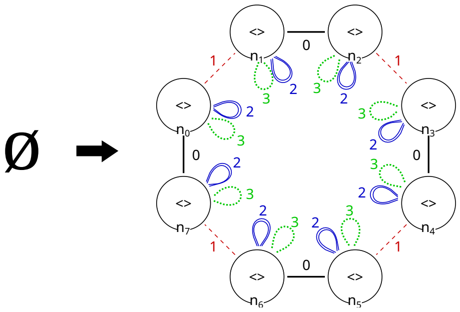


Figure 10: Rule creating a square face

For example, the square rule (Fig. 10) creates 8 darts *ex nihilo* (Fig. 8a), one per node. Thus, dart 3 is created during the first application, by the square's node n_3. The history of dart 3 is noted $[1n_3]$. Similarly, since dart 5 (resp.) 6 is created by node n_5 (resp. n_6), its history is $[1n_5]$ (resp. $[1n_6]$).

The second application (Fig. 8b and for more details Fig. 5) of the face extrusion rule (Fig. 4), creates darts from the matched darts. This rule is applied on the square face created during the first application. Since this face contains 8 darts, each of them is matched by node $n1$ of the extrusion rule (Fig 5). As a consequence, nodes n_2 to n_6 create darts copied from the 8 matched darts. Therefore, dart 6's history is now $[1n_6;2n_1]$, its copy 8's history is $[1n_6;2n_2]$, its copy 9's history is $[1n_6;2n_3]$, its copy 10's history is $[1n_6;2n_4]$ and so on. Similarly, histories of darts 46 and 47 are $[1n_5;2n_5]$ and $[1n_5;2n_6]$, respectively.

For the same reason, histories of darts 33 and 35 are $[1n_3;2n_2]$ and $[1n_3;2n_4]$, respectively. It follows that the cube's 48 darts all have a different history.

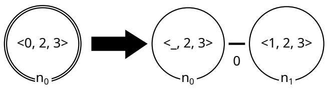


Figure 11: Rule inserting a vertex on an edge

The third application (Fig. 8c) uses the rule inserting a vertex on an edge. Both darts 46 and 47 are matched by node n_0 of the insertion rule. Their respective histories are $[1n_5;2n_5;3n_0]$ and $[1n_5;2n_6;3n_0]$ now. Conversely, darts 33 and 35 have not been not matched during the vertex insertion and their respective histories remain $[1n_3;2n_2]$ and $[1n_3;2n_4]$. The object's 52 darts, again, have each a different history.

In short, the history of any dart of the topological model is entirely defined by the rules applied during the evaluation process. This process guarantees to associate each dart with a unique history; hence, we use this history as the persistent name of the dart. Therefore, $PN_1 = [1n_6]$, $PN_2 = [1n_5;2n_5]$, $PN_3 = [1n_5;2n_6;3n_0]$ and so on.

3.2 Parametric Specification syntax

A number of fields are used to describe an application within a parametric specification, namely an application number, a rule application with the topological parameters and the geometric parameters.

Listing 1: Initial parametric specification

```
1-square(pos)
2-extrude(PN1=[1n6], vec)
3-insert(PN2=[1n5;2n5])
4-extrude(PN3=[1n5;2n6;3n0], vec)
5-triangulate(PN4=[1n3;2n2])
6-collapse(PN5=[1n4;2n6;4n6])
7-chamfer(PN6=[1n5;2n6;3n0;4n4;6n2])
8-colour(PN7=[1n3;2n4;5n0])
```

The parametric specification above represents the modeling process of an initial evaluation (Fig. 8) where each persistent name uses the history of the dart's number displayed in the previous construction step (dart 6 for PN_1, 46 for PN_2, 47 for PN_3, 33 for PN_4 and so on.).

Furthermore, a set of tags are used to describe an application whenever it is either added (`ADD`), deleted (`DELETE`) or moved (`MOVE`) at reevaluation. Thus, the initial parametric specification, once edited as shown in List 2 produces the reevaluation process shown in Fig. 9.

Listing 2: Edited parametric specification

```
1-square(pos)
ADD1-insert(3)
2-extrude(PN1=[1n6], vec)
DELETE 3-insert(PN2=[1n5;2n4])
...
8-colour(PN7=[1n3;2n4;5n0])
```

4 EVALUATION

Although a persistent name represents the history of a dart, an orbit is subject to topological changes and, thus, requires the construction of its own history in order to be accurately matched at reevaluation. Once the initial evaluation (Fig. 8) is done and its parametric specification (1) has been built, an evaluation's Directed Acyclic Graph (or DAG) must be issued for each persistent name before any parametric specification can be reevaluated. An evaluation DAG traces the history of each topological parameter back to the first created orbits it originates from, thus allowing the matching of the corresponding topological parameter at reevaluation time.

4.1 Evaluation DAG

An evaluation DAG is built parsing the applications and nodes of a persistent name from end to start. It is sorted by levels representing the different applications inside a history. Each level is made of an orbit level and an event level. The orbit level contains a node's name and some orbits. The event level contains a rule's application number and some events.

For example, let us consider $PN_3 = [1n_5;2n_6;3n_0]$ (Fig. 8d). PN_3 represents the topological parameter upon which the face extrusion rule is applied. This rule is filtered by hook n_1, matching the face $\langle 0,1\rangle(47)$ (Fig. 8c). The matched orbit's history is built from its dart's history (*i.e.* its persistent name). The DAG is built bottom-up by a backward traversal through the persistent name. Since PN_3 is made of 3 parts, its evaluation DAG contains 3 levels. The last part of PN_3 is $3n_0$, meaning that the dart of interest is filtered by the node n_0 of the third operation in the initial parametric specification. Therefore, the Orbit level 3 contains both n_0 and the matched orbit $\langle 0,1\rangle$. As shown in (List. 1), the third operation is the vertex insertion on an edge (Fig. 11). The right side of this rule has modified the orbit $\langle 0,1\rangle(n_0)$. We infer that the Event level 3 contains the third operation (`3-insert`) and the event's name `MODIFICATION`.

Continuing through the persistent name's backward traversal, the previous element `2n6` allows determining the DAG's second level in a similar way. Finally, `1n5` allows computing the first level. At last, the produced evaluation DAG (Fig 12) represents the volume face $\langle 0,1,3\rangle$-orbit's history resulting from applying `4-extrusion`. This DAG can be read top-down:

Level 1 The application `1-square` creates the volume face $\langle 0,1\rangle(n_5)$.

Level 2 From this volume face, `2-extrusion` creates the volume face $\langle 0,1\rangle(n_6)$.

Level 3 Finally, this latter volume face is modified by `3-insert`, inserting a vertex on its edge.

The application of the extrusion rule matches face $\langle 0,1\rangle(n_5)$. Then, the extrusion creates the face

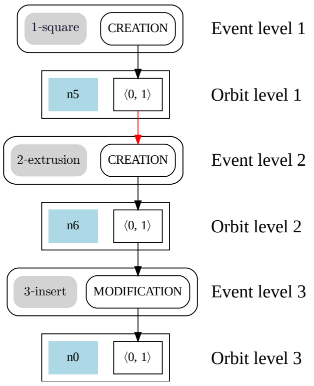


Figure 12: PN_3's evaluation DAG

$\langle 0,1\rangle(n_6)$ (Fig. 8b). Finally, applying the vertex insertion rule modifies the face $\langle 0,1\rangle(n_0)$ (Fig. 8c).

The syntactic analysis of the rules enables events to be computed only once. These events can be stored in cache to automatically build other evaluation DAGs.

4.2 Traces and origins

In order to accurately represent the history of an orbit, two types of arrows are used in an evaluation DAG (and later in the reevaluation DAG): black trace arrows and red origin arrows.

A *black trace arrow* allows orbit evolution tracing. For example, the triangulation rule splits an initial face into multiple subfaces. If one of these subfaces is referenced in the DAG, it is connected by a trace arrow to the initial face. Therefore, a trace arrow connects two orbits of the same dimension. A *red origin arrow* allows linking an orbit with the orbit that generated it, thereby connecting two orbits of potentially different dimensions. For example, continuing with the triangulation rule (Fig. 6), we observe that upon applying this rule, each edge of the initial face generates a different subface. If any of these subfaces is referenced in the DAG, it is then connected by an origin arrow to the edge that generated it. This is what can be observed considering again the example in Fig. 8 and more precisely the colour rule's PN_7 parameter. PN_7 represents dart 35's history and PN_7's evaluation DAG (Fig. 13) represents the history of the volume face that needs to be colored (the volume face adjacent to dart 35). This volume face $\langle 0,1\rangle(n_0)$ is the result of operation `5-triangulate` which splits the volume face $\langle 0,1\rangle(n_4)$ and has the face edge $\langle 0\rangle(n_4)$ as its origin (respectively represented by a black and red arrow between orbit level 2 and event level 3).

As explained in section 4.1, this DAG is built using a traversal of PN_7 and a bottom-up construction. This process is done in a similar way for both traces and origins, allowing for an efficient persistent naming mechanism that also takes into account the impact of origin modifications during reevaluation. To illustrate this, let us consider the previous face edge $\langle 0\rangle(n_4)$, which is the origin of the volume face $\langle 0,1\rangle(n_0)$ that needs to be colored. Suppose that, due to the addition of an operation in the edited specification, this origin may be split into two face edges. Upon applying `5-triangulate`, these two face edges will generate two volume faces, which can then be matched during reevaluation to the face to be colored, leveraging on the origin orbit recorded in the DAG.

In a formal way, an origin orbit can be automatically deduced through the syntactic analysis of a rule. More precisely, if n is a hook and n' is not a preserved node different from n, the origin of an orbit $R\langle o\rangle(n')$ is the suborbit $L\langle o'\rangle(n)$ consisting of the set of n's implicit arcs which are:

- rewritten on $R\langle o\rangle(n')$;
- not rewritten on $R\langle o\rangle(n')$ and belonging to $\langle o\rangle$.

For example, let us calculate the origin of a volume face ($\langle 0,1\rangle$-orbit) resulting from a split in the triangulation rule (Fig. 6). Only the implicit arc 0 in the $L\langle 0,1\rangle$-orbit incident to the hook n_0 is rewritten on $R\langle 0,1\rangle(n_0)$). Hence, the origin orbit of a volume face split by the triangulation rule is a face edge ($\langle 0\rangle$-orbit). When the orbit is either *split* or *merged*, the syntactic analysis allows the deduction of an *origin* in addition to a traced orbit. When the orbit is *created*, if the left-hand side of the rule is empty (meaning the orbit is created from scratch), there is neither trace nor origin (event level 1 in Fig. 13 where the 1-square rule creates the edge face $\langle 0\rangle(n_3)$ from scratch). Otherwise, there is no trace but an origin (event level 2 in Fig. 13 where the 2-extrusion rule creates the front face $\langle 0,1\rangle(n_4)$ of the cube from the origin previous face edge $\langle 0\rangle(n_3)$). When the orbit is just *modified* or *not modified*, there is only a traced orbit (event level 3 in Fig. 12 where the 3-insert rule modifies the top face $\langle 0,1\rangle(n_6)$ of the cube inserting a vertex on its boundary).

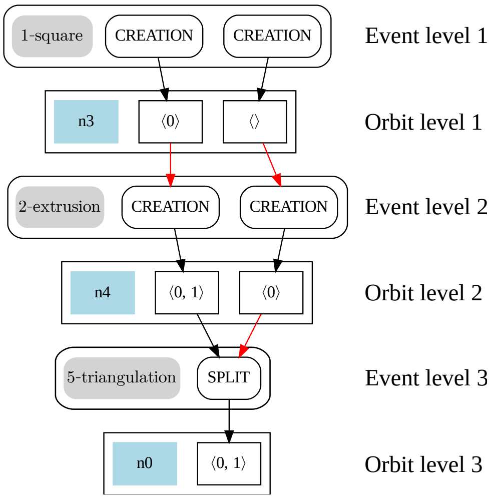


Figure 13: PN_7's evaluation DAG

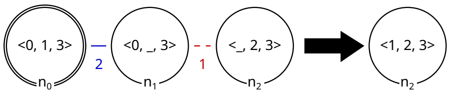


Figure 14: Rule collapsing a face into a vertex

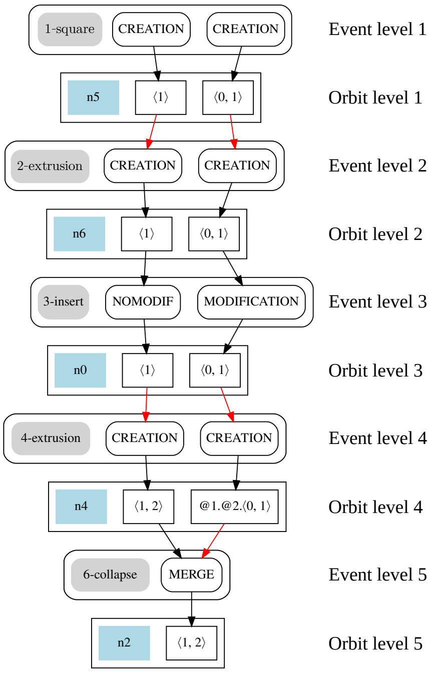


Figure 15: PN_6's evaluation DAG

4.3 Paths to origins

We have defined the origin of an orbit $R\langle o\rangle(n')$ when n' is not a preserved node different from the hook. In the opposite case, it is necessary to add to the origin the *path* that allows reaching the implicit arcs of the hook from n', because the implicit arcs of an origin are those of the hook, not those of n'.

For example, in the face collapse rule (Fig. 14), node n_2 is a preserved node different from the hook. Assume we want to define the origin of the volume vertex $\langle 1,2\rangle(n_2)$. A path represents the traversal in the right-hand side of the rule from the node n_2 to the hook node n_0. The traversed explicit arcs from n_2 to n_0 are, in the following order, 1 and 2 (written $@1.@2$ in the evaluation DAG). This can be seen in the PN_6's evaluation DAG (Fig. 15), where applying `6-collapse` generates a merge of vertices and the origin of the volume vertex $\langle 1,2\rangle(n_2)$ is the volume face $@1.@2.\langle 0,1\rangle(n_4)$. Actually, node n_4 of the extrusion rule (Fig. 4) used at the previous level of the DAG (level 4) matches dart 55 on the lateral faces of the cube (Fig. 5) because it is the dart matching the history stored in PN_6 DAG (`1-square` creates the initial bottom volume face $\langle 0,1\rangle(n_5)$. Then, `2-extrusion` applied on this bottom face creates the volume face $\langle 0,1\rangle(n_6)$ which is then modified by `3-insert` in $\langle 0,1\rangle(n_0)$. Finally, `4-extrusion` applied on this modified volume face creates node n_4, which matched dart 55). The volume face $@1.@2.\langle 0,1\rangle(n_4)$ reached starting from dart 55 and following links 1 and 2 is indeed the top face expected to be collapsed.

5 REEVALUATION

Each evaluation DAG represents an orbit's history which is valid with regards to the initial evaluation. When reevaluating, editing the parametric specification makes the topological parameters subject to changes. Thus, it is necessary to build reevaluation DAGs from the evaluation DAGs in order to update topological parameters. Once built, a reevaluation DAG can designate one, several, or no orbit depending on the editing of the parametric specification.

In this section, we keep using the previous example (Fig. 8 and 9) and its edited parametric specification (Lists 1 and 2), containing an added vertex insertion on the square's edge right after its creation and the deletion of the vertex insertion on the cube's edge.

5.1 Reevaluation DAG

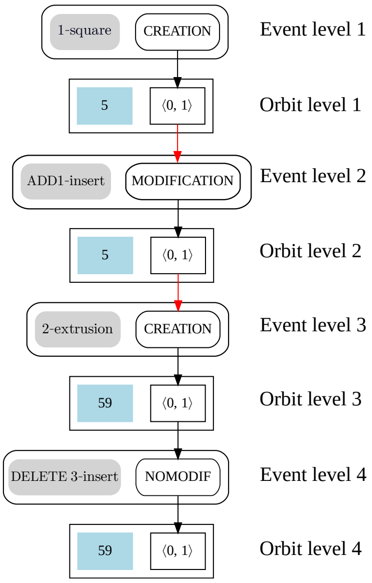


Figure 16: PN_3's reevaluation DAG

Contrary to the related evaluation DAG, a reevaluation DAG is built top-down throughout the reevaluation process. While an evaluation DAG represents the orbit's history of a topological parameter, the reevaluation DAG represents the history of this very same orbit after editing the parametric specification.

For example, let us consider PN_3 being the topological parameter of `4-extrusion`, which extrudes the cube's top face to produce a second cube. The reevaluation process builds PN_3's reevaluation DAG (Fig. 16 step-by-step from its evaluation DAG (Fig. 12):

Level 1 The application `1-square` has no topological parameter and, thus, is identically reevaluated. Once again, its node $n5$ creates a single dart 5 and the volume face $\langle 0,1\rangle(5)$ is identically reevaluated. This is why the orbit level 1 of the reevaluation DAG references dart 5.

Level 2 The second application is an added one. `ADD1-insert` does match an edge of the tracked volume face $\langle 0,1\rangle(5)$ and modifies it, as deduced from the rule. Therefore, the event level 2 contains `MODIFICATION` and the orbit level 2 contains the same dart 5 and orbit $\langle 0,1\rangle$.

Level 3 The third application is the extrusion of the square face into a cube. During its application, dart 5 is matched by the hook n_1 (Fig. 4). From the evaluation DAG, the tracked dart is the copy of dart 5 created by node n_6. Applying the rule allows finding out this dart, numbered 59.

Level 4 Finally, the last application of PN_3 is deleted. Consequently, the modification that occurred during the initial evaluation does not occur at reevaluation. Therefore, the event level 4 contains `NOMODIF` and the orbit level 4 contains the same dart 59.

The reevaluation DAG identifies the concrete name using its persistent name. PN_3's concrete name is 59 (cf. Fig. 9c and 9e). We now study a more complex example with the reevaluation DAG of PN_6 (Fig. 17),which represents the pyramid's top vertex (Fig. 8f):

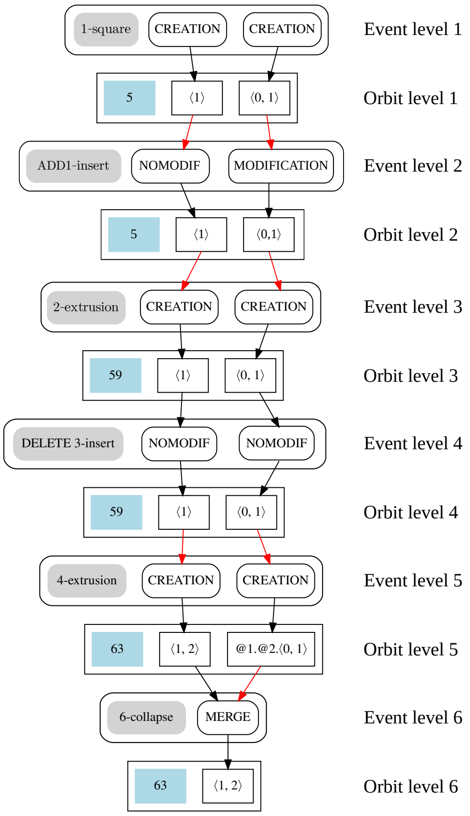


Figure 17: PN_6 reevaluation DAG

Level 1 to 4 These three levels are similar to PN_3's reevaluation DAG (Fig. 16), with the tracking of the volume face vertex $\langle 1\rangle$-orbit in addition to the $\langle 0,1\rangle$-orbit one. The $\langle 1\rangle(5)$-orbit is created by the `1-square` application. `ADD1-insert` matches and preserves one dart of this orbit with the node n_0 of the insertion rule (Fig. 11). Thus, the event level 2 contains `NOMODIF`. Then, this orbit is copied to create a new volume face vertex by `2-extrusion`. Again, `DELETE 3-insert` does not modify the orbits.

Level 5 The fifth application `4-extrusion` of the extrusion rule (Fig. 4) matches the volume face $\langle 0,1\rangle$ with its hook $n1$ and creates a copy of dart 59 dart 63. As in the initial evaluation, it creates a volume vertex and a face. Then, the event level contains two `CREATION`. The orbit level references dart 63 and contains both orbits $\langle 1,2\rangle$ and $@1.@2.\langle 0,1\rangle$.

Level 6 Finally, while the tracking of `5-collapse` shows that the application keeps merging the volume vertices incident to the matched face, it preserves dart 63 which is matched by node $n2$. Therefore, the event level contains one `MERGE` and the orbit level references dart 63 and contains the volume vertex $\langle 1,2\rangle$.

At last, PN_6's concrete name is dart 63 (cf Fig. 9g).

These two examples here are quite straightforward as there was only one possible candidate dart each time. However, in some complex specifications, there can be more than a single dart to choose between.

5.2 Parameter matching strategies

The editing of the parametric specification leads to having a different DAG at reevaluation (with event levels and/or branches being added, deleted or both). For example, an orbit split present in the evaluation DAG may disappear during the reevaluation, a merging can be added and so on. Several matching strategies can then be considered depending on the application's context. This can be illustrated with PN_7's example which designates the face that must be coloured (Fig. 8h). The addition of `ADD1-insert` application at reevaluation splits the origin of designated face, resulting in the addition of a branch in the reevaluation DAG. Let's work through PN_7's reevaluation DAG shown in Fig. 18:

Level 1 As seen previously, `1-square` creates the tracked orbits traced in the evaluation DAG.

Level 2 `ADD1-insert` matches dart 3. Since the vertex insertion rule (Fig. 11) splits the volume face edge $\langle 0\rangle(3)$, its history is also split and there are two concrete names to consider. It follows that the event level contains two `SPLIT`, one for each volume face edge suborbit, and two `NOMODIF`, one for each dart 3 and 4.

Level 3 `2-extrusion` extrudes the square face into a cube. The extrusion rule (Fig. 4) matches dart 3 with its hook n_1. It extrudes the face edge $\langle 0\rangle(3)$ into a volume face, the dart $\langle\rangle(3)$ into a face edge and its node n_2 creates the dart 37 as a copy of dart 3. The same goes

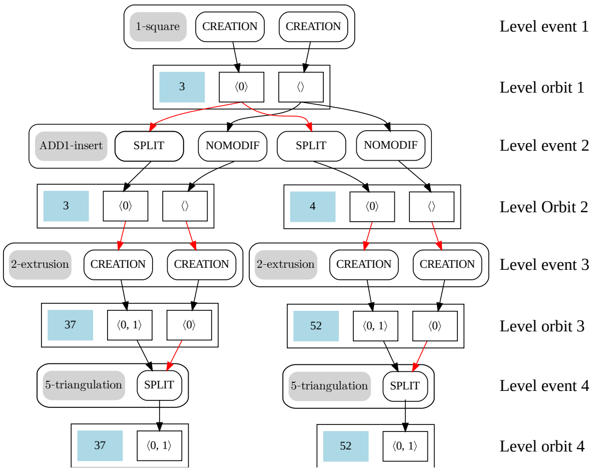


Figure 18: PN_7 reevaluation DAG

for the history on the right. The rule matches dart 4, it creates the same two orbits and a dart 52 as a copy of dart 4. both event levels contain two `CREATION`. Both orbit levels contain orbits $\langle 0,1\rangle$ and $\langle 0\rangle$.

Level 4 `5-triangulation` triangulates the faces designated by the level above. Both darts 37 and 52 are matched and preserved by the rule.

Upon reevaluation, PN_7's DAG matches to different darts. An option would be to colour only one face (either $\langle 0,1\rangle(37)$ or $\langle 0,1\rangle(52)$). Another option would be to apply `8-colour` two times, one for dart 37 and one for dart 52 (as shown in Fig. 9i which represent our default strategy). In case the reevaluation DAG has two or more leaves, it shows all the possible entities that can be matched for a specific persistent name and to set a strategy up. Either way, such a strategy allows users to have a choice and best fit their modeling intents, depending on the application's context.

6 CONCLUSION

In this paper, we widen the naming problem studies to the rule-based graph transformation modeling systems. We take advantage of the formalism of both generalized maps and graph transformation rules to tackle the reevaluation mechanism task. Generalized maps offer an homogeneous representation of an object in all dimensions while Jerboa's rules define geometric modeling operations on which it is actually possible to perform syntactical analysis. We implement: a persistent name scheme where each persistent name represents a unique dart's history through the successive applications of rules and their matching nodes. Then, for each persistent name, an evaluation DAG is built in order to trace an orbit's history from the bottom and up to the orbits it originates from. To our knowledge, unlike other methods, our solution tracks only the entities used in the parametric specification and the ones they originate from. Representing the complete history of an orbit in an evaluation DAG allows for an efficient persistent naming mechanism that takes into account the impact of both origins and traces modifications during reevaluation. Finally, reevaluation DAGs are built from a top-down traversal of evaluation DAGs and allow matching each topological parameter on one or more, sometimes none, values depending on the editing of the parametric specification. Thanks to our method, not only the naming problem is tackled within the usual framework of parameters edition, but we also take the specification edition into account (operation addition, deletion or move). Moreover, this approach provides all the possible updated values of the parameters and, thus, enables implementing different strategies.

More complex operations can make use of several rules brought together in a script. Later works will revolve around widening this reevaluation mechanism to scripts.

REFERENCES

[ALS15] K. Arroyo Ohori, H. Ledoux, and J. Stoter. "A dimension-independent extrusion algorithm using generalised maps". In: *International Journal of GIS* 29.7 (2015), pp. 1166–1186.

[Arn+22] A. Arnould et al. "Preserving consistency in geometric modeling with graph transformations". In: *Mathematical Structures in Computer Science* 32.3 (2022), pp. 300–347.

[Bab10] M. Baba-Ali. "Systeme de nomination hierarchique pour les systemes parametriques". PhD thesis. 2010. URL: `http://theses.univ-poitiers.fr/notice/view/5362`.

[Bei+10] J.N. Beirao et al. "Implementing a Generative Urban Design Model: Grammar-based design patterns for urban design". In: *eCAADe*. 2010, pp. 265–274.

[Bel+14] H. Belhaouari et al. "Jerboa: A graph transformation library for topology-based geometric modeling". In: *International Conference on Graph Transformation*. Springer. 2014, pp. 269–284.

[Ben+17] F. Ben Salah et al. "A general physical-topological framework using rule-based language for physical simulation". In: *12th International Conference on Computer Graphics Theory and Application (VISIGRAPP/GRAPP)*. 2017.

[BTG15] E. Bohl, O. Terraz, and D. Ghazanfarpour. “Modeling fruits and their internal structure using parametric 3Gmap L-systems”. In: *The Visual Computer* 31 (2015), pp. 819–829.

[Car+19] A. Cardot et al. “Persistent naming based on graph transformation rules to reevaluate parametric specification”. In: *CADA* 16.5 (2019), pp. 985–1002.

[CBS23] D. Cascaval, R. Bodik, and A. Schulz. “A Lineage-Based Referencing DSL for Computer-Aided Design”. In: *Proceedings of the ACM on Programming Languages* 7 (2023), pp. 76–99.

[CH95] X. Chen and C. M. Hoffmann. “On editability of feature-based design”. In: *CAD* 27.12 (1995), pp. 905–914.

[DL14] G. Damiand and P. Lienhardt. *Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing*. A K Peters/CRC Press, Sept. 2014.

[DZ24] F. Dai and W. Zhao. “A Persistent Naming Discrimination Method Based on the Sweeping Direction”. In: *2024 4th International Conference on Consumer Electronics and Computer Engineering (ICCECE)*. IEEE. 2024, pp. 75–83.

[ESR] ESRI. *ArcGIS CityEngine product page*. `https://www.esri.com/en-us/arcgis/products/arcgis-cityengine/overview`. Accessed 2023-05-09.

[FH18] S. H. Farjana and S. Han. “Mechanisms of persistent identification of topological entities in CAD systems: A review”. In: *Alexandria engineering journal* 57.4 (2018), pp. 2837–2849.

[Gai+23a] M Gaide et al. “Automatic Detection of Topological Changes in Geometric Modeling Operations”. In: *Computer Graphics and Visual Computing*. 2023, pp. 9–18.

[Gai+23b] M. Gaide et al. “Model Reevaluation Based on Graph Transformation Rules”. In: *Computer Graphics and Visual Computing*. 2023, pp. 61–63.

[HH00] Y. Halbwachs and Ø. Hjelle. “Generalized maps in geological modeling: Object-oriented design of topological kernels”. In: *Advances in Software Tools for Scientific Computing*. Springer. 2000, pp. 339–356.

[HMV09] S. Haegler, P. Müller, and L. Van Gool. “Procedural modeling for digital cultural heritage”. In: *EURASIP Journal on Image and Video Processing* (2009).

[Hor+09] S. Horna et al. “Consistency constraints and 3D building reconstruction”. In: *CAD* 41.1 (2009), pp. 13–27.

[Lie91] P. Lienhardt. “Topological models for boundary representation: a comparison with n-dimensional generalized maps”. In: *CAD* 23.1 (1991), pp. 59–82.

[Lin74] A. Lindenmayer. “Adding continuous components to L-systems”. In: *L systems* (1974), pp. 53–68.

[Mar06] D. Marcheix. “A persistent naming of shells”. In: *International Journal of CAD/CAM* 6.1 (2006), pp. 125–137.

[MH05] D. Mun and S. Han. “Identification of topological entities and naming mapping for parametric cad model exchanges”. In: *International Journal of CAD/CAM* 5.1 (2005), pp. 69–81.

[Mül+06] P. Müller et al. “Procedural modeling of buildings”. In: *ACM SIGGRAPH Papers*. 2006, pp. 614–623.

[QB15] R. Quattrini and E. Baleani. “Theoretical background and historical analysis for 3D reconstruction model. Villa Thiene at Cicogna”. In: *Journal of Cultural Heritage* 16.1 (Jan. 2015), pp. 119–125.

[Saf+20] M. Safdar et al. “Feature-based translation of CAD models with macro-parametric approach: issues of feature mapping, persistent naming, and constraint translation”. In: *Journal of Computational Design and Engineering* 7.5 (2020), pp. 603–614.

[Ter+09] O. Terraz et al. “3Gmap L-systems: an application to the modelling of wood”. In: *The Visual Computer* 25 (2009), pp. 165–180.

[Wu+01] J. Wu et al. “A face based mechanism for naming, recording and retrieving topological entities”. In: *CAD* 33.10 (2001), pp. 687–698.

[Xue+16] G. Xue-Yao et al. “Name and Maintain Topological Faces in Rotating and Scanning Features”. In: *International Journal of Grid and Distributed Computing* 9 (Mar. 2016), pp. 21–26.

PLOD: Point cloud level of detail for polygon mesh

Eric Konks
Phystech School of Applied Mathematics and Computer Science
Moscow Institute of Physics and Technology
Institutskiy Pereulok, 9
Dolgoprudny, Moscow Oblast, 141701, Russia
eric.konks@gmail.com

Alexandr Shcherbakov
Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Moscow, 119991, Russia
alex.shcherbakov.as@gmail.com

Abstract

Rendering high-polygonal models from distant perspectives has certain performance issues related to high density of subpixel triangles, which can be solved by levels of detail, a classical optimization method. Since a mesh occupies a small area on the screen, an alternative representation of the geometry in the form of a point cloud can replace the original geometry with little or no change to the image, and allowing for significant performance improvements due to the smaller number of primitives rendered. This paper presents a method for automatically generating a point cloud from a polygonal mesh with nested levels of detail. It also considers a method for rendering cloud with dynamically varying cloud density in real time based on view frustum and distance.

Keywords

point cloud, level of detail, impostor, point based rendering, optimization

1 INTRODUCTION

Modern 3D models in real-time computer graphics applications can be extremely detailed, with millions of polygons and high-resolution textures. However, models only occupy a certain area on the screen, which may not correspond to excessive geometry detail, thereby causing performance degradation without any improvement in image quality.

Various rendering technologies may be used to present a complex and heavy 3D scenes [DGY07]. The main methods are primitive rasterization and ray tracing, which have proven to be simple, robust and widely used. Geometry simplification and visibility culling are also common techniques to achieve interactive frame rates for a rendering application.

One of the most common optimization methods are levels of detail [Kru+97]. The basic idea is to create simplified versions of the original model that can replace each other, while preserving or slightly changing the original image. Simplification can occur in a number of ways. First of all by reducing the number of rendered primitives, which naturally affects performance and the amount of memory required. Replacing subpixel triangles with larger triangles increases the coverage of the fragment shader due to a greater ratio between active and helper invocations. Simplification can also affect the materials and lighting shaders used, speeding up the rendering process. It is clear that if a model is simplified significantly, visual quality tends to deteriorate. Therefore, in order for the degradation to have little effect on the image, the simplified model should replace the original model at a certain moment, when the difference between the details is not noticeable on the corresponding area of the screen. In other words, some dynamic real-time switching between levels of detail is required depending on the change of the distance to the camera and consequently the projected screen space size.

Manual adjustment of LOD (level of detail) is hampered by the need to balance performance and quality. Therefore, various optimization techniques and generation methods are designed to speed up the process of creating new assets and models, allowing the author to focus on direct visual representation.

2 RELATED WORKS

The most obvious way to create a level of detail is to simplify the geometry by reducing the number of poly-

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.

gons, what can be done automatically using various algorithms.

There are various ways to algorithmically generate simplified polygonal models [GGK02]. For example, clustering-based algorithms that use different criteria depending on the required topology accuracy. Simplification is performed by partitioning vertices into clusters followed by iterative edge collapsing.

Another method is remeshing [Kha+22], where instead of using existing polygons, new vertices are sampled on the mesh surface and a completely different mesh is created by triangulation. This approach produces a fairly uniform distribution of triangles of approximately the same size, but topology details may be compromised due to the discreteness of the sampling. Therefore, it is necessary to adjust the parameters and use sufficient depth and resolution.

LOD generation by mesh resampling can be combined with vertex data compression and streaming to reduce memory consumption [NFS22]. After resampling the mesh, new vertices are merged into patches and the data is quantized. Later, during rendering, the data is decompressed on GPU at the hardware tessellation stage. For mesh parts close the camera, patches have higher tessellation factor and it's decompression version produces more triangles. Thus, the method reduces both memory consumption and rendering time.

Completely different way is to use some alternative, image-based representation, an imposter, instead of the original polygonal representation. The term imposter refers to some object that replaces the original geometry and is rendered faster with little noticeable visual difference [Déc+03]. The method uses very simple geometric objects that can be rendered with a pre-generated image facing the camera. The disadvantages of this approach include the dependence of the image on the angle at which the object is observed. To address this problem, atlases are used in which the impostor consists of multiple images, each corresponding to a specific viewing angle [MFL21]. However, if there is insufficient variation, intermediate angles can provide a visually poor substitute for the original geometry. In addition, given that modern lighting models require material data, the memory consumption of impostors increases significantly.

One drawback of the approaches described above is that there is no unambiguous criterion by which the transition to the simplified model should be made. It is unclear at what point this transition can be made imperceptibly and when subpixel detail does not affect the image. To deal with this problem, a triangular mesh can be broken down into small clusters of vertices – meshlets [Bad+20]. Each meshlet is a set of connected vertices that represents some part of the geometry. Using mesh shaders, we can perform occlusion and LOD selection based on these individual meshlets, thereby maintaining a constant number of polygons on the screen. For example, Nanite technology in the Unreal Engine 5 heavily relies on the cluster mesh representation [Kar22]. This technology consists of automatic generation of a cluster representation of the mesh with levels of detail, a cluster streaming system, software rasterization and dynamic switching. In addition, it uses impostor atlas for the most distant LOD.

Triangular polygons are not the only way to represent geometry. For example, points can also be used as a drawing primitive [LW00]. They define the position of some point in the space in which the object exists, but, unlike polygons, they do not uniquely define a surface. Therefore, points must have sufficient density to visually simulate the continuity of a surface. In addition to position, a point can have other characteristics such as normal, tangent, color, metalness and other material properties.

The idea of using points as impostors was first introduced by Wimmer et al. [WWS01]. Since the screen resolution is limited, with enough point primitives it is possible to cover the required number of pixels on the screen without visible holes. This paper discusses a method for image-based generating of point cloud using multiple virtual cameras that perform ray tracing to produce object samples. Each point is used to sample a specially filled texture with the point radiance contribution for the current view. This point cloud is then rendered using the hardware drawing capabilities of point primitives.

Point primitives in graphics pipeline may be used to render raw point cloud [Wim15]. The method uses glPointSize for high-quality splatting to get good visual quality and appropriate surface representation. However some graphics API do not fully support point size, leaving us to use only 1 pixel points. Also screen space filters may be used [PGA11] to remove artifacts due to low cloud density. But these additional computations reduce the performance, which is very important for our usage case.

Our work further develops point cloud representation of polygon mesh using modern generation method and rendering techniques.

3 PROPOSED SOLUTION

Cloud generation

To create an alternative representation of geometry in the form of a point cloud, it is necessary to sample points on the surface of models in a certain way. The sampling should provide visual correspondence of the geometry to the original polygonal mesh from any possible angles, as well as have a number of properties that allow to maximize the quality of surface sampling. For this task, the distribution based on blue noise is suitable, the frequency parameters of which allow to reduce the number of ‘holes‘ or even totally eliminate them. Uniform distribution is extremely important because it allows to create the illusion of surface continuity without using real-time smoothing or restoring filters. Uniform distribution also has a positive impact on performance, as fewer points fall within a single pixel, reducing overdraw. To obtain such a distribution, we modernize the sampling algorithm using the Poisson sampling algorithm presented in [CCS12].

The cloud generation scheme consists of generating a pool of points and Poisson disk filtering. The first step sets the initial cloud surface density D_0, on which the minimum sampling distance will depend, as well as a density reduction step to specify nested levels of detail. This results in some set of densities D_i and $\forall i < j \Rightarrow D_i > D_j$.

Then we are generating a pool of points by sampling points on the surface of polygons with the largest density. The *pool* may be represented by some spatial hash container. The algorithm listing 1.

Listing 1: Point sampling

```
vector<float> densities = {D0, ... };
for (const auto &p : polygons)
{
  float area = getPolygonArea(p);
  int count = (size_t)area * D0;
  count = max(count, 1);
  while (count--)
  {
    auto sample = polygonSample(p);
    pool.add(sample);
  }
}
return pool;
```

For each density, starting from the lowest density, a Poisson disk is sampled from the *pool* of pre-generated points on the surface of the mesh. Moreover, the set of points S_i corresponding to the smaller density is a subset of the larger density $S_i \subset S_{i-1}$. The algorithm listing 2.

Listing 2: Cloud generation

```
int N = densities.size() - 1;
for (int lvl = N; lvl >= 0; --lvl)
{
  float density = densities[lvl];
  float r = poissonRadius(density);
  // Remove samples from next lods
  // as part of the current lod
  for (int i = lvl + 1; i <= N; ++i)
    for (sample : samples[i])
      pool.removeSamples(sample, r);
  // Then sample what is left
  while (!pool.isEmpty())
  {
    auto sample = pool.pop();
    samples[lvl].push_back(sample);
    pool.removeSamples(sample, r);
  }
  pool.restore();
}
return samples;
```

After sampling, we have a list of sets *samples*, where $samples_i$ corresponds to $S_i \setminus S_{i-1}$. We can combine all sets into a single list, where first n_i points of set S_0 correspond to a cloud of set S_i with surface density D_i.

Since the points are sampled on the surface of polygons, all attributes of the corresponding vertices are interpolated. Each point in the cloud has a position, normal and texture coordinates. So by rendering this cloud using S_0 as a vertex buffer, we can sample all the necessary textures and bake the material into points, which will save us from having to use textures later. However, this approach has a couple of difficulties. First, it requires no randomization of texture coordinates, which can occur in non-trivial shaders. Second, a point itself does not provide any derivatives used to compute the mip level of the material texture. This problem can be solved by knowing in advance at what distance the cloud will be further rendered or by calculation based on Poisson radius. The mip level calculation has not been considered in detail in this paper. Since the cloud is designed to be rendered at long distances, the use of some constant mip level is appropriate in this case.

Dynamic cloud density

As a result of generation, we have a point cloud with all necessary data for drawing on the screen (Fig 1).

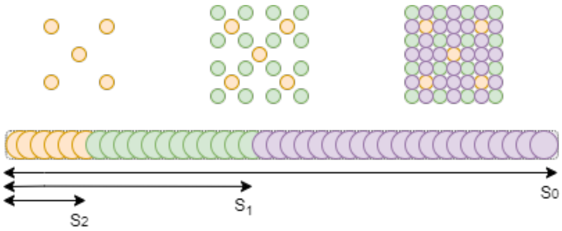


Figure 1: Point cloud list with nested densities

The point cloud with the structure of nested densities allows us to dynamically vary the number of drawn primitives. To do this, we need to find such a density of the cloud that, when projected on the screen, the density is sufficient to obtain a complete image without hole artifacts. Knowing the camera parameters and object position in view space we can estimate this value individually for each instance of the object.

Let L be the distance to the cloud, z_{near} be the near plane of the camera frustum, $(target_x, target_y)$ be the render resolution of target texture. Then we can find the surface density D_{target} that the cloud must have at distance L and making an angle α with the camera forward vector to completely cover the given area on the screen without noticeable holes.

$$D_{target} \approx \frac{target_x target_y}{zNearArea} \frac{z_{near}^2}{L^2 cos^2 \alpha} \qquad (1)$$

As a result, it is sufficient to draw only the first n_i points corresponding to the smallest upper bound of the possible density D_i, where $D_i >= D_{target}$ and $D_{i-1} < D_{target}$. The calculation of the number of primitives for vertex input can be performed on the GPU using compute or mesh shaders, which may give even more performance benefits to the application.

Rendering

Points can be rendered in two ways: using the compute or graphics pipeline. The compute method based on 64 bit atomic operation is well described in [SKW22] and achieves better performance in comparison to the hardware method. However the method has special render architecture and relies on 64 bit unsigned textures to perform depth testing and visibility calculation. Since our approach is a hybrid approach in which polygonal meshes are used equally with point clouds, we choose the graphics pipeline as it simplifies the implementation of the method in classical rendering.

We will look at several optimizations that noticeably improve performance when rendering point primitives.

Normal culling. Since unlike polygons points have no face, we need somehow determine orientation to cull out points, that can't be visible. In this case, point normal can be used for culling calculation. So culling can be done by the hardware in a a view port clipping by setting the position of a point in the vertex shader outside the boundary of the clip space. The condition is that the point-to-camera vector and the normal vector must point in different direction. This reduces the number of fragment shader invocations and allows us for significant performance improvements in case of not too dense clouds.

Software vertex fetching. To perform transformation and normal culling, it is not necessary to have information about the material, which is always fetched by the input assembly. To get only the necessary data, we obtain the data by sampling vertex buffer directly with an empty vertex shader input. This optimization saves us memory bandwidth and reduces the total number of read operations, since the vertex shader gets only the position and normals. Subsequently, only for points that have not been culled, the fragment shader fetch information about the shading material.

Conservative rasterization. Some graphics API allow conservative rasterization for point primitives, which is very useful for our method, since it increases point screen coverage. In this mode, a single primitive can produce up to 4 fragments. This means that visually the surface density increases by the same number of times. That is, we can render significantly fewer primitives while relatively preserving the same image.

4 EXPERIMENTAL RESULTS

As a demonstration of the method, a cloud was generated for a high-polygonal geometrically small mesh. Performance measurements was performed on the RTX 3060 at 1920x1080 screen resolution and using the Vulkan API. Additionally, the renderer uses temporal anti-aliasing, which works great for point primitives.

The density steps for the point cloud are degrees of two. Along with performance comparison, we compared images using mean square error (MSE) and structural similarity index measure (SSIM).

A highly polygonal mesh consists of 60k triangles, while the cloud at maximum density contains ten times fewer primitives, which is already sufficient for mid-range display with comparable visual quality (Fig. 2). Point cloud has significantly better performance compared to the high-polygonal mesh, and render time tends to decrease gradually (Fig. 3). The tendency has clear steps where cloud density switches to a lower level of detail (60m, 80m, 120m, 160m).

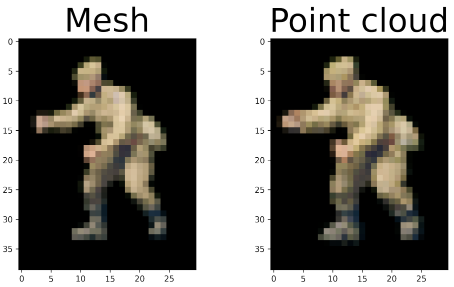


Figure 2: High polygonal mesh. Image comparison at 50 meters

MSE and SSIM metrics has pretty low variation and confirm the similarity of the rendered images (Fig. 4 Fig. 5).

A low-polygon mesh contains 1k polygons, and the cloud for it contains 41k points (Fig. 6). Since the triangles are quite large, the point cloud loses at short and medium ranges in terms of performance (Fig. 7), and only at large distances (250m) does the cloud gain an advantage. This result is not surprising since the cloud has much more compute pressure on the vertex processing stage. However, hardware conservative rasterization can significantly reduce the number of vertex shader calls, allowing the cloud to have fewer rendered primitives, but at the cost of losing visual detail.

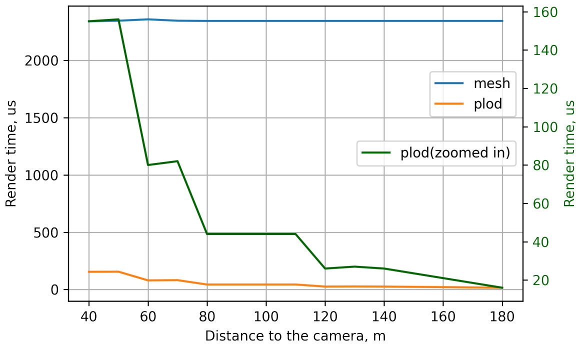


Figure 3: High polygonal mesh. Performance comparison

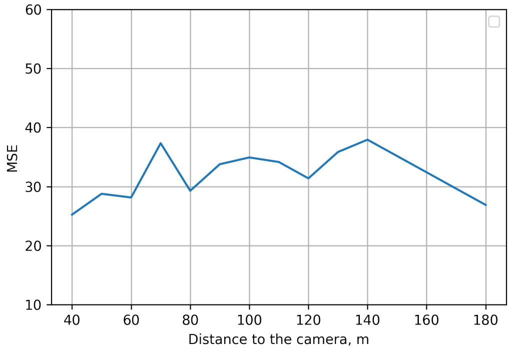


Figure 4: High polygonal mesh. MSE

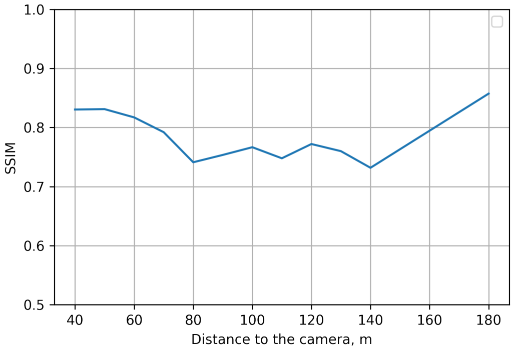


Figure 5: High polygonal mesh. SSIM

MSE and SSIM metrics for point cloud rendered with conservative rasterization shows less image similarity, but the overall scores trend is identical (Fig. 8 Fig. 9).

In the presence of multiple rendering optimizations, the bottleneck of the current rendering method is the rasterization stage, which limits the SM occupancy of the vertex stage. Comparing the images using MSE and SSIM scores, it can be seen that, the point cloud image tends to degrade due to the rasterization of points at the edge of the cloud, which slightly expands the silhouette of the model. Full distance range image comparisons for high-polygonal mesh (Fig. 10a) and for a low-polygonal large mesh (Fig. 10b).

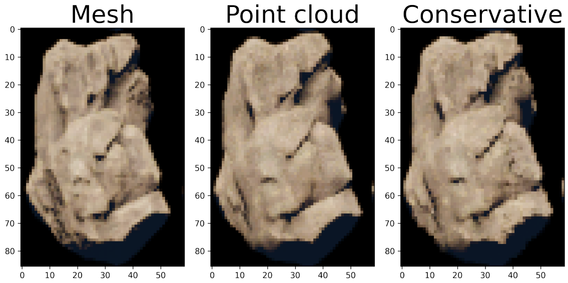


Figure 6: Low polygonal mesh. Image comparison at 50 meters of mesh, point cloud and conservative point rasterization

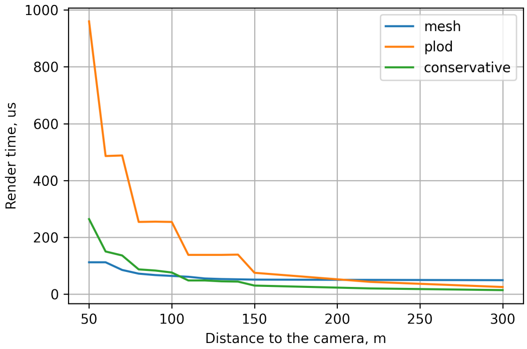


Figure 7: Low polygonal mesh. Performance comparison

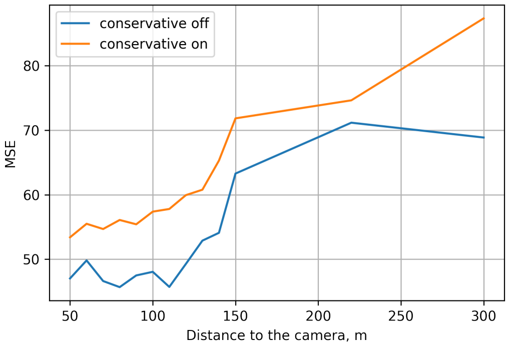


Figure 8: Low polygonal mesh. MSE

5 CONCLUSION

We presented a method for rendering and generating a special type of level of detail that significantly optimizes the long-range rendering of high-polygonal models with non-critical image changes. Due to dynamic density, each mesh has a distance at which point cloud may replace the image with performance gains. The point cloud level of detail shows significant performance advantages for highly polygonal models even at medium distances. In this method, low-polygon meshes can be replaced at the large distances at which imposters are typically used. Therefore, the point cloud can be considered as 3d impostors that can incorporate mesh detailing and impostor performance.

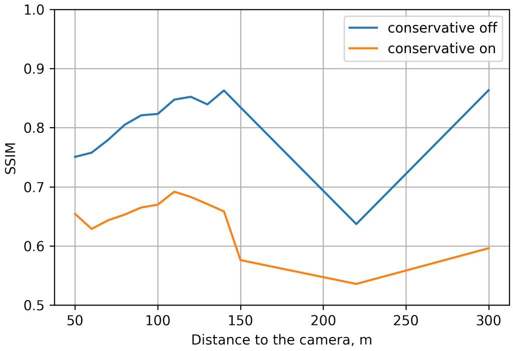


Figure 9: Low polygonal mesh. SSIM

The disadvantages of point clouds are the small image changes that can be noticeable, especially with conservative rasterization, due to the difference between point and triangle rasterization and the difference in mip levels between mesh rendering and point sampling.

Another drawback is the dependence on screen resolution, so we need to draw more points to match the screen space density, losing performance due to redundant vertex invocations. However compute rasterization and point size in some API are able to mitigate this limitation, by increasing the screen primitive size.

Moreover, by increasing the size of primitives and decreasing their number, we can adjust the ratio between image quality and performance. Finding the optimal ratio between quantity and size is a task for future research.

REFERENCES

[Bad+20] Abhishek Badki et al. “Meshlet Priors for 3D Mesh Reconstruction”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2846–2855.

[CCS12] Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno. “Efficient and Flexible Sampling with Blue Noise Properties of Triangular Meshes”. In: *IEEE transactions on visualization and computer graphics* 18 (Jan. 2012), pp. 914–24.

[Déc+03] Xavier Décoret et al. “Billboard clouds for extreme model simplification”. In: *ACM Trans. Graph.* 22.3 (July 2003), pp. 689–696.

[DGY07] Andreas Dietrich, Enrico Gobbetti, and Sung-Eui Yoon. “Massive-Model Rendering Techniques: A Tutorial”. In: *IEEE Computer Graphics and Applications* 27.6 (2007), pp. 20–34.

[GGK02] Craig Gotsman, Stefan Gumhold, and Leif Kobbelt. “Simplification and Compression of 3D Meshes”. In: *Tutorials on Multiresolution in Geometric Modelling* (Feb. 2002).

[Kar22] Brian Karis. “Journey to Nanite”. en. In: *High Performance Graphics* (2022).

[Kha+22] Dawar Khan et al. “Surface Remeshing: A Systematic Literature Review of Methods and Research Directions”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.3 (2022), pp. 1680–1713.

[Kru+97] Mike Krus et al. “Levels of detail & polygonal simplification”. In: *XRDS* 3 (1997), pp. 13–19.

[LW00] Marc Levoy and Turner Whitted. “The Use of Points as a Display Primitive”. In: 2000.

[MFL21] Martin Misiak, Arnulph Fuhrmann, and Marc Erich Latoschik. “Impostor-based Rendering Acceleration for Virtual, Augmented, and Mixed Reality”. In: *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*. VRST ’21. Osaka, Japan: Association for Computing Machinery, 2021.

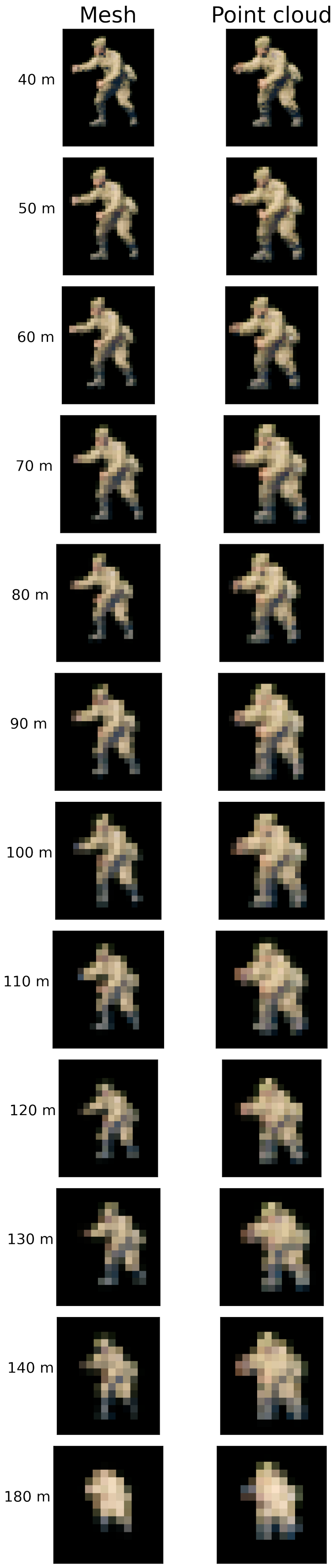
[NFS22] Anton Nikolaev, Vladimir Frolov, and Alexandr Shcherbakov. “Mesh compression method with on-the-fly decompression during rasterization and streaming support”. en. In: 2022, pp. 209–216.

[PGA11] Ruggero Pintus, Enrico Gobbetti, and Marco Agus. “Real-time rendering of massive unstructured raw point clouds using screen-space operators”. In: *Proceedings of the 12th International Conference on Virtual Reality, Archaeology and Cultural Heritage*. VAST’11. Prato, Italy: Eurographics Association, 2011, pp. 105–112.

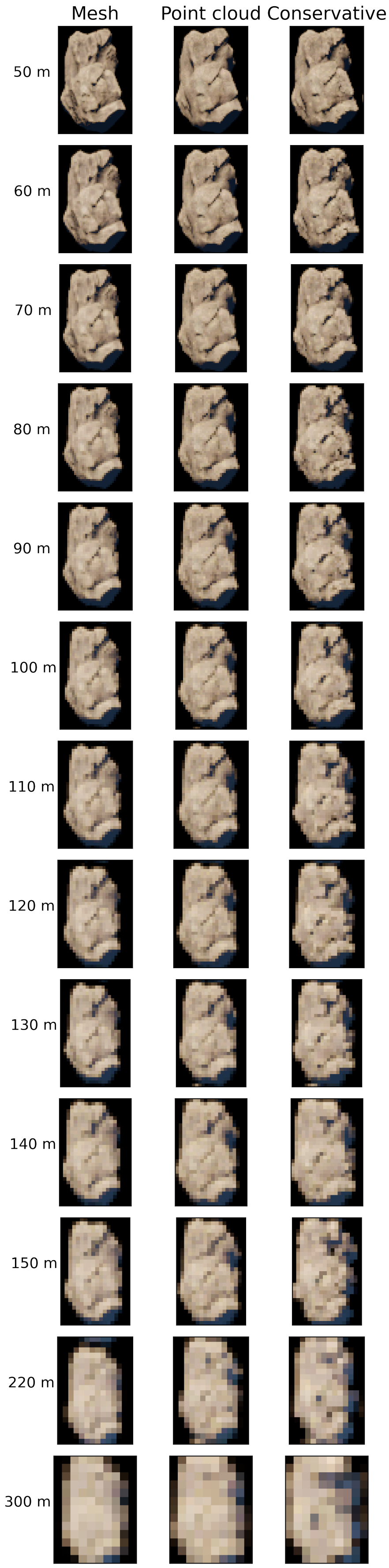
[SKW22] Markus Schütz, Bernhard Kerbl, and Michael Wimmer. “Software Rasterization of 2 Billion Points in Real Time”. In: *Proc. ACM Comput. Graph. Interact. Tech.* 5.3 (July 2022).

[Wim15] Michael Wimmer. “Rendering Large Point Clouds in Web Browsers”. In: 2015.

[WWS01] Michael Wimmer, Peter Wonka, and François Sillion. “Point-Based Impostors for Real-Time Visualization”. In: Jan. 2001, pp. 163–176.



(a) Image comparison. Left is a high polygonal mesh. Right is a point cloud



(b) Image comparison. Left is a low polygonal mesh. Middle is a point cloud. Right is conservative rasterization for point cloud

ISSN 2464-4617 (print)
ISSN 2464-4625 (online)
Computer Science Research Notes - CSRN 3401
http://www.wscg.eu
WSCG 2024 Proceedings

Towards Multi-Species Animal Re-Identification

Maik Fruhner
University of Applied Sciences Osnabrueck
Albrechtstrasse 30
49076, Osnabrueck, Germany
m.fruhner@hs-osnabrueck.de

Prof. Dr. Heiko Tapken
University of Applied Sciences Osnabrueck
Albrechtstrasse 30
49076, Osnabrueck, Germany
h.tapken@hs-osnabrueck.de

ABSTRACT

Animal Re-Identification (ReID) is a computer vision task that aims to retrieve a query individual from a gallery of known identities across different camera perspectives. It is closely related to the well-researched topic of Person ReID, but offers a much broader spectrum of features due to the large number of animal species. This raises research questions regarding domain generalization from persons to animals and across multiple animal species. In this paper, we present research on the adaptation of popular deep learning-based person ReID algorithms to the animal domain as well as their ability to generalize across species. We introduce two novel datasets for animal ReID. The first one contains images of 376 different wild common toads. The second dataset consists of various species of zoo animals. Subsequently, we optimize various ReID models on these datasets, as well as on 20 datasets published by others, with the objective of evaluating the performance of the models in a non-person domain. Our findings indicate that the domain generalization capabilities of OSNet AIN extend beyond the person ReID task, despite its comparatively small size. This enables us to investigate real-time animal ReID on live video data.

Keywords

re-identification, deep learning, computer vision, animals

1 INTRODUCTION

Re-identification (ReID) within computer vision pertains to the identification of individuals among various images of different camera angles. The complexity arises from diverse factors like pose, lighting, obstructions and appearance discrepancies, such as alterations in clothes, accessories, hairstyles in humans, or shifts in fur, feather patterns, and skin in the animal domain. To tackle this challenge, modern ReID systems commonly employ deep learning algorithms to extract image features, followed by a similarity measure to determine matches.

The ReID of animals is an active field of research [Rav+20] that faces challenges due to the sheer diversity and different appearance of the various species and the fact that they are often difficult to distinguish within a species by non-experts. Because of this, current animal ReID literature mostly focuses on a single species with manually crafted features such as skin landmarks, scars, fur patterns and face recognition. Only few papers show feasible results in a cross species setup.

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.

Our work addresses the adaptation and optimization of various person ReID algorithms to the animal domain. We demonstrate the effectiveness of established CNN-based person ReID algorithms on two datasets created by our own as well as several open source datasets. Our new datasets are made public to the research community with download links provided in the summary.

2 RELATED WORK

Image-based re-identification of animals has been an active research topic for many years. Photo identification of animals can be traced back to 1996, when Raj investigated the possibilities of recognizing wild marine animals over several years by hand [Raj98].

Methods for animal ReID based on artificial neural networks were not introduced until years later. Especially the rise of CNNs has brought new ideas and possibilities into the field of re-identification in general. In the following we present related work based on the person and animal ReID tasks. Our research focuses on the domain generalization between persons and animals and

https://www.doi.org/10.24132/CSRN.3401.15

not on the development of completely new algorithms for animal ReID, as AI models for human ReID have already proven successful. We therefore try to build upon these findings instead of starting from scratch. Over the past years ReID-specific models have been developed for the person recognition task. Some well performing ones have been implemented in the highly popular Torchreid [Zho+19a] framework, which was used for our study.

Yu et al. [Yu+17] present a ReID model based on ResNet-50, in which not only the high-level features of the output layer are used. Instead, the modified architecture has a parallel branch in the last residual block, which taps the results of the two penultimate layers. According to the authors, these should contain the *mid-level* features. To calculate the overall result, the features of all three final layers are combined before the loss function is applied.

The “Multi-Level Factorization Net” (MLFN), presented by Chang et al. [Cha+18], is based on the idea that more features are needed for a robust ReID than a camera image from a single perspective. The researchers investigated the possibility of automatically learning and finding view-independent discriminative features and combined their results in a new network architecture.

Sun et al. [Sun+18] use an approach that internally divides the image into several areas in order to examine and compare important features at the part level. The part-based convolutional baseline (PCB) network splits an input image into p different fragments, which are stacked vertically to represent different body parts. For $p = 6$ these can be head, shoulders, chest, hips, legs and shoes. These sections are used to compare them with the corresponding parts of other images.

Li et al. [Li+18] investigated the problem that people are not always perfectly aligned within their bounding boxes. The team addressed this phenomenon using the attention mechanism. A novel module for *Harmonious Attention* (HA) is able to learn hard and soft attentions, which are tailored for coarse and detailed features respectively.

Zhou et al. [Zho+19b] state that features are to be found not only on multiple, but on all scaling levels. They therefore define an “omni-scale” approach, which is a hybrid of different homogeneous and heterogeneous scaling features. Based on this approach the authors present the Omni Scale Network (OSNet). A novel deep convolutional network family, which is an order of magnitude smaller than ResNet-50, but at the same time achieves better results in the ReID task. According to the authors, “omni-scale feature learning” also proves to be a useful approach for other computer vision tasks. The OSNet [Zho+19b] and OSNet AIN [Zho+21] model families have shown outstanding results in the person ReID task and multi-dataset domain generalization scenarios.

Like many current approaches, the latest ReID advances are based on transformers. Vision transformers (ViT) [Dos+20] show remarkable results in various computer vision tasks, although they have not been researched as long as CNNs. An early attempt to address person ReID via ViT is TransReID [He+21]. The authors justify this fundamentally new strategy by arguing that ViT has the advantage over CNN approaches of being able to better understand the global context of the image input and also to better recognize fine details. This approach was recently further improved to the SOLIDER architecture [Che+23], which uses the SwinTransformer presented by Microsoft [Liu+21].

Ravoor et al. conducted a survey on animal ReID [Rav+20] and mention several studies that use person ReID models for this topic. They found that (variants of) PCB and ResNet50 were frequently used for feature extraction and as backbones, respectively. However, they conclude that PCB might not be suitable for animal ReID due to its vertical structure intended for analyzing the human upright pose.

Schneider et al. compared the siamese and triplet-loss similarity methodologies based on different CNN architectures [Sch+20] for the animal ReID task. They used one person dataset and four animal datasets and found that the triplet-loss comparisons can outperform human observers for the selected datasets.

A notable development for animal ReID is *MegaDescriptor* presented at the beginning of 2024 by Cermák et al. [Čer+24]. MegaDescriptor is intended to be a foundation model that can solve many computer vision tasks relating to animals, including ReID. The authors show impressive results across 29 public datasets. However, the authors treat animal ReID as a closed world classification problem, where all the animals to be found in the gallery set are already present during training. In the person ReID setting we adopt, training and evaluation sets are disjoint, so no ID specific features can be learned by the model.

3 DATASETS

The difference between the ReID of people and animals lies in the diversity of appearances of different animal species and the method of data acquisition. While the task for humans mostly involves processing pedestrians on surveillance cameras, the development of animal focused algorithms is much more diverse due to many factors.

3.1 Public

An increasing number of animal datasets with annotations on the identity of the individuals can be found online. Due to permissive licenses, they are often also

Dataset Name	# Images	# IDs
AerialCattle2017 [And+17]	46340	23
ATRW [Li+19]	5415	182
BelugaID [Lil22a]	5902	788
Cows2021 [Gao+21]	8670	181
FriesianCattle2015 [Til+16]	377	40
FriesianCattle2017 [And17]	940	89
GiraffeZebraID [Par+17]	6925	2056
HappyWhale [Che+22]	51033	15587
HumpbackWhaleID [How+18]	15697	5004
HyenaID2022 [Lil22b]	3129	256
IPanda50 [Wan+21]	6874	50
LeopardID2022 [Lil22c]	6806	430
NDD20 [Tro+20]	2657	82
NOAARightWhale [Chr15]	4544	447
NyalaData [Dla+20]	1942	237
OpenCows2020 [Wil+20]	4736	46
SealID [Nep+22]	2080	57
SeaTurtleID [Ada+24]	7774	400
StripeSpotter [Lah+11]	820	45
WhaleSharkID [Hol+09]	7693	543
ZindiTurtleRecall [Zin23]	12803	2265

Table 1: Evaluated public datasets

available for further research. In most cases, a dataset contains animals of exactly one species. A database that shows and annotates different species in multiple videos was published by Kuncheva et al. [Kun+22]. They aggregated a dataset on pigs, koi and pigeons with a total of 93 identities.

However, a problem with using and combining many public datasets is that often each research team publishes their data in a non standard format. As a result, a great effort of pre-processing work is required to integrate all the necessary datasets into the training process. This problem was addressed by Cermak et al. with the fairly new Wildlife Toolkit [Čer+24]. The framework bundles various datasets into a unified Python API. This allows researchers to download and use public animal ReID datasets in a streamlined workflow without the need for manual data pre-processing and conversion.

For our work, we selected 21 medium to large-sized datasets showing whole bodies of wild animals, zoo animals and farm animals. Datasets containing very few individuals or only showing animal faces were not considered. The evaluated datasets and their references are listed in table 1.

3.2 Ours

Additionally, we introduce two novel datasets for animal re-identification. The first one, ToadID [Fru+24b], contains images of 376 individual common toads from different camera angles. The second one is named ZooMixID [Fru+24a] and contains images of 180 animals of five different species.

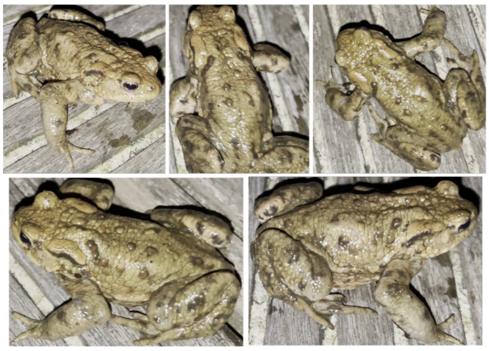


Figure 1: A toad from the ToadID dataset captured from five camera angles

Perspective	# of Images
Front	1513
Left	983
Right	1025
Back	985
Top	2739
Total	**7245**

Table 2: Summary of the ToadID dataset

3.2.1 *ToadID*

During the spring seasons of 2022 and 2023, a conservation effort in southern Lower Saxony, Germany, led to the rescue of more than 400 toads at a local lake. As there is currently no public dataset about toads available, these animals were recorded on video, before they were released at their natural habitats. Each video was carefully crafted to showcase only one toad at a time from various angles, all under one minute in length. Out of the total videos produced, 376 were deemed suitable for use, providing an equal number of unique toad identities for the research dataset.

Videos were processed to extract frames at a rate of one frame per second. A select subset of these frames received bounding box annotations to facilitate the creation of a preliminary object detection dataset. These annotated frames were used in training a Yolov5m object detector, which was subsequently utilized to extract the animals from the remaining images.

The result of this effort is a comprehensive dataset containing 7,245 unique images, representing 376 distinct identities of common toads. These images are categorized according to five different camera perspectives as listed in table 2: front, back, top, left, and right. Figure 1 gives an example of the dataset by displaying images of a single toad identity captured from all five viewpoints.

3.2.2 *ZooMix*

The objective of the second dataset is to present a greater ReID challenge by being smaller in size while

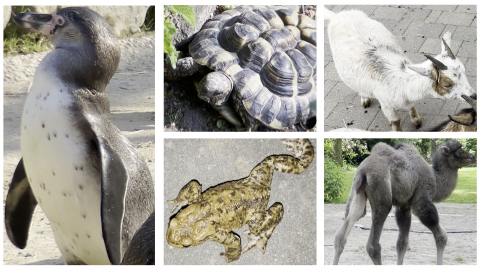


Figure 2: Examples of each species from the ZooMix multi domain dataset

Species	# of Images	# of IDs
Camel	92	5
Goat	144	30
Penguin	149	24
Toad	183	50
Tortoise	272	51
Total	**840**	**160**

Table 3: Summary of the ZooMix dataset

at the same time ranging across multiple animal domains. It serves as the basis for exploring two specific hypotheses. The first hypothesis questions whether a re-identification task remains feasible with a limited amount of training data. The second hypothesis examines whether the inclusion of highly distinct species benefits or hinders the training process, specifically whether it enhances the overall outcome by providing diversity or if it introduces complications that degrade the performance for individual species.

It contains 840 images featuring 160 individual animals of five distinct species: tortoise, camels, penguins, goats, and a selection of toads from the previous dataset. With the exception of the toads, these animals were filmed over several weeks at a local zoo. Identifying animals in a zoo with computer vision might be of interest for the employees to support their daily tasks. However, unlike the scenario with the toads, filming each animal individually was impractical due to the zoo's environment, necessitating the subsequent extraction of individuals through manual annotation. figure 2 presents an example of each species. The dataset's composition is detailed in table 3.

4 EXPERIMENTS

Our experiments on the transfer of personen ReID algorithms to the animal domain were carried out on the university's HPC cluster containing multiple A100 GPUs using the public as well as own datasets described above. To save time and computational resources, not all possible permutations of models and hyper parameters were tested on all datasets. Instead, the ToadID dataset was used in a grid search to generally determine whether there are person ReID models that are suitable for the identification of animals. The best performing model was then also trained on the remaining datasets.

The model architectures examined are (see chapter 2):

- Harmonious Attention CNN (hacnn) [Li+18]
- Multi-level Factorisation Net (mlfn) [Cha+18]
- Omni-Scale Net (osnet) [Zho+19b]
- Omni-Scale Net with Batch Normalization (osnet_ibn) [Zho+19b]
- Omni-Scale Net with Instance Normalization (osnet_ain) [Zho+19b]
- Part-based convolutional baseline (pcb) [Sun+18]
- Resnet50 with Mid-level Representations (resnet50mid) [Yu+17]

The OSNet models come in different scales, later indicated by an x, followed by a scaling factor. The PCB model was used with $p = 4$ and $p = 6$, representing the number of parts used for splitting the inputs.

A crucial step that strongly influences the outcome of an experiment is the organization of the input datasets. In our study, we used Torchreid's Train/Query/Gallery approach. In this scenario, the individuals in the training dataset are disjoint from those in the reference gallery. Therefore, no animal seen during the evaluation was seen while training before. This results in the model learning general features and patterns rather than the details of individual identities.

Each dataset and model combination was trained in three different training/test splits, which are 75/25, 50/50 and 25/75. We expect that the larger the test split, the more difficult the task becomes, as there are not only fewer identities to train on, but also more identities to choose from when testing.

There are also several approaches for distributing the remaining data to the query and gallery datasets. It must be decided whether an identity can appear multiple times in the gallery, which increases the probability of finding a correct match (by chance). This is called a multi-shot gallery, in contrast to the single-shot gallery, which contains only one image per identity. In our experiments, we investigate both scenarios, where each identity in the test set is represented by exactly one image in the query set (i.e., each animal must be found once in the gallery). During evaluation we consider this task to be a closed world scenario, meaning that each individual in the query can be found inside the gallery. An open world task, where unknown identities might appear, will be studied in future experiments.

Due to the nature of the different postures of animals, we have adapted the input layer of all CNNs. Person ReID models usually define a rectangular, portrait-oriented input layer to depict standing persons in a minimal bounding box. For the transfer to the animal domain, we decided to use a uniform, square input layer, as animals might appear in any orientation. The ReID models that already define a fixed input size were adapted accordingly. Tests were carried out with 256x256 pixels input size. Preliminary tests showed that increasing the input size to 512x512 had no positive effect on the results, apart from a huge increase in allocated VRAM and longer training times.

In summary, a total of 14 models were studied in a grid search over 30 epochs each with a square input shape of 256x256 pixels. The following hyper parameter permutations were evaluated resulting in a total of 672 runs for the model search:

- randomly initialized weights vs. weights pre-trained on person datasets
- softmax vs. triplet loss functions
- single-shot vs. multi-shot gallery setting
- sequential vs. random data sampler
- 75/25, 50/50 and 25/75 data splits

For the evaluation metric, we used the ReID ranking system typical for persons. For this metric, feature vectors are calculated from input images by the convolutional neural networks. These can be compared by measuring their distances (Euclidean in our case) in a high-dimensional space. The distances are ranked in ascending order, resulting in a top-k list of predictions. We report the Rank-1/-5/-10 results of our experiments. All training runs were carried out with deterministic calculation modes of all relevant software components in order to make the results comparable between models and datasets as well as reproducible by others.

5 RESULTS

We present our results in two sections. First, we show how the different models performed on our ToadID dataset in order to deduce which models might be generally suitable for animal ReID. Then we highlight the test results of the other datasets on the best model.

5.1 Model Search

The model search revealed some clear insights into the potentials for a domain adaptation between persons and animals. We summarize the results, as not all 672 runs can be displayed here. Firstly, pre-training on the human domain clearly helps the models listed in chapter 4 to recognize animals as well. The top 10 models in the search results all used pre-trained weights, while all but a few of the randomly initialized models occupied the last ranks. No model without pre-training achieved an mAP and Rank-1 score greater than 48.6 and 71.3 respectively in the case of a 25/75 data split and a multi-shot setting. We found that the single-shot scenario removed too many gallery images from the task, as each individual is only shown in one image, making the task much simpler. Therefore, we chose to use the multi-shot setting with pre-trained weights for the remaining experiments.

Model	mAP	R-1	R-5	R-10
hacnn	79.0	92.2	98.9	98.9
mlfn	65.5	81.1	90.0	93.3
osnet_ain_x0_25	83.0	95.6	98.9	100.0
osnet_ain_x0_5	90.9	97.8	100.0	100.0
osnet_ain_x0_75	93.6	98.9	98.9	100.0
osnet_ain_x1_0	94.4	98.9	100.0	100.0
osnet_ibn_x1_0	74.4	87.8	97.8	97.8
osnet_x0_25	83.8	94.4	100.0	100.0
osnet_x0_5	91.6	97.8	100.0	100.0
osnet_x0_75	92.6	96.7	98.9	98.9
osnet_x1_0	94.8	98.9	100.0	100.0
pcb_p4	87.2	95.6	97.8	98.9
pcb_p6	88.6	95.6	96.7	96.7
resnet50mid	92.5	97.8	100.0	100.0

Table 4: Model search results with a data split of 75/25, pretrained weights and a multi-shot setting

As mentioned in chapter four, the split had an immense impact on the reported model performances. With much training data and few query / gallery IDs, almost all models achieved high ranking scores, as shown in table 4. This also holds true for the 50/50 split shown in table 5. However, the results become more meaningful as soon as the number of training samples is reduced to 25% and the number of possible individuals in the gallery is increased. Table 6 shows the corresponding outcomes of a model search using a random data sampler, triplet loss and pretrained weights for a multi-shot gallery containing 75% of the animal IDs. It can be seen that the OSNet family in particular continues to achieve high scores, while the results of other model architectures seem to fall off.

5.2 Single Dataset

Based on the results of the model search, the generalization to different animal domains can be investigated. In addition to our two datasets, 20 public datasets were used to train OSNet AIN (osnet_ain_x1_0). A dataset split of 20/80 was used in accordance with the training configuration of the WildlifeToolkit authors. The results are listed in table 7. The datasets marked with an asterisk have been modified to make them more suitable for the train/query/gallery evaluation method. Although they show the animals from several camera angles, the viewing angles are so drastically different that a match-

Model	mAP	R-1	R-5	R-10
hacnn_square	63.1	84.3	92.4	95.1
mlfn	52.4	72.4	86.5	93.0
osnet_ain_x0_25	71.1	89.7	96.2	99.5
osnet_ain_x0_5	81.8	94.6	100.0	100.0
osnet_ain_x0_75	84.0	95.7	97.8	99.5
osnet_ain_x1_0	89.1	95.7	98.9	100.0
osnet_ibn_x1_0	57.9	80.0	91.9	94.1
osnet_x0_25	71.9	89.7	95.1	97.8
osnet_x0_5	82.7	94.1	98.4	99.5
osnet_x0_75	84.4	94.6	99.5	100.0
osnet_x1_0	89.2	98.4	98.9	100.0
pcb_p4	77.1	90.3	95.7	97.8
pcb_p6	79.8	92.4	96.2	97.8
resnet50mid	83.5	94.6	98.9	99.5

Table 5: Model search results with a data split of 50/50, pretrained weights and a multi-shot setting

Model	mAP	R-1	R-5	R-10
hacnn	46.8	70.9	84.4	88.3
mlfn	28.6	52.1	69.1	77.0
osnet_ain_x0_25	57.7	83.7	92.6	95.7
osnet_ain_x0_5	63.1	85.5	93.6	96.8
osnet_ain_x0_75	66.7	86.2	94.7	**97.5**
osnet_ain_x1_0	**71.7**	**90.4**	94.3	**97.5**
osnet_ibn_x1_0	35.8	58.9	75.2	81.6
osnet_x0_25	55.9	78.7	93.3	95.4
osnet_x0_5	64.3	83.3	93.3	95.7
osnet_x0_75	67.4	85.8	94.7	96.1
osnet_x1_0	70.2	87.6	**95.0**	97.2
pcb_p4	57.9	77.0	87.6	92.2
pcb_p6	57.3	78.4	91.8	94.7
resnet50mid	65.5	85.8	94.0	95.7

Table 6: Model search results with a data split of 25/75, pretrained weights and a multi-shot setting

ing was not possible. Therefore, only the camera angle with the most images was retained.

Our ToadID dataset again achieved a high rank-1 performance of 85.0%, while the model did not reach more than 50% rank-1 accuracy for any other dataset. Compared to the performance of OSNet AIN on small and large person ReID datasets, it can be said that domain adaptation works. Zhou et al. [Zho+21] report 38.3%, 68.0% and 86.6% rank-1 scores on the small GRID, VIPeR and CUHK01 datasets, respectively. Rank-1 results of 94.8%, 72.3% and 88.7% were obtained for the large datasets Market1501, CUHK03 and Duke, respectively. Considering that many animal ReID datasets contain less than a hundred IDs, some between 200 and 1000, while very few datasets contain more than a thousand different animals, the overall results of the training runs for single datasets show solid performances.

dataset	mAP	r1	r5	r10
ToadID (ours)	62.6	85.0	92.4	95.0
OpenCows2020	54.0	48.6	62.2	62.2
ATRW	52.1	47.2	56.0	60.4
Cows2021	52.0	45.5	57.2	66.2
StripeSpotter	31.8	43.8	59.4	65.6
FriesianCattle2017	49.0	39.1	60.9	67.2
HyenaID2022	19.0	36.5	58.9	66.0
SeaTurtleIDHeads	12.9	35.7	53.6	60.4
GiraffeZebraID	31.0	35.3	41.8	45.4
ZooMix (ours)	43.6	34.9	50.8	65.9
HumpbackWhaleID	32.8	26.1	39.0	45.3
LeopardID2022	15.1	21.8	38.2	44.7
FriesianCattle2015	33.4	18.8	46.9	84.4
ZindiTurtleRecall	7.7	17.1	46.3	54.9
WhaleSharkID	8.9	15.2	29.6	36.3
SealID	15.4	13.3	15.6	20.0
BelugaID*	16.9	11.6	22.2	27.8
NOAARightWhale	10.8	10.0	11.4	12.0
AerialCattle2017	14.1	10.0	20.0	20.0
HappyWhale	11.7	8.3	14.3	17.8
NyalaData	6.1	6.4	18.1	27.1
NDD20*	6.6	3.1	9.4	14.1
IPanda50	3.3	2.5	2.5	5.0

Table 7: Results for single dataset training runs

5.3 Multi Dataset

Interesting effects were observed when different datasets are combined in a training run and jointly influence the learning process of osnet_ain_x1_0. Using multiple datasets from the same species (e.g. cattle) as the training sources results in a significantly larger number of images to learn from. As a result, all tested cattle datasets receive a massive improvement in rank-1 scores when used together, as opposed to when used for training individually. The improved results are displayed in table 8.

However, the combination of seemingly independent datasets can also lead to an improvement in model performance. While blindly joining all datasets does not improve the model's performance, successes were achieved when merging datasets of somewhat visually related species. As shown in table 9 animals living on land and in the sea were combined in two experiments, respectively. In the land inhabitants, the rank-1 results of half of the species were improved, while the other half declined minimally. Three quarters of the results of the evaluated datasets improved for the marine species. We observed that overall, datasets with good recognition scores can help the weaker, usually smaller datasets. We assume that the large increase in training data makes it easier to train the feature extractors and that the learned features are therefore (at least partially) transferable between the domains.

dataset	mAP	r1	r5	r10
OpenCows2020	61.1	59.5	62.2	62.2
FriesianCattle2017	66.8	59.4	73.4	82.8
Cows2021	56.5	51.0	63.4	70.3
AerialCattle2017	45.1	45.0	45.0	45.0
FriesianCattle2015	44.6	31.2	56.2	90.6

Table 8: Results improve when combining datasets from the same domain (cattle).

dataset	mAP	r1	r5	r10
ToadID	62.9	82.7	91.0	93.4
HyenaID2022	21.7	**38.6**	52.3	62.9
StripeSpotter	32.0	34.4	56.2	59.4
GiraffeZebraID	29.1	32.4	40.2	47.1
LeopardID2022	18.3	**24.8**	41.6	46.9
NyalaData	8.5	**10.1**	23.4	38.3
HumpbackWhaleID	30.0	23.6	36.4	42.4
WhaleSharkID	8.7	**16.6**	27.7	35.2
BelugaID	17.8	**12.4**	23.1	27.8
NDD20	11.1	**7.8**	12.5	15.6

Table 9: Results can improve when combining datasets from different domains (Top: *Land*, Bottom: *Marine*). Rank-1 improvements (highlighted in bold) of up to almost 5% can be observed.

6 CONCLUSION

6.1 Summary

In this paper we investigated the transferability of algorithms for person ReID to animals of different species. Using a cross-search through different CNN-based models and hyper parameters, the family of OSNets was found to be suitable. We applied OSNet AIN to over 20 different datasets, two of which we created ourselves.

While small datasets suffer from too few training examples, some larger marine datasets present a major challenge with the task of matching only fins or fins with underwater images. The application of the comparison of two images from different perspectives - as it's defined for person ReID - has only proven successful for some datasets. Best results were achieved when the animal ReID task was closer to the human domain. Our ToadID dataset with high-resolution, pre-cropped images showing feature-rich animal textures achieved the highest rank-1 results in our experiments.

OSNet AIN achieves reasonable results with its 2.2M parameters in the standard configuration without much customization. The rank-1 scores across the several animal datasets are comparable to those across person ReID datasets. In some cases, ReID performance can be improved by combining multiple datasets from different animal species.

6.2 Discussion

Due to the different approaches of individual research teams, such as the creation and splitting of datasets, the use of randomized or deterministic calculations and the way experiments and their results are presented, findings are difficult to compare.

Although many wildlife datasets have been streamlined into a single API framework, they still have very different structures and content. Some datasets contain fully cropped images, others provide full images with bounding box or even segmentation annotations. Therefore, not every dataset can be effectively combined with others to train a common task.

Furthermore, the reported model performances cannot be easily compared with each other, as research teams use different evaluation measures for the ReID of animals. Many see the task as a closed-world classification problem, where the training IDs should be identified in a reference set during testing. However, in person ReID, and thus in our context, the training and test IDs are disjoint, which makes the task much more difficult.

Finally, for many datasets, there is no default split between training and testing data, so other researchers have to create their own split. The WildlifeToolkit, for example, applies an automatically generated split to each dataset for the closed-world scenario. However, this split does not fit our setting in the person ReID transfer context. This forced us to create a different split using the disjoint split method, which naturally leads to a rather unrelated research question.

6.3 Outlook

The potential of omni-scale learning will be further investigated in subsequent experiments. More in-depth hyperparameter searches and investigations into adjustments to the network architecture are just two starting points for further improving animal ReID using OSNets.

A third dataset is being developed that differentiates pigs using top-down recordings. Pig farming is a good example of the relevance of the closed-world scenario examined in this paper, in which no unknown individuals can occur. The wildlife datasets, on the other hand, already indicate their open-world setting in their name, which is also part of our further research. In addition to publishing our datasets, we will contribute to the WildlifeToolkit to integrate the datasets directly so that other researchers can easily use them.

7 REFERENCES

[Ada+24] Lukáš Adam et al. "SeaTurtleID2022: A Long-Span Dataset for Reliable Sea Turtle Re-Identification". In: *Proceedings of* *the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2024, pp. 7146–7156.

[And+17] William Andrew, Colin Greatwood, and Tilo Burghardt. “Visual Localisation and Individual Identification of Holstein Friesian Cattle via Deep Learning”. In: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2017, pp. 2850–2859.

[And17] Will Andrew. *FriesianCattle2017*. 2017.

[Čer+24] Vojtěch Čermák et al. “WildlifeDatasets: An Open-Source Toolkit for Animal Re-Identification”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2024, pp. 5953–5963.

[Cha+18] Xiaobin Chang, Timothy M. Hospedales, and Tao Xiang. “Multi-level Factorisation Net for Person Re-identification”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE, 2018, pp. 2109–2118. ISBN: 978-1-5386-6420-9.

[Che+22] Ted Cheeseman et al. *Happywhale - Whale and Dolphin Identification*. 2022. URL: `https://kaggle.com/competitions/happy-whale-and-dolphin`.

[Che+23] Weihua Chen et al. “Beyond Appearance: A Semantic Controllable Self-Supervised Learning Framework for Human-Centric Visual Tasks”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE, 2023, pp. 15050–15061. ISBN: 979-8-3503-0129-8.

[Chr15] Wendy Kan Christin B. Khan Shashank. *Right Whale Recognition*. 2015. URL: `https://kaggle.com/competitions/noaa-right-whale-recognition`.

[Dla+20] Nkosikhona Dlamini and Terence L van Zyl. “Automated Identification of Individuals in Wildlife Population Using Siamese Neural Networks”. In: *2020 7th International Conference on Soft Computing & Machine Intelligence (ISCMI)*. 2020, pp. 224–228.

[Dos+20] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations* (2020).

[Fru+24a] Maik Fruhner, Heiko Tapken, and Eva Stroetmann. *Images of 180 individual zoo animals*. 2024. URL: `https://doi.pangaea.de/10.1594/PANGAEA.967637`.

[Fru+24b] Maik Fruhner, Heiko Tapken, and Eva Stroetmann. *Images of 376 individuals of common toads (Bufo bufo) from southern Lower Saxony, Germany*. 2024. URL: `https://doi.pangaea.de/10.1594/PANGAEA.967135`.

[Gao+21] Jing Gao et al. *Towards Self-Supervision for Video Identification of Individual Holstein-Friesian Cattle: The Cows2021 Dataset*. 2021. arXiv: `2105.01938 [cs.CV]`.

[He+21] Shuting He et al. “TransReID: Transformer-based Object Re-Identification”. In: *2021 IEEE/CVF International Conference on Computer Vision*. Ed. by Eric Mortensen. Piscataway, NJ: IEEE, 2021, pp. 14993–15002. ISBN: 978-1-6654-2812-5.

[Hol+09] Jason Holmberg, Brad Norman, and Z Arzoumanian. “Estimating population size, structure, and residency time for whale sharks Rhincodon typus through collaborative photo-identification”. In: *Endangered Species Research* 7 (Apr. 2009), pp. 39–53.

[How+18] Addison Howard, Ken Southerland, and Ted Cheeseman. *Humpback Whale Identification*. 2018. URL: `https://kaggle.com/competitions/humpback-whale-identification`.

[Kun+22] Ludmila I. Kuncheva et al. “A Benchmark Database for Animal Re-Identification and Tracking”. In: *The Fifth IEEE International Image Processing, Applications and Systems Conference (IPAS’22)*. Piscataway, NJ: IEEE, 2022, pp. 1–6. ISBN: 978-1-6654-6219-8.

[Lah+11] Mayank Lahiri et al. “Biometric animal databases from field photographs: identification of individual zebra in the wild”. In: *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*. ICMR ’11. Trento, Italy: Association for Computing Machinery, 2011. ISBN: 9781450303361. URL: `https://doi.org/10.1145/1991996.1992002`.

[Li+18] Wei Li, Xiatian Zhu, and Shaogang Gong. "Harmonious Attention Network for Person Re-identification". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE, 2018, pp. 2285–2294. ISBN: 978-1-5386-6420-9.

[Li+19] Shuyuan Li et al. *Amur Tiger Re-identification in the Wild*. June 13, 2019.

[Lil22a] Lilawp. *Beluga ID 2022 - LILA BC*. 2022. URL: `https://lila.science/datasets/beluga-id-2022/`.

[Lil22b] Lilawp. *Hyena ID 2022 - LILA BC*. 2022. URL: `https://lila.science/datasets/hyena-id-2022/`.

[Lil22c] Lilawp. *Leopard ID 2022 - LILA BC*. 2022. URL: `https://lila.science/datasets/leopard-id-2022/`.

[Liu+21] Ze Liu et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows". In: *2021 IEEE/CVF International Conference on Computer Vision*. Ed. by Eric Mortensen. Piscataway, NJ: IEEE, 2021, pp. 9992–10002. ISBN: 978-1-6654-2812-5.

[Nep+22] Ekaterina Nepovinnykh et al. "SealID: Saimaa Ringed Seal Re-Identification Dataset". In: *Sensors* 22.19 (2022). ISSN: 1424-8220.

[Par+17] Jason Parham et al. "Animal population censusing at scale with citizen science and photographic identification". In: AAAI Spring Symposium - Technical Report. United States: AI Access Foundation, 2017, pp. 37–44.

[Raj98] Lesley Raj. "Photo-identification of Stichopus mollis". In: *SPC Beche-de-mer Information Bulletin* 10 (1998), pp. 29–31.

[Rav+20] Prashanth C. Ravoor and Sudarshan T.S.B. "Deep Learning Methods for Multi-Species Animal Re-identification and Tracking – a Survey". In: *Computer Science Review* 38 (2020), p. 100289. ISSN: 15740137.

[Sch+20] Stefan Schneider, Graham W. Taylor, and Stefan C. Kremer. "Similarity Learning Networks for Animal Individual Re-Identification - Beyond the Capabilities of a Human Observer". In: *2020 IEEE Winter Applications of Computer Vision workshops (WACVW)*. Piscataway, NJ: IEEE, 2020, pp. 44–52. ISBN: 978-1-7281-7162-3.

[Sun+18] Yifan Sun et al. "Beyond Part Models: Person Retrieval with Refined Part Pooling (and A Strong Convolutional Baseline)". In: *15th European Conference on Computer Vision*. Ed. by Vittorio Ferrari et al. 2018, pp. 480–496.

[Til+16] Tilo Burghardt and Will Andrew. *FriesianCattle2015*. 2016.

[Tro+20] Cameron Trotter et al. *NDD20: A large-scale few-shot dolphin dataset for coarse and fine-grained categorisation*. 2020. URL: `http://arxiv.org/pdf/2005.13359.pdf`.

[Wan+21] Le Wang et al. "Giant Panda Identification". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 2837–2849. URL: `https://github.com/iPandaDateset/iPanda-50`.

[Wil+20] William Andrew et al. *OpenCows2020*. 2020.

[Yu+17] Qian Yu et al. "The Devil is in the Middle: Exploiting Mid-level Representations for Cross-Domain Instance Matching". In: *arXiv.org* (2017).

[Zho+19a] Kaiyang Zhou and Tao Xiang. "Torchreid: A Library for Deep Learning PersonRe-Identification in Pytorch". In: *arXiv preprint arXiv:1910.10093* (2019). URL: `https://arxiv.org/pdf/1910.10093.pdf`.

[Zho+19b] Kaiyang Zhou et al. "Omni-Scale Feature Learning for Person Re-Identification". In: *2019 International Conference on Computer Vision*. Piscataway, NJ: IEEE, 2019, pp. 3701–3711. ISBN: 978-1-7281-4803-8.

[Zho+21] Kaiyang Zhou et al. "Learning Generalisable Omni-Scale Representations for Person Re-Identification". In: *IEEE transactions on pattern analysis and machine intelligence* (2021).

[Zin23] Zindi. *Turtle Recall: Conservation Challenge*. 12/13/2023. URL: `https://zindi.africa/competitions/turtle-recall-conservation-challenge/data`.

Texture reading optimization for two-dimensional filter based graphics algorithms

Konstantin Zubatov
Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Moscow, 119991, Russia
konstantin.zubatov@graphics.cs.msu.ru

Alexandr Shcherbakov
Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Moscow, 119991, Russia
alex.shcherbakov@graphics.cs.msu.ru

ABSTRACT

There are many graphics algorithms that require reading textures in window 3x3, 5x5, etc. Both large window sizes and high rendering resolutions decrease performance. We propose an algorithm-independent method for reducing the number of texture samples implemented in a fragment shader for a local 2x2 area and based on the architectural features of the graphics pipeline — transferring data between fragments via difference derivatives.

Keywords
fragment shader, compute shader, difference derivatives, filters, texture sampling

1 INTRODUCTION

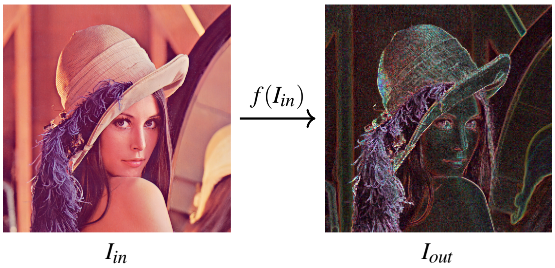


Figure 1: Filter illustration

Image filtering is widely used in computer graphics and image processing. Filters are required both as independent operations and for image processing before or after the main algorithms. Formally, we have an input image $I_{in} \in [0,1]^{w \times h \times c}$ and want to get an output image $I_{out} \in [0,1]^{w \times h \times c}$ by applying a two-dimensional filter function $f : I_{in} \mapsto I_{out}$. Usually it takes a local area (negative indices are used for convenience)

$$A(x,y) = (I_{in}(x+i, y+j))_{-r_w \le i \le r_w, -r_h \le j \le r_h}, \; r_w, r_h \in \mathbb{N}$$

of each texel $I_{in}(x,y)$ and produces single fragment $I_{out}(x,y)$, where $(x,y) \in [0, w-1] \times [0, h-1]$. In the following, only local areas of size $(2r+1) \times (2r+1)$, $r \in \mathbb{N}$, will be considered for two-dimensional filters. As the image or local area size increases, the performance decreases due to the larger number of texture samples required.

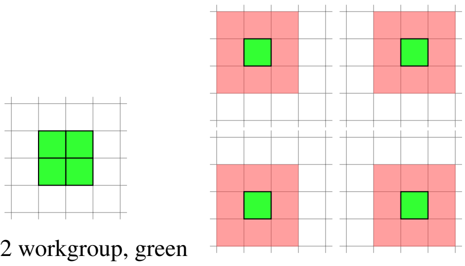
Filtering is usually done on the GPU because it is well parallelized and the result is needed for other render targets (in the case of real-time computer graphics). There are two suitable types of shaders — compute and fragment shaders.

A compute shader allows to reduce bandwidth requirements by using shared memory [Kil12]. In this case, each invocation in the workgroup reads several texels and writes them to fast shared memory located on the chip (see Figure 2), from where each invocation then reads the required texels. Practice shows that this method is not always applicable for mobile devices. For example, Arm GPUs do not implement dedicated on-chip shared memory for compute shaders. The shared memory that is available to use is system RAM that is backed up by the load-store cache [Arm].

A fragment shader does not have shared memory, but is used in rendering, so the filter can be combined with additional operations in the fragment shader. Regardless of the type of shader, another approach to optimizing memory consumption is used — two-pass rendering for separable filters. Separable filters are filters that can be written as the product of two one-dimensional

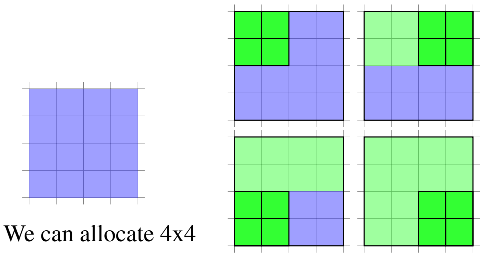
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or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

https://www.doi.org/10.24132/CSRN.3401.16



(a) 2x2 workgroup, green squares are threads

(b) Each thread runs 3x3 filter (local areas are red), invocations are separated for better readability



(c) We can allocate 4x4 array in shared memory (it is blue) and fill it with necessary texels

(d) Reading texels in four steps. The bright green squares are threads and the pale green squares are read texels

Figure 2: Example for 2x2 workgroup and 3x3 filter in compute shader. Each thread performs only 4 reads. Without shared memory, this number will increase to 9. In practice, larger workgroups (e.g. 16x16) and other sampling strategies (e.g. linear) are used.

filters. Using this feature, we can compute the same result with $4r+2$ texture samples instead of $(2r+1)^2$, but this would require an intermediate image of the same resolution and additional render pass.

2 RELATED WORKS

2.1 Commonly used filters

Discrete convolutional filters (1) are often used.

$$f(I_{in}(x,y)) = \frac{\sum_{i=-r}^{r}\sum_{j=-r}^{r} g(i,j) I_{in}(x-i,y-i)}{\sum_{i=-r}^{r}\sum_{j=-r}^{r} g(i,j)} \quad (1)$$

Moving from the classical definition of convolution to implementation in a shader, we rewrite the formula for a convolution filter

$$f(I_{in}(x,y)) = \frac{\sum_{i=-r}^{r}\sum_{j=-r}^{r} g(i,j) I_{in}(x+i,y+i)}{\sum_{i=-r}^{r}\sum_{j=-r}^{r} g(i,j)} \quad (2)$$

Hereafter in the text the filter is understood as a function (2) with different kernels $g(i,j)$, and r everywhere means the radius of the local area (hence its size is $(2r+1)\times(2r+1)$) of the filter. In this paper we will discuss four discrete convolutional filters: Gaussian filter, bilateral filter, tent filter and variance clipping for temporal anti-aliasing (TAA).

2.1.1 *Gaussian filter*

A Gaussian filter is a separable filter that is used to produce post effects (like bloom or depth-of-field), as part of Canny edge detection [Can86], and in variance shadow maps [Don06]. Filter's function is described using kernel (3) (factor $\frac{1}{2\pi\sigma^2}$ is reduced)

$$g(i,j) = \exp\left(-\frac{i^2+j^2}{2\sigma^2}\right) \quad (3)$$

where σ is the standard deviation of the Gaussian distribution (greater value — stronger blur).

2.1.2 *Bilateral filter*

A bilateral filter is described using kernel (4)

$$g(i,j) = \exp\left(-\frac{i^2+j^2}{2\sigma_d^2} - \frac{\|I_{in}(x+i,y+j)-I_{in}(x,y)\|^2}{2\sigma_r^2}\right) \quad (4)$$

where σ_d and σ_r are smoothing parameters. It is a nonlinear, edge-preserving filter. One of its uses is to prevent incorrect diffusion in skin rendering [Buc16]. Another application is image upsampling [Kop07].

2.1.3 *Tent filter*

A tent filter is described using kernel (5)

$$g(i,j) = \max(k-b|i|,\ 0)\max(k-b|j|,\ 0) \quad (5)$$

where k and b are parameters. It is used in temporal upsampling for pre-processing stage [Yan09], in image downscaling and upscaling for video encoding [Bri99], and as individual filter for blur in image processing [App]. In our solution we modified tent filter and made it nonseparable, with kernel rank 2. It is almost equal to the original filter and has kernel (6)

$$g(i,j) = k - b(|i|+|j|) \quad (6)$$

where k and b are same. This downgrade was done to test the applicability of the proposed method to the filter with rank 2.

2.1.4 *Variance clipping, TAA*

Temporal Antialiasing (also known as Temporal AA, or TAA) is a family of techniques that perform spatial antialiasing using data gathered across multiple frames. One of the important part of any TAA algorithm is history rectification — a process of adapting a color from previous frame(s). Variance clipping [Sal16] addresses

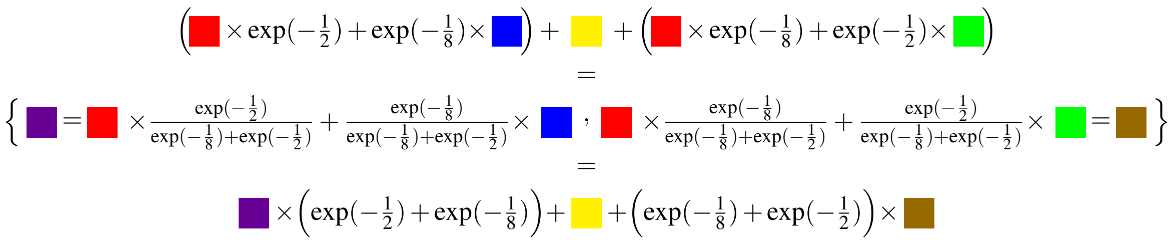

$\times \exp(-\frac{1}{2}) + \exp(-\frac{1}{8}) \times$
$+$
$+$
$\times \exp(-\frac{1}{8}) + \exp(-\frac{1}{2}) \times$
$=$
$\times \frac{\exp(-\frac{1}{2})}{\exp(-\frac{1}{8})+\exp(-\frac{1}{2})} + \frac{\exp(-\frac{1}{8})}{\exp(-\frac{1}{8})+\exp(-\frac{1}{2})} \times$
$\times \frac{\exp(-\frac{1}{8})}{\exp(-\frac{1}{8})+\exp(-\frac{1}{2})} + \frac{\exp(-\frac{1}{2})}{\exp(-\frac{1}{8})+\exp(-\frac{1}{2})} \times$
$=$
$\times \left(\exp(-\frac{1}{2}) + \exp(-\frac{1}{8})\right) +$
$+ \left(\exp(-\frac{1}{8}) + \exp(-\frac{1}{2})\right) \times$

Figure 3: [Rak10] method, $\sigma = 2$, 5x5 local area, horizontal pass, colored squares are read texels. The result in the third row is the same as in the first row, but requires three texels. The purple and brown texels in the third row are linear interpolation (see second row). The readout time of a texel for a linear sampler is independent of whether it is a real texel or a bilinear interpolation of four real texels

outliers by using the local color mean and standard deviation to center and size the color extents used for rectification:

$$C_{min} = \mu - \gamma\sigma \qquad C_{max} = \mu + \gamma\sigma$$

where μ and σ are the mean and standard deviation of the color samples in the local area, and γ is a scalar parameter, $\gamma \in [0.75, 1.25]$. Calculation of μ and γ can be written as filters (7) and (8):

$$\mu = \frac{\sum\limits_{i=-r}^{r} \sum\limits_{j=-r}^{r} I_{in}(x+i, y+j)}{(2r+1)^2} \tag{7}$$

$$\sigma = \sqrt{\frac{\sum\limits_{i=-r}^{r} \sum\limits_{j=-r}^{r} I_{in}^2(x+i, y+j)}{(2r+1)^2} - \mu^2} \tag{8}$$

2.2 Memory optimization work

[Rak10] proposed an improvement for separable Gaussian filter based on a linear texture sampler that returns a bilinear interpolation of the four nearest pixels instead of nearest pixel color for the linear sampler. Let us describe a one-dimensional filter for the first (along x-axis) pass:

$$f(I_{in}(x,y)) = \frac{\sum\limits_{i=-r}^{r} \exp(-\frac{i^2}{2\sigma^2}) I_{in}(x+i, y)}{\sum\limits_{i=-r}^{r} \exp(-\frac{i^2}{2\sigma^2})}$$

We can describe the contribution of two neighboring pixels $I_{in}(x+j, y)$ and $I_{in}(x+j+1, y)$ as:

$$\frac{I_{in}(x+j,y)w_1 + I_{in}(x+j+1,y)w_2}{\sum\limits_{i=-r}^{r} exp(-\frac{i^2}{2\sigma^2})} =$$

$$\frac{\{I_{in}(x+j,y)\frac{w_1}{w_1+w_2} + I_{in}(x+j+1,y)\frac{w_2}{w_1+w_2}\}(w_1+w_2)}{\sum\limits_{i=-r}^{r} exp(-\frac{i^2}{2\sigma^2})}$$

$$= \frac{I_{in}(\frac{(x+j)w_1+(x+j+1)w_2}{w_1+w_2}, y)(w_1+w_2)}{\sum\limits_{i=-r}^{r} exp(-\frac{i^2}{2\sigma^2})}$$

where $w_1 = \exp(-\frac{j^2}{2\sigma^2})$, $w_2 = \exp(-\frac{(j+1)^2}{2\sigma^2})$. The expression demonstrates linear interpolation multiplied by the sum of weights. This method reduces the number of texture samples from $4r+2$ to $2r+2$ in two passes (see example in Figure 3). This is a very efficient method, but it's only applicable to separable filters and requires intermediate image to store the output of the first pass.

3 PROPOSED SOLUTION

Our solution is based on the fact that the fragment shader always works for 2x2 groups of fragments. It is useful for calculating the MIP-level for a texture and provides differential derivatives along the x and y axes. They are typically used to calculate texture MIP-level (separate from texture reading), flat normals, or edge detection. We will use fine differential derivatives along the x and y axes ($dFdxFine$ and $dFdyFine$ in GLSL) and rename them $ddx(p)$ and $ddy(p)$, respectively, where p is a scalar or vector. $ddx(p)$ and $ddy(p)$ calculate derivatives using local differencing based on the value of p for the current fragment and it's immediate neighbor (see Figure 4).

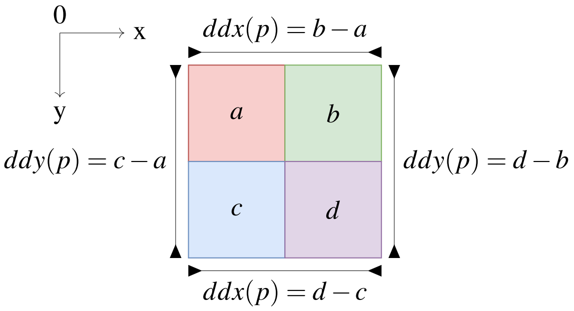


Figure 4: $ddx(p)$ and $ddy(p)$ for 2x2 group of fragments. a, b, c, d are the value of p in the corresponding fragment. Fragments in the column {row} have the same $ddy(p)$ $\{ddx(p)\}$. The x and y axes show the coordinate directions.

It is required to define t_x, t_y

$$t_x = \begin{cases} -1 & \text{if fragment in odd col} \\ 1 & \text{if fragment in even col} \end{cases}$$

$$t_y = \begin{cases} -1 & \text{if fragment in odd row} \\ 1 & \text{if fragment in even row} \end{cases}$$

so that the fragments can exchange data with their immediate neighbors:

$$p_{recvd} = p_{our} + t_x \cdot ddx(p_{our}), \text{ swap along x-axis} \quad (9)$$

$$p_{recvd} = p_{our} + t_y \cdot ddy(p_{our}), \text{ swap along y-axis} \quad (10)$$

Using data transfer, we can describe a naive algorithm that exchanges texels between fragments and reduces the number of texture reads from $(2r+1)^2$ to $(r+1)^2$ for each fragment (see Figure 5):

1. Each fragment reads texels $I_{in}(x-r+2i, y-r+2j)$, where $0 \leq i, j \leq r$. Now each fragment has $r+1$ rows with $r+1$ texels in each (see Figure 5b).
2. We can select r texels from each non-empty row and do swaps along x-axis (see Figure 5c).
3. Now we have $r+1$ full rows of texels. r rows from them are needed for swaps along y-axis (see Figure 5d).
4. After swaps along y-axis each fragment has all texels (see Figure 5e) and computes the filter function.

Unfortunately, use of fine difference derivatives for all texels transfer ($3r^2+2r$ times) works slower than $(2r+1)^2$ texture readings. To improve performance, the shader should perform multiple partial sum transfers computed with its own texels but kernel values required by its neighbours. Let us describe this method with partial sums for the four filters mentioned earlier.

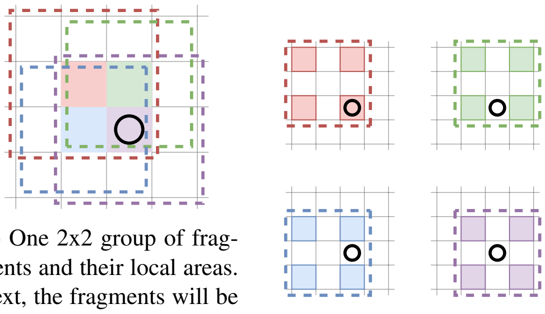
3.1 Gaussian filter

The Gaussian filter kernel (3) has rank 1, which means that all rows (columns) in kernel are a linear combination of one row (column). Therefore, it is enough to:

1. For each $i \in T$, $T = \{-r+2k \mid 0 \leq k \leq r\}$:
   (a) Read texels $I_{in}(x+j, y+i)$, $j \in T$
   (b) initialize the partial sum p_i

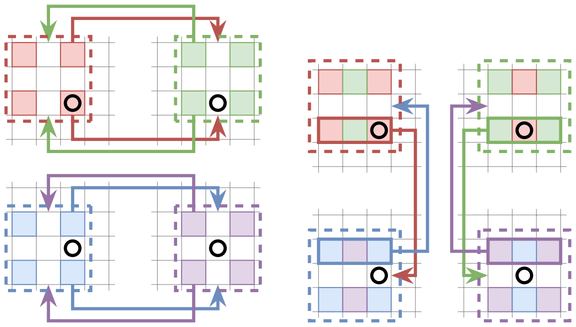
$$p_i := \sum_{j \in T} \exp\left(-\frac{i^2+j^2}{2\sigma^2}\right) I_{in}(x+j, y+i)$$

   (c) Select texels for the neighbour on the x-axis. If the fragment (which is executing the shader, not the neighbour!) has even column number, it should select all texels in row except the first one, $\Omega = T \setminus \{-r\}$. Otherwise — except the last one, $\Omega = T \setminus \{r\}$.



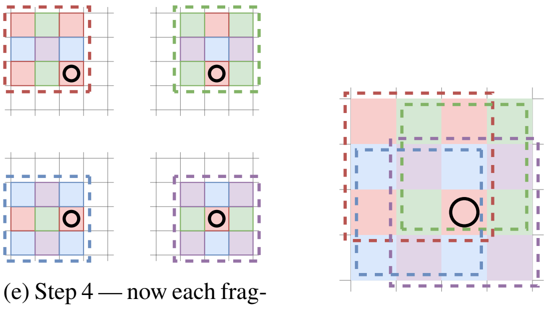
(a) One 2x2 group of fragments and their local areas. Next, the fragments will be visually separated for better readability

(b) Step 1 — read 4 texels (these are colored squares).



(c) Step 2 — select 2 required texels and do swaps along x-axis. The arrows show the transfer of the desired texels

(d) Step 3 — select required row and do 3 swaps along y-axis. The arrows and outline rectangles show transfer of the desired texels



(e) Step 4 — now each fragment contains all required texels

(f) Visual merging of fragments

Figure 5: Naive algorithm illustration for 3x3 filter. Texel with black circle is the same for all fragments. Dashed rectangles cover local areas for each fragment

   (d) Compute partial sum p'_i for the neighbour on the x-axis (consider the bias with t_x) (see Figure 6a)

$$p'_i := \sum_{j \in \Omega} \exp\left(-\frac{i^2+(j-t_x)^2}{2\sigma^2}\right) I_{in}(x+j, y+i)$$

   (e) Make a swap along the x-axis and complete the partial sum p_i

$$p_i^{final} := p_i + p'_i + t_x \cdot ddx(p'_i),$$

   (f) p_i^{final} equals $\sum_{j=-r}^{r} \exp\left(-\frac{i^2+j^2}{2\sigma^2}\right) I_{in}(x+j, y+i)$

2. initialize the partial sum of the filter

$$S := \sum_{i \in T} p_i^{final}$$

3. Choose the partial sums p_i^{final} required for the neighbour on the y-axis. If the fragment has even row number, it should select all sums p_i^{final} except the first one, $\Omega = T \setminus \{-r\}$. Otherwise — except the last one, $\Omega = T \setminus \{r\}$.

4. It was mentioned that each row of the kernel matrix (3) can be expressed as a linear combination. We can interpolate partial sums for rows that our neighbour on the y-axis does not know about by multiplying them by certain exponents. If p_i^{final} has a common factor $\exp(-\frac{i^2}{2\sigma^2})$ in this fragment, the neighbor on the y-axis has a common factor $\exp(-\frac{(i-t_y)^2}{2\sigma^2})$. Therefore, we should compute partial sum S' for the neighbor on the y-axis (see Figure 6b)

$$S' := \sum_{i \in \Omega} \exp(\frac{2it_y - 1}{2\sigma^2}) p_i$$

5. Make a swap along the y-axis and complete the sum

$$S^{final} := S + S' + t_y \cdot ddy(S')$$

6. S^{final} equals $\sum\limits_{i=-r}^{r} \sum\limits_{j=-r}^{r} \exp(-\frac{i^2+j^2}{2\sigma^2}) I_{in}(x+i, y+j)$. Normalize S^{final} by the sum of the kernel.

3.2 Tent filter

The kernel (5) is suitable for the previous method. The kernel of modified tent filter (6) has rank 2. Therefore, we need to modify the previous algorithm:

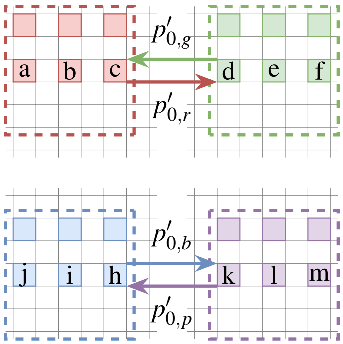
1. For each $i \in T$, $T = \{-r + 2k \mid 0 \le k \le r\}$:
   (a) Read texels $I_{in}(x+j, y+i)$, $j \in T$
   (b) initialize the partial sum p_i

   $$p_i := \sum_{j \in T} (k - b(|i| + |j|)) I_{in}(x+j, y+i)$$

   and additional partial sum u_i (required in step 4)

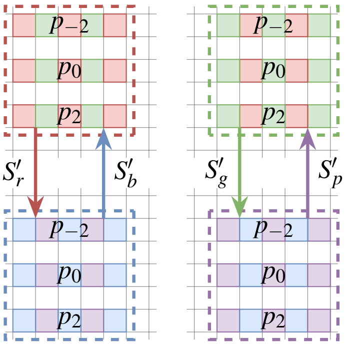
   $$u_i := \sum_{j \in T} I_{in}(x+j, y+i)$$

   (c) Select texels for the neighbour on the x-axis. If the fragment (which is executing the shader, not the neighbour!) has even column number, then it should select all texels in row except the first one, $\Omega = T \setminus \{-r\}$. Otherwise — except the last one, $\Omega = T \setminus \{r\}$.



$$p'_{0,r} = (b+c)\exp(-\tfrac{1}{2\sigma^2}),\ p'_{0,g} = (d+e)\exp(-\tfrac{1}{2\sigma^2})$$
$$p'_{0,b} = (i+h)\exp(-\tfrac{1}{2\sigma^2}),\ p'_{0,p} = (k+l)\exp(-\tfrac{1}{2\sigma^2})$$

(a) Data swap along x-axis in step 1 when $i = 0$. The letters in the squares represent texels



$$S'_r = S'_g = p_0 \times \exp(-\tfrac{1}{2\sigma^2}) + p_2 \times \exp(\tfrac{3}{2\sigma^2})$$
$$S'_b = S'_p = p_0 \times \exp(-\tfrac{1}{2\sigma^2}) + p_{-2} \times \exp(\tfrac{3}{2\sigma^2})$$

(b) Steps 4 and 5. p_{-2}, p_0, p_2 are different in each fragment

Figure 6: Proposed method for 5x5 Gaussian filter. The arrows indicate the transfer of the respective sums

   (d) Compute partial sum p'_i for the neighbour on the x-axis (consider the bias with t_x)

   $$p'_i := \sum_{j \in \Omega} (k - b(|i| + |j - t_x|)) I_{in}(x+j, y+i)$$

   $$u'_i := \sum_{j \in \Omega} I_{in}(x+j, y+i)$$

   (e) Make a swap along the x-axis and complete the partial sums p_i and u_i

   $$p_i^{final} := p_i + p'_i + t_x \cdot ddx(p'_i),$$

   $$u_i^{final} := u_i + u'_i + t_x \cdot ddx(u'_i),$$

2. initialize the partial sum of the filter

$$S := \sum_{i \in T} p_i^{final}$$

3. Choose the partial sums p_i^{final} and u_i^{final} required for the neighbor on the y-axis. If the fragment has even row number, it should select all sums except p_{-r}^{final} and u_{-r}^{final}, $\Omega = T \setminus \{-r\}$. Otherwise — $\Omega = T \setminus \{r\}$.

4. Compute partial sum for the neighbor on the y-axis

$$S' := \sum_{i\in\Omega} p_i^{final} + b(|i|-|i-t_y|)u_i^{final}$$

5. Make a swap along the y-axis and complete the partial sum S

$$S^{final} := S + S' + t_y \cdot ddy(S')$$

6. S^{final} equals $\sum_{i=-r}^{r}\sum_{j=-r}^{r}(k-b(|i|+|j|))I_{in}(x+i,y+j)$. Normalize S^{final} by the sum of the kernel.

3.3 TAA, variance clipping

The kernels for filters (7) and (8) (matrices of ones) have rank 1. Therefore, for 3x3 local area we can apply the same algorithm as for the Gaussian filter, but twice. For larger local areas this is not optimal in terms of the number of derivatives. Instead of rows interpolation, we can directly calculate partial sums for other three fragments in the constant derivatives number. This method uses nonuniform texture sampling (see Figure 7) for efficient choose of required texels (see Figure 8).

$$T = \{-r+2k \mid 0 \le k \le r\},\ Q = T \setminus \{-r\}$$

1. Read texels in nonuniform mode:

$$I_{in}(x+it_x, y+jt_y),\ i,j \in T$$

2. Compute the partial sums for the neighbour on the y-axis (see Figure 8a):

$$\mu'_y := \sum_{i\in T}\sum_{j\in Q} I_{in}(x+it_x, y+jt_y)$$

$$\sigma'_y := \sum_{i\in T}\sum_{j\in Q} I_{in}^2(x+it_x, y+jt_y)$$

3. Make a swap along the x-axis and contribute a part in diagonal neighbour's partial sums (see Figure 8b):

$$\mu'_d := \mu'_y + t_x \cdot ddx(\mu'_y) + \sum_{i\in Q}\sum_{j\in Q} I_{in}(x+it_x, y+jt_y)$$

$$\sigma'_d := \sigma'_y + t_x \cdot ddx(\sigma'_y) + \sum_{i\in Q}\sum_{j\in Q} I_{in}^2(x+it_x, y+jt_y)$$

4. Make a swap along the y-axis and contribute a part in the partial sums for the neighbour along the x-axis (see Figure 8c):

$$\mu'_x := \mu'_d + t_y \cdot ddy(\mu'_d) + \sum_{i\in Q}\sum_{j\in T} I_{in}(x+it_x, y+jt_y)$$

$$\sigma'_x := \sigma'_d + t_y \cdot ddy(\sigma'_d) + \sum_{i\in Q}\sum_{j\in T} I_{in}^2(x+it_x, y+jt_y)$$

5. Make a swap along the x-axis and complete sums (see Figure 8d):

$$\mu'_{final} := \mu'_x + t_x \cdot ddx(\mu'_x) + \sum_{i\in T}\sum_{j\in T} I_{in}(x+it_x, y+jt_y)$$

$$\sigma'_{final} := \sigma'_x + t_x \cdot ddx(\sigma'_x) + \sum_{i\in T}\sum_{j\in T} I_{in}^2(x+it_x, y+jt_y)$$

6. Complete the calculations of filters (7) and (8):

$$\mu := \frac{\mu'_{final}}{(2r+1)^2}$$

$$\sigma := \sqrt{\frac{\sigma'_{final}}{(2r+1)^2} - \mu^2}$$

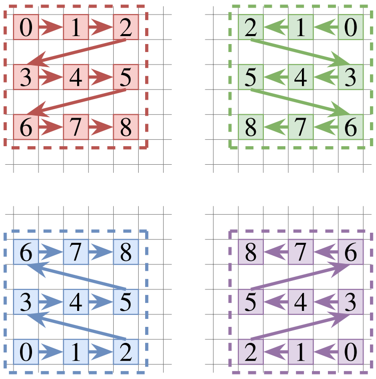
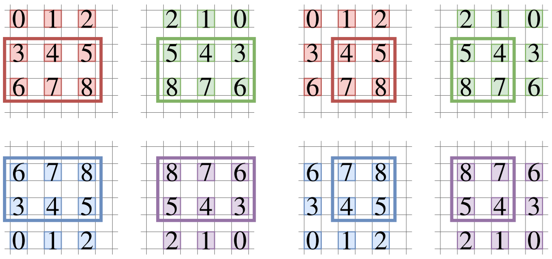
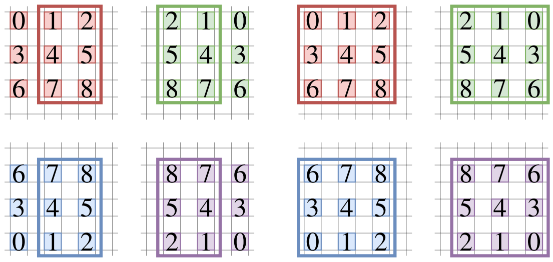


Figure 7: Nonuniform texture sampling for 2x2 group, 5x5 local area. The numbers represent the texel indices in the array. Arrows indicate the direction of sampling.



(a) For neighbour on y-axis (b) For diagonal neighbour



(c) For neighbour on x-axis (d) For fragment itself

Figure 8: Proposed method, 5x5. The rectangles in each subfigure cover the texels required for that step.

3.4 Bilateral filter

Computing the partial sums of other fragments with nonuniform sampling (see Figure 7) is the only way, since bilateral filter is nonlinear. Since its kernel (4) depends on the central texel $I_{in}(x,y)$ (each fragment has a unique central texel), we must also pass it between fragments. At the beginning, we have one of two situations:

1. The local area radius is $r = 2k$, $k \in \mathbb{N}$ and central texel $I_{in}(x,y)$ is sampled (3rd case in Figure 9).
    (a) In this case, we swap the central texel with our neighbour along the x-axis and proceed to the 4th case.
    (b) In the 4th case, we compute the partial sums for the numerator and denominator of the filter and proceed to the 1st case by swapping partial sums and central texel with the neighbour along the y-axis.
    (c) In the 1st case, we compute the partial sums for the numerator and denominator of the filter and proceed to the 2nd case by swapping partial sums and central texel with the neighbour along the x-axis.
    (d) In the 2nd case, we compute the partial sums for the numerator and denominator of the filter and proceed to the 3rd case by swapping partial sums with the neighbour along the y-axis.
    (e) In the 3rd case, we use our own central texel and complete the sums.
2. The local area radius is $r = 2k-1$, $k \in \mathbb{N}$ and central texel $I_{in}(x,y)$ is sampled by our diagonal neighbor (1st case in Figure 9)
    (a) In this case, we compute the partial sums for the numerator and denominator of the filter and proceed to the 2nd case by swapping partial sums and central texel with the neighbour along the x-axis.
    (b) In the 2nd case, we compute the partial sums for the numerator and denominator of the filter and proceed to the 3rd case by swapping partial sums with the neighbour along the y-axis.
    (c) In the 3rd case, we compute the partial sums for the numerator and denominator of the filter and proceed to the 4th case by swapping partial sums with the neighbour along the x-axis.
    (d) In the 4th case, we compute the partial sums for the numerator and denominator of the filter and proceed to the 3rd case by swapping partial sums with the neighbor along the x-axis. After that we have completed sums.

In the 1st situation at the beginning (and if $r = 1$), 6 partial derivatives are required. In the 2 situation (except $r = 1$), 7 partial derivatives are required (the numerator is a three-dimensional vector and the denominator is a scalar, so they were combined into one four-dimensional vector). Let us describe the partial sums for numerator and denominator in each case (number corresponds to the case number on Figure 9):

$$T = \{-r + 2k \mid 0 \le k \le r\},\ Q = T \setminus \{-r\}$$

1.
$$c := \sum_{i \in Q} \sum_{j \in Q} I_{in}(x + it_x, y + jt_y) g(it_x, jt_y)$$
$$w := \sum_{i \in Q} \sum_{j \in Q} g(it_x, jt_y)$$

2.
$$c := \sum_{i \in T} \sum_{j \in Q} I_{in}(x + it_x, y + jt_y) g(it_x, jt_y)$$
$$w := \sum_{i \in T} \sum_{j \in Q} g(it_x, jt_y)$$

3.
$$c := \sum_{i \in T} \sum_{j \in T} I_{in}(x + it_x, y + jt_y) g(it_x, jt_y)$$
$$w := \sum_{i \in T} \sum_{j \in T} g(it_x, jt_y)$$

4.
$$c := \sum_{i \in Q} \sum_{j \in T} I_{in}(x + it_x, y + jt_y) g(it_x, jt_y)$$
$$w := \sum_{i \in Q} \sum_{j \in T} g(it_x, jt_y)$$

where c is the partial sum of the numerator, w is the partial sum of the denominator. Swaps are performed according to the formulas (9) and (10).

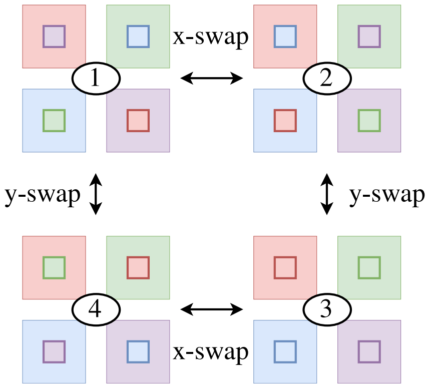


Figure 9: Four cases for the bilateral filter. The big squares are fragments. The small squares are the central texels for the corresponding fragments. From each case, you can move to one of the two neighbouring cases through swaps along the x or y axis.

4 EXPERIMENTAL RESULTS

We have used the Khronos framework [Khr19] for the Vulkan API to run our GLSL shaders on PC and Android. The comparison includes the following types of shaders (see Figures 11 and 12):

- Default version — filters are implemented in fragment shader and read $(2r+1)^2$ texels
- Compute version — filters are implemented in compute shader using shared memory. Gaussian filter also uses two-pass rendering with 128x1 workgroups. Other compute shaders use 16x16 workgroups
- Proposed version — filters are implemented in fragment shader and use derivatives
- Linear version — Gaussian filter is implemented in fragment shader and uses two-pass rendering with a linear sampler [Rak10]

Adreno 650 shows 5-100% performance degradation depending on filter and area size when using shared memory. It can be concluded that Adreno has a high cost of fine derivatives. For 3x3 filters, the performance of our method is worse by at least 56% compared to a default fragment shaders (or does not change for variance clipping). However, for large areas, our method outperforms default fragment shaders by at least 15% (except 7x7 tent filter). Also worth mentioning is the method with the linear sampler — at least 37% better performance than competitors.

RTX 2070 has relatively cheap derivatives, due to which we get a 25-50% improvement over a default fragment shader and compute shader. For this video card, our method bypasses the fragment shader with linear sampler by 42% and 24% for 3x3 and 5x5 area respectively. However, for the 7x7 region, our method is 13% slower.

It can be seen from the graphs that different GPU architectures have different costs of fine derivatives. However, as the region size increases, the proposed method shows an increasingly noticeable difference in performance, although it depends on the filter type.

4.1 Metrics and coarse derivatives

Area size	Gaussian	Tent	Bilateral	VC
3x3	1.62e-6	0	1.89e-6	1.89e-6
5x5	2.71e-6	0	2.98e-6	3.25e-6
7x7	2.98e-6	0	3.52e-6	4.34e-6

Table 1: MSE for default and proposed (with fine derivatives) methods

The proposed method with fine derivatives yields images that are almost identical to those obtained in the default method (see Tables (1) and (2)).

Area size	Gaussian	Tent	Bilateral	VC
3x3	0.999	1.0	0.999	0.999
5x5	0.999	1.0	0.999	0.999
7x7	0.999	1.0	0.999	0.999

Table 2: SSIM for default and proposed (with fine derivatives) methods

Area size	Gaussian	Tent	Bilateral	VC
3x3	3.62	4.34	14.19	685.31
5x5	1.165	1.19	14.51	122.98
7x7	0.513	0.62	21.06	29.99

Table 3: MSE for default and proposed (with coarse derivatives) methods

Area size	Gaussian	Tent	Bilateral	VC
3x3	0.991	0.99	0.988	0.503
5x5	0.995	0.995	0.992	0.854
7x7	0.997	0.997	0.984	0.966

Table 4: SSIM for default and proposed (with coarse derivatives) methods

In the early stages, testing the proposed method with coarse derivatives did not yield any noticeable performance gains for the RTX 2070. Because of this and the metrics (see Tables (3) and (4)), we abandoned the coarse derivatives in performance measurement. While for the Gaussian filter the difference is negligible, for variance clipping the method with coarse derivatives is not applicable at all (see Figure 10).

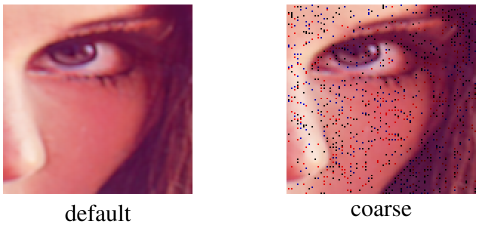


Figure 10: Mean value in variance clipping, 3x3. The default method produces correct results, while proposed method with coarse derivatives produces visual artefacts due to incorrect texels received via swaps

5 CONCLUSION

We have presented a few strategies for memory optimization in two-dimensional filters based graphics algorithms. It does not require intermediate images, gives noticeable performance gains almost everywhere, and can only be implemented by rewriting shaders. Unfortunately, explicit difference derivatives are not always cheap enough to allow blind replacement of shaders with the proposed ones. The source code of the shaders is available on GitHub: `https://github.com/kzubatov/FiltersOptimizationInfo`. Performance tests made by the community on other GPU architectures will also be available there.

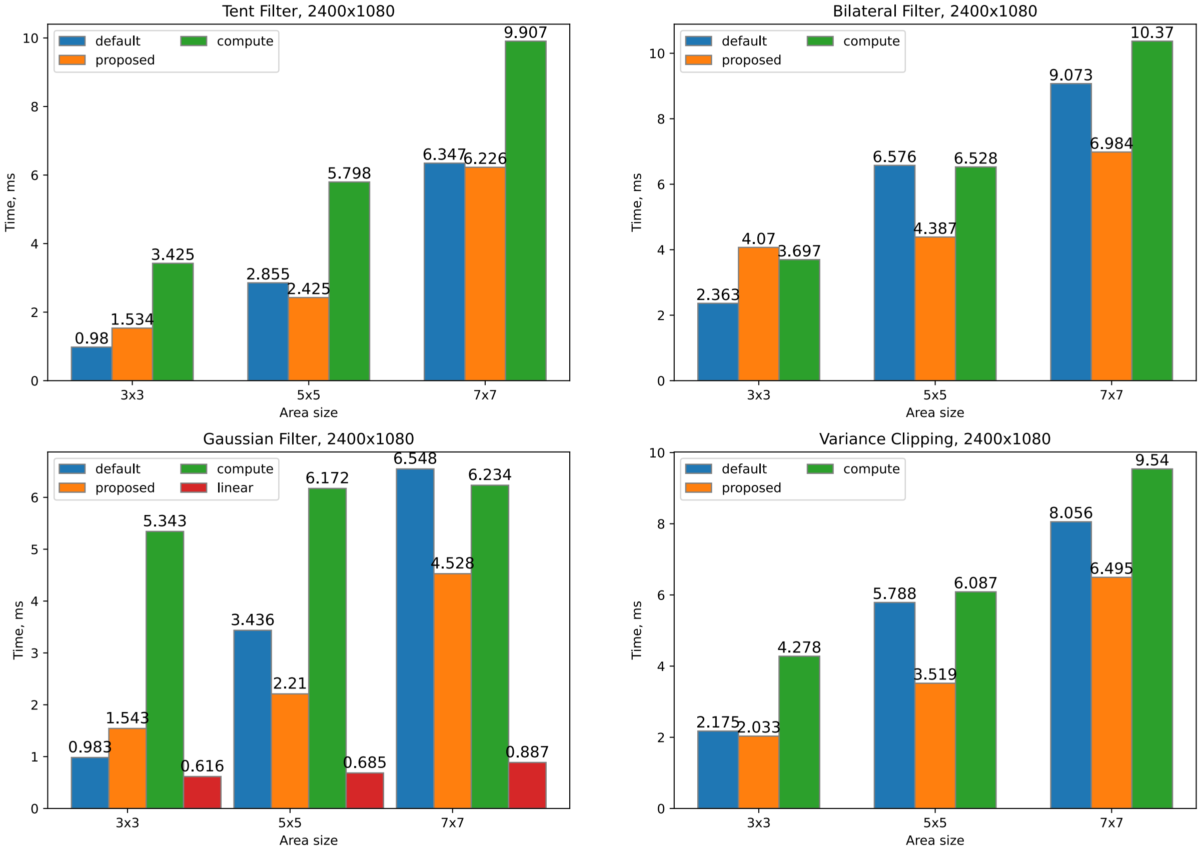


Figure 11: Performance tests, Adreno 650, Android 13

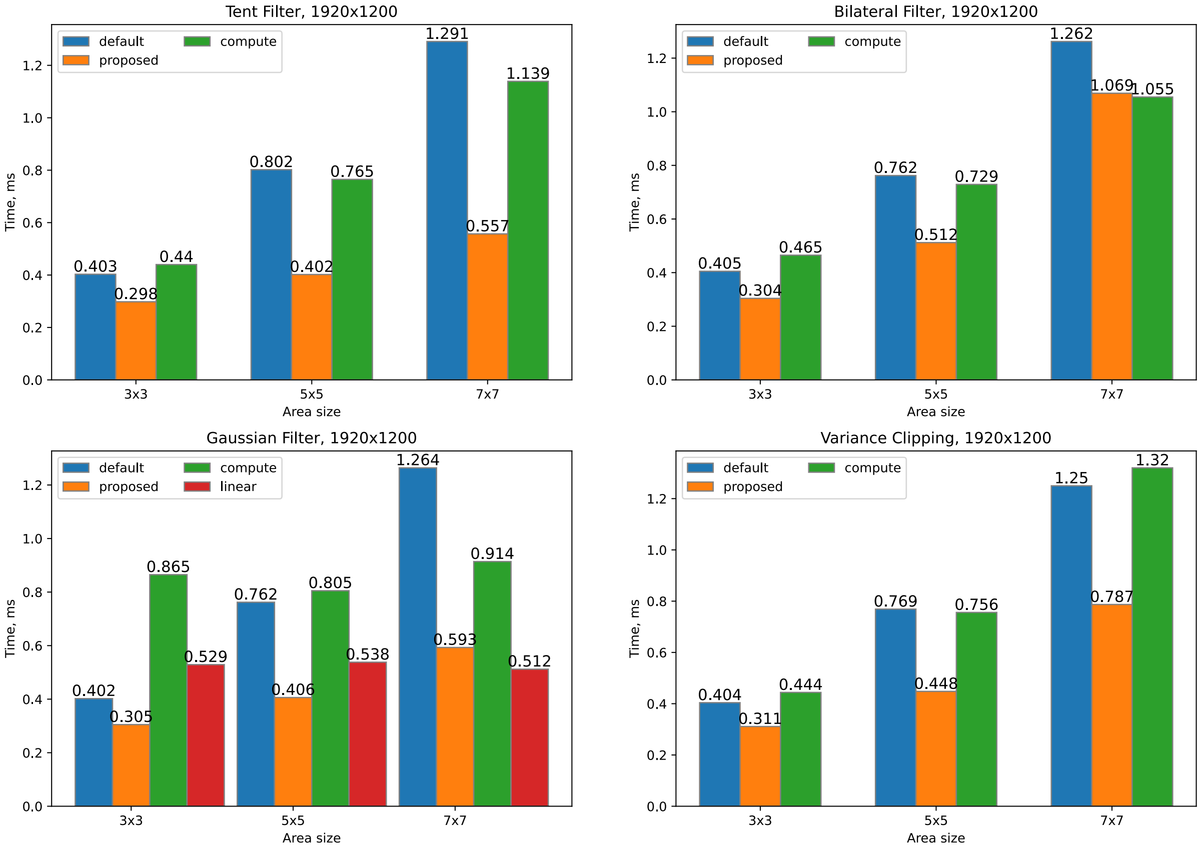


Figure 12: Performance tests, RTX 2070, Linux 5.15

6 REFERENCES

[App] Apple documentation. Blurring an image. `https://developer.apple.com/documentation/accelerate/blurring_an_image`

[Arm] Arm GPU Best Practices Developer Guide. Compute shading. Shared memory. `https://developer.arm.com/documentation/101897/0302/Compute-shading/Shared-memory`

[Bri99] Briceno, Hector, Steven Gortler, and Leonard McMillan: NAIVE - Network Aware Internet Video Encoding. Proceedings of he 7th ACM International Multimedia Conference (Part 1), 1999.

[Buc16] R.H. Buch, A.C. Calaf, P.P.V. Alcocer, "Optimized skin rendering for scanned models". In WSCG 2016: short communications proceedings: The 24th international conference in Central Europe on computer graphics, Visualization and Computer Vision 2016, pp. 89-96.

[Can86] J. Canny, "A computational approach to edge detection". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679-698, 1986.

[Don06] W. Donnelly, A. Lauritzen, "Variance shadow maps". In Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games, I3D'06, pp. 161-165, New York, 2006.

[Khr19] One stop solution for all Vulkan samples. `https://github.com/KhronosGroup/Vulkan-Samples`

[Kil12] M. Kilgard, "NVIDIA OpenGL in 2012: Version 4.3 is here!" In SIGGRAPH 2012.

[Kop07] J. Kopf, M. F. Cochen, D. Lischinski, M. Uyttendaele, "Joint bilateral upsampling". In ACM Transactions on Graphics, Volume 26, Issue3, 2007.

[Rak10] D. Rakos, "Efficient Gaussian blur with linear sampling", 2010. `https://www.rastergrid.com/blog/2010/09/efficient-gaussian-blur-with-linear-sampling`

[Sal16] M. Salvi, "An excursion in temporal supersampling". In Game Developers Conference, 2016.

[Yan09] L. Yang, D. Nehab, P. V. Sander, P. Sitthi-Amorn, J. Lawrence, H. Hoppe, "Amortized supersampling". In ACM Trans. Graph. 28, 5 (Dec. 2009), 135:1-135:12, 2009.

An automated Pipeline to bring NeRFs to the Industrial Metaverse

Sabine Schleise
SICK AG
79183 Waldkirch,
Germany
sabine.schleise@sick.de

Uwe Hahne
Hochschule Furtwangen
78120, Furtwangen,
Germany
uwe.hahne@hfu.eu

Jakob Lindinger
SICK AG
79183 Waldkirch,
Germany
jakob.lindinger@sick.de

ABSTRACT

The Industrial Metaverse offers new opportunities in various industrial sectors, such as product development, collaboration, sales and marketing. It relies on rendering 3D scenes of industrial plants, factories, and machines that are viewed by multiple human observers in various industrial applications. Creating these 3D scenes requires either elaborate 3D modeling efforts or complex 3D reconstruction methods. Both approaches require expensive hardware and highly trained individuals to create photo-realistic results. To overcome these limitations, we offer an end-to-end pipeline that generates photo-realistic 3D scene renderings from video input using NeRF (neural radiance fields). Our tool automates the entire process, resulting in remarkable efficiency gains. It reduces the creation time from days to minutes, even for untrained users.

Keywords

3D Vision, NeRF, Industrial Metaverse, immersion, neural rendering

1 INTRODUCTION

Rapid advancements in the availability of graphics processing unit (GPU) technology allow new opportunities across various industrial sectors, including training, product development, and collaboration that are often described with the term *Industrial Metaverse*. At its core, the Industrial Metaverse relies on rendering three-dimensional (3D) scenes that are viewed by multiple human observers in various industrial applications. For instance, in automation, engineers and technicians can design, test, and implement robotic automation systems. By creating virtual simulations of automated workflows, also known as digital twins, the efficiency of robots in manufacturing lines or logistics operations can be optimized by human operators via remote access. It becomes also possible to predict potential issues before they arise and plan maintenance without disrupting the production process.

Currently, these 3D scenes (as well as some digital twins) are usually rendered from explicit representations such as meshes, point clouds, or volumetric grids. Those are known for their structural integrity and compatibility with modern rendering systems. However, in 2020, a novel approach called Neural Radiance Fields (NeRF) [MST*20] emerged. It diverges from these conventional methods as it uses neural networks and volume rendering methods for novel view synthesis. It creates visually impressive results from only a few photos shot by regular consumer cameras.

NeRFs are implicit scene representations, typically employing a Multi-Layer Perceptron (MLP) that is trained by comparing volumetric renderings with captured photos. Instead of reconstructing the 3D scene geometry, NeRF generates a volumetric representation called a *radiance field*. This radiance field is stored in a neural network which returns color and density values for any given point and viewing direction within the relevant 3D space. Due to this novel representation, there is currently no existing framework to generate a neural radiance field from real world data for novice users. To address this, we introduce a user-friendly software tool called *NeRF Trainer* that simplifies the creation of neural radiance fields. Our tool provides an end-to-end solution that streamlines the training of NeRFs. It automates the entire process including image selection, extraction of camera orientation using COLMAP [SZPF16, SF16], training of NeRF models and export of resulting new images, videos and 3D meshes. By leveraging the improved NeRF algorithm *Instant-NGP* [MESK22] our tool achieves short training times of a mere 20 seconds. We aim to democratize the use of NeRF technology, making it accessible to a broader audience without the barriers of technical

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.

https://www.doi.org/10.24132/CSRN.3401.17

expertise or hardware limitations. We believe that this innovation represents a significant step forward in realizing the potential of NeRF in industrial settings and beyond.

2 BACKGROUND

Before we delve into the steps required to create NeRFs in the upcoming sections, we present fundamental findings below.

2.1 NeRF

NeRF is a method for synthesizing novel views of complex scenes by optimizing a continuous volumetric function using a sparse set of 2D images. Given a set of images showing a static scene from multiple view angles, along with their corresponding camera poses, the neural network learns to represent that scene in a way that allows new views to be synthesized. The NeRF algorithm represents a continuous scene as a vector-valued function F_Θ whose input is a 3D spatial location $\mathbf{x} = (x, y, z)$ and a 2D viewing direction $\mathbf{d} = (\phi, \theta)$. Its output is the emitted color $\mathbf{c} = (r, g, b)$ and a volume density σ, that ranges from $[0, \infty)$. To approximate this continuous 5D scene representation, an MLP network F_Θ is utilized, where $F_\Theta : (\mathbf{x}, \mathbf{d}) \longrightarrow (\mathbf{c}, \sigma)$. The weights Θ of the network are optimized to map each $5D$ input vector to its corresponding volume density and directional emitted color. The algorithm enables view synthesis through a three-step process: first, sampling 5D coordinates along camera rays and processing them through an MLP to produce both color and volume density; second, using the MLP's output to generate individual pixels of an image using classical volume rendering techniques; and finally, leveraging the differentiability of the rendering function to optimize the MLP's weights by minimizing the loss function defined as the total squared error between the rendered and true pixel colors.

The NeRF algorithm encounters difficulties when rendering scenes with high-frequency variations in color and geometry due to its direct operation on 5D input coordinates. This inefficiency is caused by the inherent bias of deep neural networks towards learning lower-frequency functions, a limitation that Rahaman et al. [RBA*19] have highlighted. To address this, NeRF uses a positional encoding that leverages sine and cosine functions to improve its representation of geometry. However, this approach can complicate training and affect GPU performance due to the high cost of complex control flow and memory operations.

These challenges are addressed by Müller et al. in "Instant Neural Graphics Primitives" (Instant-NGP) [MESK22] which introduces a novel input encoding method that significantly streamlines the training process. By employing a smaller network integrated with a multi-resolution hash table, this method reduces computational demands and optimizes memory usage. The result is a speed-up of several orders of magnitude in training time for neural graphics components, enabling high-quality renderings to be trained in seconds, even on consumer hardware. This breakthrough demonstrates the potential for rapid, efficient training of neural scene representations, as illustrated in Figure 1, which shows a scene reaching convergence in training within just 20 seconds. The scene was captured from a 180-degree perspective, utilizing 145 frames for training, and trained using a laptop with an NVIDIA A3000 GPU.

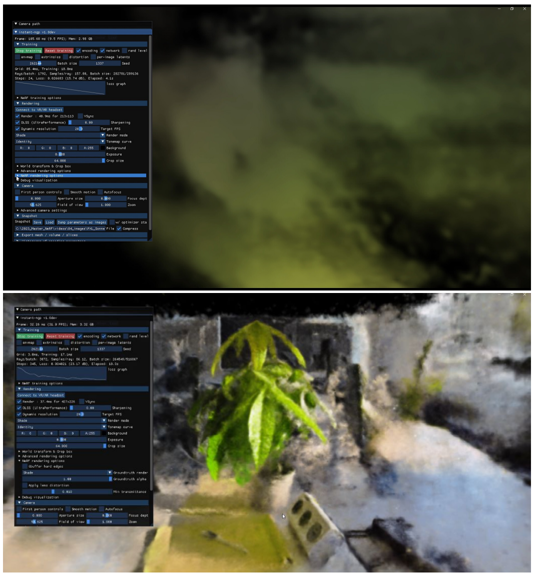


Figure 1: Training of a neural radiance field using the Instant-NGP algorithm, displayed in the Instant-NGP GUI [MESK22]. Rendering at initialization (top) and after 20 seconds of training (bottom). The object was captured from a 180-degree perspective, utilizing 145 images for training.

2.2 Manual pipeline

After laying the conceptual groundwork for NeRF and its advances in handling high-frequency scene variations, we delve into the practical methodology for training a NeRF model, focusing on the seamless integration of theory and application.

The pipeline typically starts with capturing a short video of a scene or object with a smartphone. This is followed by frame extraction, which is the process of assembling a set of images that capture the static scene from different angles. These images, whether extracted from video sequences or collected as single images, form the backbone of the dataset, capturing the spatial intricacies and visual essence of the scene from multiple perspectives.

Once the frames have been extracted, the focus shifts to determining the camera positions for each frame in the dataset. This is achieved by using COLMAP [SF16], a structure-from-motion (SfM) software, which reconstructs the scene in 3D by identifying common points across the image set and inferring the camera positions and orientations from which each image was captured. This process lays the foundation for a spatial understanding of the scene, allowing the NeRF algorithm to accurately interpret and reconstruct the 3D environment.

With the frames extracted and the camera poses computed, the dataset is prepared for the next phase: training the NeRF model. The model, represented by a multi-layer perceptron (MLP), is tasked with learning a continuous volumetric scene function. It takes as input the 3D coordinates and 2D viewing directions, and outputs the color and volume density for each point in space. Training involves feeding the model with 5D coordinates derived from the camera rays intersecting the scene, and adjusting the network weights to minimize the discrepancy between the synthesized and actual image pixels. This optimization process uses gradient descent to iteratively refine the model to improve its ability to render novel views with high fidelity.

3 RELATED WORK

The impact of NeRF on the computer vision community has been profound and widespread. Since its publication in 2020 numerous papers and preprints emerging on platforms, reflecting the intense interest and rapid development within this area. We refer to the recent survey papers [GGH*22, TTM*22] to get an impression about the broad evolution of NeRF-based methods in the last years.

Recognizing its significance, developers have created frameworks that addressed a variety of needs, from simplifying the integration of different NeRF techniques to adapting complex algorithms for broader use. For instance, the introduction of frameworks with modular architectures [YL22, HLY*20, XC22, YSK22] have made it easier for users to work with various NeRF models within a single application. This integration supports experimentation and refinement of NeRF models, benefiting the research community. In this frameworks, new methods can be easily integrated without having to rewrite encodings, network architectures or volume rendering functions. This is particularly useful as the field of NeRF is constantly evolving, with new methods and approaches emerging regularly. The framework NVIDIA Kaolin Wisp [TPT*22] was designed as a dynamic, research-focused library for neural fields to help researchers address the challenges of this discipline. Notably, efforts have transitioned from reimplementing existing NeRF methods in different programming environments [Bha22, YC20] to offering comprehensive solutions such as Nerfstudio [TWN*23] that span the entire spectrum of NeRF technology, from data preprocessing to model training and visualization.

Despite NeRF technology's growing adoption across various fields, there are significant challenges with the current frameworks that hinder its broader accessibility, especially for non-technical users. The support for development remains basic, with a proliferation of research papers and a lack of code consolidation making it difficult to track advancements. Many researchers publish their work in isolated repositories, complicating the transfer of features and contributions between different NeRF implementations. Furthermore, the existing tools for running NeRFs on real-world data are limited. In particular, the data preparation step involving COLMAP [SF16] requires command line knowledge, which is a barrier for those unfamiliar with such interfaces. In addition, the plethora of features and NeRF models offered by most frameworks can overwhelm users new to the technology, making it difficult for them to understand its capabilities and effectiveness. These issues underscore the need for more intuitive and consolidated frameworks that can meet the needs of users who want to explore NeRF technology without deep technical expertise.

A notable advancement in this field is the startup *Luma AI*[1], which has significantly simplified the process of creating NeRFs. By releasing an iPhone app, *Luma AI* enables users to effortlessly produce videos that can then be transformed into a NeRF. The app guides users through the entire process, from capturing video to generating the NeRF, thereby making this advanced technology accessible to a wider audience. In addition, companies like *Dromni*[2] are also working to democratize the process of generating NeRFs by developing platforms that are easy to use. Nonetheless, a significant barrier to their widespread adoption in sensitive sectors is the necessity for data upload to cloud services. This is particularly critical in various industrial domains where safeguarding data confidentiality is essential.

4 NERF TRAINER

In this chapter, we present a comprehensive exploration of our innovative NeRF Trainer, a tool designed to democratize the creation of NeRFs by simplifying its process into a user-friendly application.

4.1 Requirements and design choices

Our framework is designed with specific functional requirements to facilitate the generation and exploration of neural radiance fields from video inputs. In the following, we provide a breakdown of these key requirements.

---

[1] https://lumalabs.ai

[2] https://www.veovid.com

4.1.1 Automated Pipeline

From a user perspective, the main feature of our framework is its capability to process video uploads. It automatically executes the entire pipeline described in Section 2.2 to generate the radiance field. This process involves converting the video into a series of images that capture the scene from multiple angles, then run COLMAP to extract the camera calibration and finally train the radiance field to construct a 3D representation of the scene. The video processing capability is designed with flexibility in mind, allowing the user to specify optional parameters to fine-tune the output. Through this automated pipeline, our technology simplifies the traditionally complex process of generating NeRF models. It allows users to obtain high-quality 3D representations from standard video footage, bypassing the need for manual image selection and camera calibration.

4.1.2 Advanced Algorithm Utilization

The framework's effectiveness hinges on incorporating an advanced, improved radiance field algorithm. The selected algorithm must strike a delicate balance between rapid processing and the rendering of scenes with high fidelity. The ultimate goal is to achieve a reconstruction that faithfully mirrors the real-world scene's complexity and subtleties, ensuring an authentic and immersive exploration experience.

4.1.3 Enhanced Output Options

Beyond the generation of neural radiance fields, the framework offers advanced functionalities for rendering videos and exporting meshes. This broadens the scope of exploration and utilization of the constructed 3D scene representations.

Snapshots: Save model snapshots as .ingp file for analysis and further exploration within the Instant-NGP's framework.

Video Rendering: Users have the option to render videos directly from the synthesized neural radiance fields, enabling dynamic visualization of the scene from various perspectives. This feature is instrumental in presenting and reviewing the photorealistic outputs in a more interactive and engaging format.

Mesh Exportation: The technology also supports the exportation of meshes. This feature allows for seamless integration into existing workflows, providing users with the tools to incorporate 3D models into their projects.

4.1.4 Graphical User Interface (GUI)

To ensure accessibility and ease of use, the application features a GUI. This interface will support informative interactions, data uploads, and parameter configurations, tailored to optimize training outcomes. The GUI is envisioned as a bridge between the user and the complex processes underlying neural radiance field generation, making the technology approachable for users with varying levels of expertise.

4.1.5 Screenshots as Training Verification

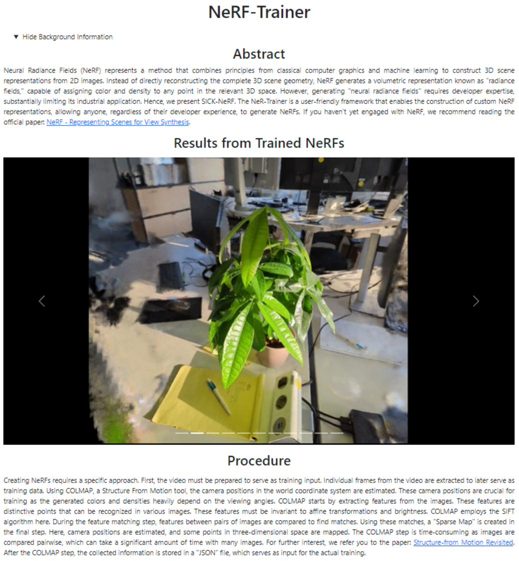
A key requirement is the ability to produce photorealistic renderings of the trained model's output. These renderings are essential for verifying the model's accuracy in recreating the scene. They act as a proof of concept, demonstrating the model's ability to replicate the real-world environment with precision and detail.

4.2 The framework

In the pursuit of democratizing advanced 3D visualization technologies, we introduce a web-based platform designed to simplify the training of NeRFs. This platform is tailored for both novice and experienced users aiming to generate high-fidelity 3D scenes from conventional images or videos. The user interface is crafted to guide users through the data upload, parameter adjustment, and visualization stages with ease, resulting in the generation of an .ingp file compatible with the Instant-NGP framework [MESK22] for immersive exploration. The back-end, built in Python, can run on any server equipped with an NVIDIA RTX GPU, eliminating the need for a local graphics card. This combination of a user-friendly interface and a separate back-end makes it simpler for users to create high-quality neural radiance fields. As illustrated in Figure 2, the front-end start screen provides the user with insights into NeRFs, results from previously trained models, and the platform's capabilities. This introductory segment aims to acquaint users with the potential and applications of NeRF technology.

4.3 Workflow

The workflow begins when users upload scene data to the platform, followed by an opportunity to modify parameters before the pipeline process begins. If users choose not to make any adjustments, the pipeline will use the default parameters by default. These defaults are designed to provide a balance between performance and quality, ensuring that the NeRF model is trained efficiently and effectively under general conditions. The pipeline process is schematically depicted in Figure 3. Our pipeline is designed to process video, images and RGB-D data. The prepocessing steps for these formats differ: When processing video data, we start by extracting individual frames. Next, camera parameters are determined using COLMAP [SF16]. Once these preliminary steps have been completed, the derived data is transformed into JSON format. This JSON-formatted data, together with the corresponding images, forms the training data set for the neural radiance field.



NeRF-Trainer

▼ Hide Background Information

Abstract

Neural Radiance Fields (NeRF) represents a method that combines principles from classical computer graphics and machine learning to construct 3D scene representations from 2D images. Instead of directly reconstructing the complete 3D scene geometry, NeRF generates a volumetric representation known as "radiance fields," capable of assigning color and density to any point in the relevant 3D space. However, generating "neural radiance fields" requires developer expertise, substantially limiting its industrial application. Hence, we present SICK-NeRF. The NeR-Trainer is a user-friendly framework that enables the construction of custom NeRF representations, allowing anyone, regardless of their developer experience, to generate NeRFs. If you haven't yet engaged with NeRF, we recommend reading the official paper: NeRF - Representing Scenes for View Synthesis.

Results from Trained NeRFs

Procedure

Creating NeRFs requires a specific approach. First, the video must be prepared to serve as training input. Individual frames from the video are extracted to later serve as training data. Using COLMAP, a Structure From Motion tool, the camera positions in the world coordinate system are estimated. These camera positions are crucial for training as the generated colors and densities heavily depend on the viewing angles. COLMAP starts by extracting features from the images. These features are distinctive points that can be recognized in various images. These features must be invariant to affine transformations and brightness. COLMAP employs the SIFT algorithm here. During the feature matching step, features between pairs of images are compared to find matches. Using these matches, a "Sparse Map" is created in the final step. Here, camera positions are estimated, and some points in three-dimensional space are mapped. The COLMAP step is time-consuming as images are compared pairwise, which can take a significant amount of time with many images. For further interest, we refer you to the paper: Structure-from Motion Revisited. After the COLMAP step, the collected information is stored in a "JSON" file, which serves as input for the actual training.

Figure 2: Rendered view of the background information section on the website. This section includes an introduction to what functionalities the website offers, followed by examples of previously trained NeRF models for qualitative evaluation, and concludes with an outline of the NeRF training pipeline process.

For iPhone users, especially the newer models such as iPhone 12 and its successors, we offer an alternative option to create a NeRF from RGB-D data. The Record3D[3] application allows the creation of recordings including the camera poses and parameters. This data is then read from the file and can be converted directly into the required JSON format, eliminating the need for camera calibration via COLMAP. This eases the transition to the training phase and makes the whole process more efficient.

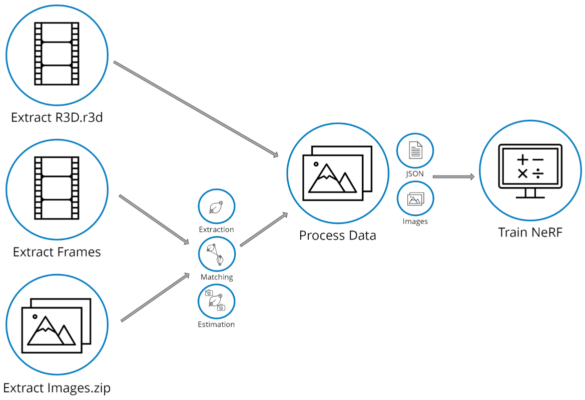


Figure 3: Comprehensive overview of our automated pipeline: The diagram delineates the distinct phases involved when utilizing either video or R3D files (the output format from the iPhone App *Record3D*).

Following the comprehensive overview of our automated pipeline, we transition to a detailed examination of each constituent step.

4.3.1 *Extract Videos*

This stage involves extracting frames from the video, a process critical for preparing the input data for subsequent NeRF model training. The *Extract Videos* component is designed with flexibility in mind, allowing users to customize the extraction process based on specific requirements. This customization includes the ability to adjust the frequency of frame extraction - such as selecting every second or third frame - and setting a maximum limit on the number of frames to be extracted. This flexibility is particularly beneficial for managing computational resources and avoiding memory access errors, which can occur when processing large video files or when operating within the constraints of limited GPU memory. This tailored approach to frame extraction ensures that the pipeline can accommodate videos of varying lengths and complexities, making it possible to generate NeRFs from a wide range of source materials without compromising on the quality or completeness of the data.

4.3.2 *Estimate camera poses from images*

Integrating COLMAP [SF16] for estimating camera poses in the process of creating NeRFs is a step that enables the successful translation of 2D images into their 3D counterparts. It is important to accurately position each frame extracted from the video within a 3D context to achieve precise 3D scene reconstruction, which is a crucial task for NeRF. The accuracy of camera pose estimation directly affects the NeRF model's ability to render new, unseen views of the scene that maintain the original photometric properties. Our framework automatically incorporates COLMAP to simplify the user experience and keep the focus on the NeRF process. This abstraction of technical intricacies allows users to engage with NeRF creation without needing to delve into complex technical details.

4.3.3 *Training Process*

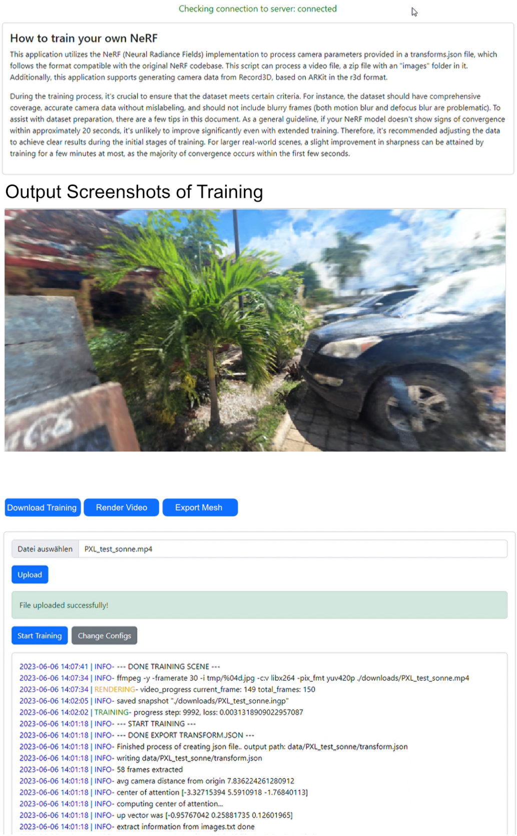
For the training phase, the COLMAP data is converted to a JSON format, encapsulating both intrinsic and extrinsic camera parameters for each image. The neural radiance field is then trained using the Instant-NGP algorithm [MESK22] over a predefined, adjustable number of steps. Upon completing the training, screenshots from the trained model are rendered to evaluate its performance and visual output.

4.3.4 *Qualitative Evaluation*

Upon completion of the training process, the quality of the generated NeRF model is evaluated based on its

---

[3] https://record3d.app/



Checking connection to server: connected

How to train your own NeRF

This application utilizes the NeRF (Neural Radiance Fields) implementation to process camera parameters provided in a transforms.json file, which follows the format compatible with the original NeRF codebase. This script can process a video file, a zip file with an "images" folder in it. Additionally, this application supports generating camera data from Record3D, based on ARKit in the r3d format.

During the training process, it's crucial to ensure that the dataset meets certain criteria. For instance, the dataset should have comprehensive coverage, accurate camera data without mislabeling, and should not include blurry frames (both motion blur and defocus blur are problematic). To assist with dataset preparation, there are a few tips in this document. As a general guideline, if your NeRF model doesn't show signs of convergence within approximately 20 seconds, it's unlikely to improve significantly even with extended training. Therefore, it's recommended adjusting the data to achieve clear results during the initial stages of training. For larger real-world scenes, a slight improvement in sharpness can be attained by training for a few minutes at most, as the majority of convergence occurs within the first few seconds.

Output Screenshots of Training

Datei auswählen PXL_test_sonne.mp4

File uploaded successfully!

```
2023-06-06 14:07:41 | INFO- --- DONE TRAINING SCENE ---
2023-06-06 14:07:34 | INFO- ffmpeg -y -framerate 30 -i tmp/%04d.jpg -c:v libx264 -pix_fmt yuv420p ./downloads/PXL_test_sonne.mp4
2023-06-06 14:07:34 | RENDERING- video_progress current_frame: 149 total_frames: 150
2023-06-06 14:02:05 | INFO- saved snapshot "./downloads/PXL_test_sonne.ingp"
2023-06-06 14:02:02 | TRAINING- progress step: 9992, loss: 0.0031318909022957087
2023-06-06 14:01:18 | INFO- --- START TRAINING ---
2023-06-06 14:01:18 | INFO- --- DONE EXPORT TRANSFORM.JSON ---
2023-06-06 14:01:18 | INFO- Finished process of creating json file.. output path: data/PXL_test_sonne/transform.json
2023-06-06 14:01:18 | INFO- writing data/PXL_test_sonne/transform.json
2023-06-06 14:01:18 | INFO- 58 frames extracted
2023-06-06 14:01:18 | INFO- avg camera distance from origin 7.836224261280912
2023-06-06 14:01:18 | INFO- center of attention [-3.32715394 5.5910918 -1.76840113]
2023-06-06 14:01:18 | INFO- computing center of attention...
2023-06-06 14:01:18 | INFO- up vector was [-0.95767042 0.25881735 0.12601965]
2023-06-06 14:01:18 | INFO- extract information from images.txt done
```

Figure 4: The user interface on the webpage after a radiance field training, encompassing back-end logs (cropped).

ability to produce photorealistic renderings of the scene from novel viewpoints. Therefore we render screenshots from new viewpoints that were not part of the initial dataset used for training. By generating these images, we provide tangible evidence of the model's ability to synthesize new perspectives with high fidelity.

Figure 4 showcases the user interface of the website upon the completion of a NeRF training session. It features a verification screenshot of the trained radiance field, alongside interactive buttons that allow the user to download the resulting data, render a video or export the trained model as a mesh. Additionally, the interface provides logs that detail the training process, offering users insights into the progression and outcomes of their session. Through a socket connection, users receive ongoing updates ensuring that they remain updated about the training's progression and outcome. Such transparent feedback and live interactions elevate the user experience, making the entire training journey more intuitive. This feature is particularly valuable during the execution of computational-intensive processes, such as the COLMAP phase. A specialized logger continually updates the user on the current state of the pipeline, providing an informed overview of the ongoing activities.

4.4 Implementation aspects

To develop and operationalize our NeRF framework, we addressed the challenges outlined in the previous sections, including the fragmented development landscape of NeRF technologies and the obstacles faced by non-technical users. Our approach has led to the creation of a robust, user-friendly framework, underpinned by two Docker containers designed for seamless communication - one dedicated to the front-end and the other to the back-end. This modular architecture not only simplifies deployment but also ensures data privacy by enabling on-premises hosting within a company's network, leveraging Kubernetes clusters for scalability and GitLab CI/CD for continuous integration and deployment. The use of GitLab CI/CD underscores our commitment to maintaining a robust, up-to-date application capable of meeting the evolving needs of our users.

Poetry is employed as a dependency management and packaging tool, which simplifies the process of managing python packages and their versions, ensuring a consistent development environment. Furthermore, Docker is utilized to containerize the back-end, allowing it to run on any server equipped with an NVIDIA RTX GPU. This eliminates the need for users to possess local graphics cards and reduces the setup complexity, offering a plug-and-play solution. By leveraging Docker, we encapsulate the NeRF application within a controlled environment, ensuring that it can be deployed and run with ease, irrespective of the underlying operating system or hardware configuration.

The splitting of the framework into a back-end, developed in Python, and a front-end, crafted with React, allows us to offer a powerful yet user-friendly tool. The React-based content management system (CMS) enables users to interact with the NeRF application without needing to navigate the command line or understand the back-end processes.

To enhance the user experience further, we employ a YAML-based configuration file, making it straightforward for users to interact with the application. YAML's human-readable format ensures that configuring the application is accessible to all, regardless of technical background. This choice underscores our commitment to making NeRF technology as user-friendly as possible.

The strategic division between front-end and back-end, combined with our deployment strategy, empowers organizations to integrate our framework into their networks seamlessly. This paves the way for a diverse range of users, including those without technical expertise, to create content for Industrial Metaverse applications. Our framework's architecture is designed not just for accessibility but also for scalability, making it an ideal solution for businesses looking to explore and expand their capabilities within the Metaverse.

5 NERF IN THE INDUSTRIAL METAVERSE

The introduction of NeRF marked the birth of the domain of Neural Scene Rendering, a subfield within computer graphics focused on generating 3D scenes using neural networks [TTM*22]. In general, neural scene rendering is a rapidly growing field with the potential to revolutionize the way we generate 3D content, offering data-driven approaches that make the process more accessible and efficient than traditional methods.

NeRF represents a significant step forward in the realm of 3D content creation, offering substantial benefits to the growth of the Industrial Metaverse. Beyond its technical achievements, what sets NeRF apart is its potential to make high-quality 3D rendering more accessible and user-friendly. This accessibility paves the way for a wider range of applications, from enhancing product development processes to enriching educational content. By simplifying the creation of detailed, realistic scenes, NeRF democratizes 3D modeling, making it possible for a broader audience to contribute to and benefit from advancements in virtual environments.

Figure 5 showcases a series of images rendered from neural radiance fields that are generated with our NeRF Trainer. These NeRFs underwent a rapid training process, completed within just one minute, demonstrating the efficiency and capability of the utilized algorithms to generate high-fidelity 3D scenes. The use of advanced optimization techniques and computing power has enabled NeRFs to efficiently process and interpret large amounts of visual data, surpassing traditional methods like photogrammetry.

Additionally, these neural radiance fields can be explored in Virtual Reality (VR) using the Instant-NGP GUI [MESK22], offering an immersive experience that has numerous advantages in the Industrial Metaverse context. The integration of VR technology marks a significant advancement in industrial operations and planning. Through VR (e.g. the production facility displayed in Figure 6), stakeholders can now explore and interact with a fully immersive 3D model of their production environments. This capability not only enhances the understanding of spatial relationships and operational flows within the facility but also facilitates a more intuitive and effective collaboration among team members. Whether for training purposes, safety protocol reviews, or optimization of manufacturing processes, the ability to virtually walk through and examine every aspect of a production line in detail offers unparalleled benefits. These include improved design accuracy, faster decision-making, and a reduction in costly physical prototypes.

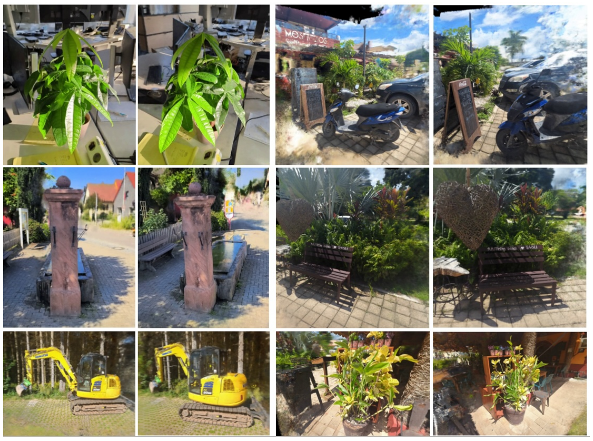
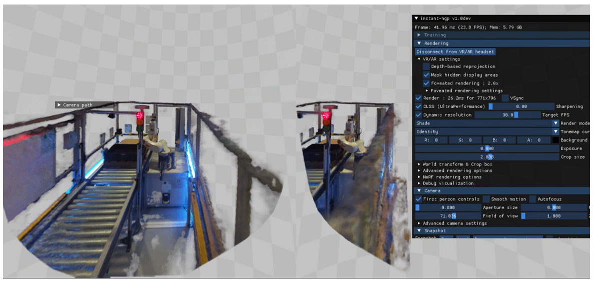


Figure 5: Rendered images from NeRFs trained within 1 minute, showcasing impressive detail and realism despite the constrained training period. This demonstrates the advancements in optimization techniques and computing power, enabling rapid processing and interpretation of visual data.


Camera path
▼ instant-ngp v1.0dev
▶ Training
▼ Rendering
Disconnect from VR/AR headset
▼ VR/AR settings
▶ Foveated rendering settings
▶ World transform & Crop box
▶ Advanced rendering options
▶ NeRF rendering options
▶ Debug visualization
▼ Camera
▶ Advanced camera settings
▼ Snapshot

Frame: 41.96 ms (23.8 FPS); Mem: 5.79 GB
☐ Depth-based reprojection
☑ Mask hidden display areas
☑ Foveated rendering : 2.0x
☑ Render : 26.2ms for 771x796 ☐ VSync
☑ DLSS (UltraPerformance) 0.00 Sharpening
☑ Dynamic resolution 30.0 Target FPS
Shade Render mode
Identity Tonemap cur
R: 0 G: 0 B: 0 A: 0 Background
0.000 Exposure
2.038 Crop size
☑ First person controls ☐ Smooth motion ☐ Autofocus
0.000 Aperture size 0.000
71.036 Field of view 1.000

Figure 6: A screenshot showing the dual perspectives as viewed through VR lenses.

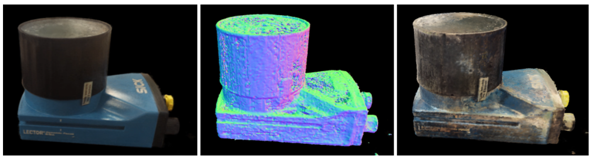


Figure 7: Comparison of renderings and meshes obtained from a NeRF. The first image shows a rendering from the neural radiance field, the second image shows the mesh with vertex normals, and the third image displays the mesh with vertex colors.

Beyond the capability to visualize NeRFs in VR, the method also allows for the option to export these 3D models as meshes, utilizing the Marching Cubes [LC87] algorithm. This process translates the dense, volumetric data produced by NeRFs into a polygonal mesh, which is more compatible with traditional 3D modeling and CAD software. In Figure 7, we provide a visual comparison highlighting the transition from NeRF to an exported mesh using the Marching Cubes algorithm. However, it is important to note that NeRFs, by their nature, are not ideally suited for direct mesh exportation and use in certain applications. NeRFs are remarkable for their ability to capture and display complex light interactions within a scene and create photorealistic images from a variety of angles. When converted into a mesh, this lighting information and the ability to seamlessly render from arbitrary viewpoints will be lost. Yet, this limitation is contrasted by the significant advantage NeRFs offer in quickly and easily capturing detailed 3D representations of scenes using a smartphone.

6 DISCUSSION

Our framework aims to reduce some of the cumbersome manual steps required for generating NeRFs. However, it comes with its own set of minor limitations: When used as intended, our framework needs, for every group of users (e.g. within a company), a maintainer who is skilled in deploying the front- and backend services. All other users then require internet (or corporate network) access, which is not the case for most other NeRF tools (but ubiquituous nowadays). When used locally (which is not intended but possible), our framework assumes a familiarity of the user with command line tools similar to other NeRF frameworks. Further limitations are inherited from relying on existing NeRF technology. We discuss this in the following, especially when considering NeRF's application within the dynamic and demanding realms of the Industrial Metaverse.

NeRFs have demonstrated remarkable proficiency in rendering static scenes, showcasing an ability to capture intricate details with high fidelity. The performance of NeRFs in representing static scenes is impressive, but standard NeRFs reach their limits when it comes to dynamic environments. In such scenarios, objects may move or change, requiring time-dependent reconstruction beyond the capabilities of the original NeRFs and their improvements. There is much ongoing research focused on filling this gap, particularly through the use of MLP-based methods. A detailed overview of these methods can be found in Section 4.3 of the article "Advances in Neural Rendering" [TTM*22]. By incorporating dynamic aspects, NeRFs could be used in a wider range of applications, from real-time entertainment to complex simulations in industry. The NeRF algorithm also lacks integrated physics for simulating the behavior of models in industrial settings. Without this, NeRF remains a somewhat superficial tool for visual representation but not for behavioral simulation.

Another feature that traditional (digital) 3D tools offer is the possibility to easily modify the scene, e.g. add/modify objects or change certain colors. For NeRFs this is inherently complicated caused by the significant computational intensity required due to the dense radiance field representation. Tools like NeRFShop and NeuralEditor are pioneering solutions to this issue, showcasing the potential for further advancements [JKK*23, CLW23].

Moreover, the lack of standardization and fragmentation within NeRF developments presents hurdles to widespread adoption. Current models' specificity to particular tasks complicates the creation of a unified platform. Additionally, the opaque nature of neural networks underlying NeRFs poses challenges for fine-tuning and integration across diverse models. However, new approaches such as Gaussian Splatting are emerging, leveraging classical machine learning techniques to generate new views more efficiently [KKLD23].

The integration of NeRF with widely used game engines and virtual environments, including Unity, Omniverse, and Unreal Engine, as well as the process of converting NeRF's output into usable meshes, remains a complex challenge. These integration issues highlight the necessity for ongoing research and development efforts aimed at enhancing NeRF's compatibility and functionality across different digital creation platforms.

Despite these limitations, the potential for reality capture through neural radiance fields remains significant. The ongoing research in this domain points to the development of increasingly efficient and versatile methods. Adaptations that integrate dynamic scenarios, real-time interactivity, and physics-based simulation are not beyond reach, and several works in the current literature are already advancing in these directions. The fast-paced growth of research in this area indicates that many of the challenges presented in this section could be overcome in the near future.

7 CONCLUSION

In this work, we have developed a framework to make training NeRFs from user content more accessible. We overcome key challenges by providing an easy to use and system-independent setup. Our approach makes NeRF accessible to a broader audience, including non-developers in various industries, democratizing this advanced 3D rendering technology.

An important next step is to validate this claim in applied user studies. We envision a comparison of the applicability of different NeRF creation tools (e.g. ours but also research code on GitHub as well as commercially available solutions) in different scenarios. At the same time, user studies should be conducted to evaluate preferences for different audiences.

Finally, by enhancing NeRF's approachability, we open up industrial applications like virtual prototyping and immersive training, transforming virtual engagement and fostering innovation. Looking ahead, integrating NeRF with the Industrial Metaverse promises to merge physical and digital worlds. It allows powerful visualization and interaction tools that could revolutionize industrial operations.

8 REFERENCES

[Bha22] BHALGAT Y.: Hashnerf-pytorch. `https://github.com/yashbhalgat/HashNeRF-pytorch/`, 2022.

[CLW23] CHEN J.-K., LYU J., WANG Y.-X.: NeuralEditor: Editing Neural Radiance Fields via Manipulating Point Clouds. In *CVPR* (2023).

[GGH*22] GAO K., GAO Y., HE H., LU D., XU L., LI J.: NeRF: Neural Radiance Field in 3D Vision, A Comprehensive Review, Nov. 2022. arXiv:2210.00379 [cs].

[HLY*20] HU S.-M., LIANG D., YANG G.-Y., YANG G.-W., ZHOU W.-Y.: Jittor: a novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences 63*, 222103 (2020), 1–21.

[JKK*23] JAMBON C., KERBL B., KOPANAS G., DIOLATZIS S., LEIMKÜHLER T., DRETTAKIS G.: NeRFshop: Interactive Editing of Neural Radiance Fields". *Proceedings of the ACM on Computer Graphics and Interactive Techniques 6*, 1 (5 2023).

[KKLD23] KERBL B., KOPANAS G., LEIMKÜHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics 42*, 4 (2023).

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics 21*, 4 (1987), 163–169.

[MESK22] MÜLLER T., EVANS A., SCHIED C., KELLER A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph. 41*, 4 (July 2022), 102:1–102:15.

[MST*20] MILDENHALL B., SRINIVASAN P., TANCIK M., BARRON J., RAMAMOORTHI R., NG R.: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, 2020. arXiv:2003.08934 [cs].

[RBA*19] RAHAMAN N., BARATIN A., ARPIT D., DRAXLER F., LIN M., HAMPRECHT F. A., BENGIO Y., COURVILLE A.: On the spectral bias of neural networks, 2019.

[SF16] SCHÖNBERGER J. L., FRAHM J.-M.: Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[SZPF16] SCHÖNBERGER J. L., ZHENG E., POLLEFEYS M., FRAHM J.-M.: Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)* (2016).

[TCY*22] TANCIK M., CASSER V., YAN X., PRADHAN S., MILDENHALL B., SRINIVASAN P. P., BARRON J. T., KRETZSCHMAR H.: Block-NeRF: Scalable Large Scene Neural View Synthesis, 2022.

[TPT*22] TAKIKAWA T., PEREL O., TSANG C. F., LOOP C., LITALIEN J., TREMBLAY J., FIDLER S., SHUGRINA M.: Kaolin Wisp: A PyTorch Library and Engine for Neural Fields Research. `https://github.com/NVIDIAGameWorks/kaolin-wisp`, 2022.

[TTM*22] TEWARI A., THIES J., MILDENHALL B., SRINIVASAN P., TRETSCHK E., YIFAN W., LASSNER C., SITZMANN V., MARTIN-BRUALLA R., LOMBARDI S., SIMON T., THEOBALT C., NIESSNER M., BARRON J. T., WETZSTEIN G., ZOLLHOEFER M., GOLYANIK V.: Advances in neural rendering. *Computer Graphics Forum 41*, 2 (2022), 703–735.

[TWN*23] TANCIK M., WEBER E., NG E., LI R., YI B., WANG T., KRISTOFFERSEN A., AUSTIN J., SALAHI K., AHUJA A., MCALLISTER D., KERR J., KANAZAWA A.: Nerfstudio: A Modular Framework for Neural Radiance Field Development. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings* (2023), ACM.

[XC22] XRNERF-CONTRIBUTORS: Openxr-lab neural radiance field toolbox and benchmark. `https://github.com/openxrlab/xrnerf`, 2022.

[YC20] YEN-CHEN L.: Nerf-pytorch. `https://github.com/yenchenlin/nerf-pytorch/`, 2020.

[YL22] YUE LUO Y.-P. C.: arcnerf: nerf-based object/scene rendering and extraction framework, 2022.

[YSK22] YOONWOO J., SEUNGJOO S., KIBAEK P.: Kakaobrain/nerf-factory: An awesome pytorch nerf library, 2022.

[ZSD*21] ZHANG X., SRINIVASAN P. P., DENG B., DEBEVEC P., FREEMAN W. T., BARRON J. T.: NeRFactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics 40*, 6 (2021), 1–18.

Segmentation of discrete surfaces into plane segments based on a distance map

Loic Drieu La Rochelle[1]
loic.drieu.la.rochelle@univ-poitiers.fr

Rita Zrour[1]
rita.zrour@univ-poitiers.fr

Gaëlle Largeteau-Skapin[1]
gaelle.largeteau.skapin@univ-poitiers.fr

Eric Andres[1]
eric.andres@univ-poitiers.fr

Olena Tankyevych[2 3]
Olena.Tankyevych@chu-poitiers.fr

Catherine Cheze Le Rest[2 3]
Catherine.Cheze-Le-Rest@chu-poitiers.fr

[1]Université de Poitiers, Univ. Limoges, CNRS, XLIM, Poitiers, France
[2]Centre Hospitalier Universitaire de Poitiers (CHUP), Poitiers, France.
[3]LaTIM, INSERM, UMR 1101, Univ Brest, Brest, France.

ABSTRACT

In this paper, we present a method for segmenting 3D discrete objects into discrete plane segments. This segmentation is the first step in obtaining a polyhedrization of a discrete object with the reversibility property. This constraint requires that the discretization result for polyhedrization be exactly the same as the initial set of points. One of our objectives is to reduce the number of planes in our segmentation and achieve more efficient surface analysis algorithms. In 3D space, direction and starting point are common issues. Our method attempts to achieve segmentation by considering surfels one after the other and agglomerating them with their neighbours based on a distance ranking. This method enables the recognition of critical points on the boundary of a plane segment. A medical application is illustrated by the presentation of a tumour segmentation.

Keywords
Polyhedrization - Segmentation - Discrete Geometry - Discrete Plane - Medical Imaging.

1 INTRODUCTION

Discrete surface polyhedrization has received increasing interest in discrete geometry and has been studied in different papers [BB06, CDJS06, LMR20, LRC22, SC03, SDC04]. The reconstruction of a discrete 3D surface is a challenging problem, especially when a reduced number of polygons is expected and even more, when a property of reversibility is required. Reconstruction is reversible when the digitization of the reconstructed surface corresponds to the original discrete surface. The reversibility property ensures that no information is created or lost during the reconstruction process which is an important property in sensitive applications such as medical imaging. A reconstruction with fewer polygons helps for fluid and interactive visualization in medical imaging where the volume of data can be important. It represents a bottleneck in medical virtual reality: in [PCD21], the authors noted that for Virtual Reality *"[...] results suggest that prerequisites such as real-time performance [...] pose the greatest limitations for clinical adoption and need to be addressed"*.

One of the widely used methods for 3D reconstruction is the Marching Cubes isosurface extraction algorithm [LC87] that proposes reversible solutions, however, it generates an important number of triangles that is proportional to the number of voxels or surface elements (surfels) of the discrete surface. In the discrete geometry community, several algorithms were proposed for discrete surface polyhedrization. These approaches are usually decomposed into two steps: segmentation of a discrete surface into plane segments [BF94, DA09, DCA06, KS01, Pap99, SDC04, VC00], and reconstruction where each of the plane segments is

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.

https://www.doi.org/10.24132/CSRN.3401.18

replaced by one or more continuous polygons [CDJS06, CGS04, DA09, DCA06]. Segmentation into discrete planes is an important step since it can guarantee the reversibility property and potentially diminish the number of polygons in the resulting reconstruction. This paper is focused on this first step. There is however a major obstacle to this segmentation process: a segmentation in a minimal number of plane segments has been shown to be an NP-hard problem [SC09].

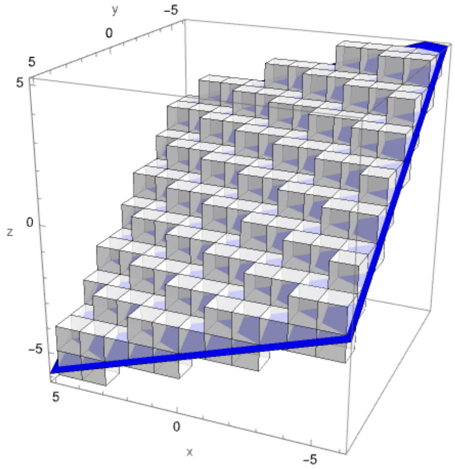


Figure 1: The standard discretization of a Euclidian plane (in blue) of equation $x+2y+3z=1$

In this work, we seek a solution that would yield fewer discrete plane segments in most cases than previously proposed methods [CDJS06, CGS04, DGT]. Our method considers surface elements (surfels) instead of voxels for the reconstruction. This means that we are going to work with *standard* plane segments (i.e. 6-connected planes segments) [And03]. The Figure 1 shows what a Standard plane looks like and how surfels are connected (here face to face). When a segmentation is computed, the starting point and the direction of traversal has an important impact on the results. While, in 2D, on curves, only two directions are possible, in 3D, each surfel has 4 neighbours and selecting a specific direction introduces a bias in the segmentation process. We propose thus labelling on the surfels of our surface, which depends on the planarity of the surfel by determining the biggest disk centred on the surfel that would belong to the same plane. We deduce from that a queue of plane segment centres. Additionally, we adapted the notion of k-cuspal cells proposed by M. Rodriguez et al [RLA08] to propose a geometric criterion for a somewhat subjective question that has not been considered often so far: some reconstruction lead to shapes that *may feel more natural* than others (see Figure 2 for an example).

This paper is organized as follows: section 2 states some definitions and notations. Section 3 presents a state of the problem and of the art on some existing segmentation and reconstruction methods. Section 4 details our segmentation method and section 5 shows some results. We finally end with some conclusions and perspectives in section 6.

2 DEFINITIONS AND NOTATIONS

In this section, we review some notions and definitions of discrete geometry that will be used in the following sections. A definition of a discrete plane, for all points $(x,y,z) \in \mathbb{Z}^3$, has been given by JP. Reveilles in [Rev91] as follow :

$$-\frac{\omega}{2} \leq ax+by+cz+\mu < \frac{\omega}{2}$$

with ω the thickness of the plane, the *intersect* μ and (a,b,c) the normal of the plane. There are classically two types of discrete planes that are considered in discrete surface segmentation problems: *naive* discrete planes are the thinnest 18-connected planes without 6-connected tunnels that can be analytically characterized by a thickness $\omega = \max(|a|,|b|,|c|)$ [AAS97] ; *Standard* discrete planes are the thinnest 6-connected planes without tunnels that can be analytically characterized by a thickness $\omega = |a|+|b|+|c|$ [And03, AAS97]. In the literature (see next section), two discrete surface paradigms have been considered: a *voxel* and a *surfel* paradigm. For the voxel paradigm, the discrete surface of a three-dimensional object is considered to be constituted by the 18-connected voxel outer layer of the object. Those voxels are then segmented into naive planes segments. A discrete plane segment is a finite set of connected points that belong to a discrete plane. In the surfel paradigm, each voxel is considered to be a cubical complex where the 2-dimensional elements are called *surfel*, the 1-dimensional elements are called *linel* and the 0-dimensional elements *pointel*. The surface of an object is composed of the outer layer of surfels of the object (all the surfels that belong to a voxel of the object and to the complementary), and the associated linels and pointels that form the boundary of the surfels. Those surfels are then segmented into standard planes where each pointel of the surfels is seen as a discrete point of the discrete plane segments.

3 STATE OF THE PROBLEM AND STATE OF THE ART

3.1 State of the problem

In dimension two, the segmentation of a curve (open or closed) into discrete line segments has been solved with a solution that is both linear in time and optimal in the number of line segments [FT05]. However, in dimension three, decomposing a discrete surface into a minimal number of discrete planes has been shown to be NP-hard [SC09]. This is not completely surprising since there is no natural order in which discrete points can be added in a seed-based approach (which is the most common approach as we will discuss next) in dimension three. Sometimes, there is no better solution than a trivial one for an object such as the one presented on the left of Figure 2. All the optimal decompositions into discrete planes will have a number of plane segments that are proportional to the number of voxels/surfels. You cannot do anything here that would be better than a simple straightforward local segmentation. This means that in some cases, globally or locally, one cannot expect a solution that is not proportional to the number of voxels/surfels of the surface. The minimality in the decomposition is not the only relevant question. One aspect of the decomposition of discrete surfaces that has not often been taken into account is that some decomposition *feel* less natural than others. Both the middle and the left image of Figure 2 present an optimal segmentation of a surfel cube into six planes, however, the middle image segmentation may be considered more natural/expected.

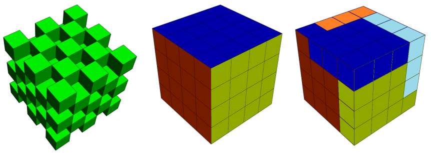


Figure 2: Left: limit configuration. Centre: desired optimal decomposition. Right: undesired optimal decomposition.

3.2 State of the art

Let us now discuss some of the previously proposed approaches to surface segmentation. In his thesis, L. Papier [FP99, Pap99], first proposed a method that creates only segments of naive planes that are rectangular (with a rectangular projection). In a second method, with standard discrete plane segments, he used an algorithm based on Fourier-Motzkin elimination. He imposed geometric and topologic constraints for the plane segments that are commonly taken into account now: connected faces that are homeomorphic to disks. The authors discussed a number of problems with this line of methods such as how to choose the starting point for a face and how to select adjacent surfels to a face and in which order.

In [KS01], R. Klette et al. proposed a segmentation algorithm based on the resolution of a system of inequalities. Surfels are processed by a breadth first traversal of the graph of surfel adjacency on the surface. However, no constraints on the recognized plane segment were imposed which led to thin and awkwardly shaped segments.

In her thesis [Siv04], I. Sivignon proposed several segmentation algorithms. She proposed to use the notion of tricubes (planar sets of 3×3 voxels) to reduce the number of small plane segments that can be recognized. She also considered plane segments that are topological disks which resulted in a smaller number of plane segments than previous approaches. Lastly, a side effect of using tricubes is to make configurations such as the central image of Figure 2 impossible to reconstruct.

In [CGS04], authors proposed a polyhedrization algorithm that has the property of reversibility with the warranty that the obtained polyhedron is topologically correct; it is based on the simplification of the Marching Cube surface. This algorithm has then been extended in [CDJS06] using linear programming techniques to reduce the number of plane segments.

A last approach [LMR20] uses an arithmetical approach to incrementally compute the normal of a plane segment. Although promising, this approach has not yet resulted in a plane segmentation algorithm.

4 OUR METHOD

In this section, we present our segmentation method to decompose a discrete surface into discrete planes segments.

For the recognition algorithm, we are using an incremental algorithm (COBA) that consists in adding discrete points one after another [CB08]. The recognition problem in $\mathbb{Z}^3$ is transformed into a feasibility problem in $\mathbb{Z}^2$. The function corresponds to the parameter of the two parallel planes enclosing points, and the diagonal distance between these two planes. The time complexity of this algorithm in the worst case is $O(n\log(n))$. It is thus an efficient algorithm for discrete plane recognition. This algorithm does not provide the *preimage* of the set of discrete points but gives the two supporting planes of the segment. In this work, we recognize standard discrete planes using the pointels (in the boundary of surface surfels) of our object.

We work with face-connected voxel objects. From this set of voxels, we compute a surfel-adjacency graph of the surface of our object. The discrete surface that we obtain is closed, thus each surfel has exactly four neighbours.

4.1 Context

Firstly, we need to retrieve the surface of a discrete object composed of voxels to obtain an unoriented graph of surfels. If the surface is closed, each vertex of the graph has exactly four neighbours, and we limit our work to this type of object (i.e. manifold). For the recognition of discrete plane segments, we use the Standard model to have standard surfaces, edges and vertices [And03].

We recognize standard discrete planes considering the four pointels of each surfel of our graph as the discrete points of these planes. A surfel can be added to only one

https://www.doi.org/10.24132/CSRN.3401.18

plane segment, thus the boundaries between our different plane segments are formed of lignels and pointels.

The method implemented in our algorithm consists in two passes over our set of surfels. The first is used to compute surfel weights by computing an approximation of the size of the maximal plane it can belong to (see section 4.2). The second pass then computes the segmentation based on the weights while taking care of special cases (see section 4.3).

4.2 First pass

In this subsection, we present the way we compute the weights. The weight of each surfel corresponds to the radius of the biggest Manhattan disk centred on the surfel that corresponds to a standard plane segment. It is computed with a Breadth-First Search order, with a method akin to classical distance map computation algorithms.

A surfel priority queue sorted by decreasing surfel weights is maintained throughout the process.

Figure 3 presents the weight of the surfels on a cube, an octahedron, and a sphere with a colour gradient from green to red. Red surfels represent surfels where it is impossible to create a large plane segment and dark green are those where a larger plane segment can be created. We can see that some *edges* are recognizable in red. Red points are akin to 3D discrete cusp introduced by M. Rodriguez [RLA08]. In 2D, a point is a discrete cusp of a discrete curve iff a line segment of length five centred on this point is not a discrete segment [BSDA03]. M. Rodriguez [RLA08] generalized this notion to dimension three.

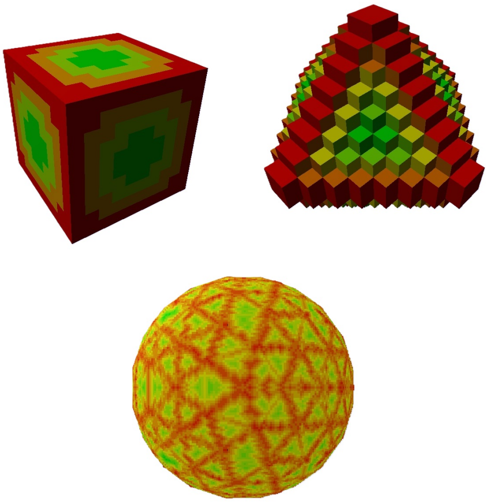


Figure 3: The weighted surfels of our surface on: a cube, an octahedron, a digitized mesh of a sphere

4.3 Second pass

In this second pass, we take the surfels one by one in the surfel priority queue (surfels with the biggest weights first). The four direct neighbours of the surfel are tested and depending on the configuration (see Figures 4 and 5), different type decisions are taken. These different configurations can be described as follows:

- First configurations: The current surfel has no neighbour already inside a discrete plane segment where it can be added: either because the neighbours have not yet been treated (Figure 4a) or because the current surfel cannot be added to any of the already existing planes (as they wouldn't be planes anymore. See Figure 4b). For these configurations, a new plane segment is created.
- Second configuration: The current surfel has one and only one discrete plane segment already constructed in its neighbourhood where it can be added. The surfel is simply added to this plane (Figure 4c).
- Third configuration: The current surfel has more than one discrete plane segment in its neighbourhood to which it can be added. To select the best plane segment to add the surfel to, we choose to compare the normals of the different plane segments. We calculate a scalar product between the normals before and after adding the surfel to each plane and retain the solution that modifies the normal the least. The union of the point sets of two discrete plane segments can also be a discrete plane segment. Once the segment on which the surfel is to be added has been selected, we then check whether other plane segments in the neighbourhood can be merged with the selected segment. This configuration potentially reduces the number of recognized planes at this step (Figure 4).

Figure 4a and 4b describe two different cases related to the first configuration. In Figure 4a, the yellow surfel is the one with the biggest weight when compared to its fours neighbours; it cannot be added to an existing plane since no plane exists yet in his neighbourhood. In Figure 4b, the yellow surfel cannot be added to the already existing red plane since the red plane is no more a standard plane when adding this surfel. In these two cases for configuration 1, a new plane segment is created that can then aggregate other surfels.

Figure 4c shows an example of the second configuration. The yellow surfel has one of its neighbours in the green plane it can be added to. This plane is the only current plane segment in its neighbourhood, it will thus be added to this plane.

Figure 5a shows an example of the third configuration where the yellow surfel can be added to two different plane segments in its neighbourhood, the green plane and the blue plane. In this case, the green and the blue plane segments along with the yellow surfel can be merged. This is what happens in this case. Figure 5b shows an example of the third configuration where the yellow surfel can be added to two different plane segments in its neighbourhood (green and blue) that cannot be merged into one plane segment. In this case, the yellow surfel will be added to the green plane as it doesn't modify the normal of the green plane, while it would modify the normal of the blue plane.

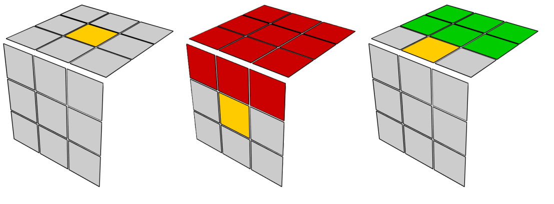

(a) (b) (c)

Figure 4: The two first configurations. (a) first configuration: no neighbours have already been treated, (b) first configuration: the current surfel (yellow) cannot be added to the red plane segment, (c) second configuration when a surfel has exactly one neighbouring plane to which it can be added.

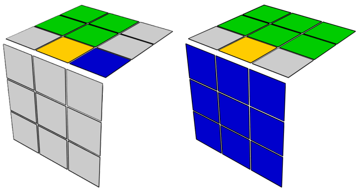

(a) (b)

Figure 5: The third configuration when the surfel can be added to at least two plane segments. (a) Merging the planes is possible and will be done. (b) Merging is not possible and the normal is used to decide in which plane segment the surfel is added (here the green one).

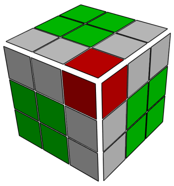


Figure 6: An example of the particular case where a surfel of distance 1 (in red) should not be computed before the grey ones also of distance 1.

There is however one special case to consider: it occurs when a surfel and all its neighbours of a surfel have a weight equal to 1. In Figure 6, the red surfels could form an autonomous plane, although each of those surfels can be added to other planes that may be created later on. The idea here is to wait before such surfels are handled. In this example, the red surfels will be treated after the grey ones, an then end up being added to three different planes. To avoid this case from happening, the weight of such surfels can be diminished to let neighbour surfel (grey in Figure 5) be computed first.

Algorithm 1: Second pass

```
input  : The $queue$ resulting from the first pass
         and $G$ the graph of adjacency of surfels
output: a segmentation of the set of surfels into a
         discrete plane segment
$segmentation \leftarrow \{\}$
while $queue \neq \{\}$ do
    $surfel \leftarrow queue[0]$
    $queue \leftarrow queue \setminus surfel$
    $potential \leftarrow GOOD\_NEIGHBORS(surfel)$
    if $size(potential) = 0$ then
        Create a new discrete plane segment $P$
        $segmentation \leftarrow segmentation \cup P$
        $P \leftarrow P \cup surfel$
    else
        if $size(potential) = 1$ then
            $potential[0] \leftarrow potential[0] \cup surfel$
        else
            foreach $P$ in $potential$ do
                $P\_temp \leftarrow P \cup surfel$
                $pre\_n \leftarrow$ normal of $P$
                $post\_n \leftarrow$ normal of $P\_temp$
                $scalar \leftarrow pre\_n \cdot post\_n$
            $B \leftarrow P$ with the minimal $scalar$
            $B \leftarrow B \cup surfel$
            foreach $Pi$ in $potential \setminus B$ do
                Try to merge $Pi$ with $B$
```

Algorithm 1 shows the second pass of our algorithm. The *GOOD_NEIGHBORS* function retrieves all the direct plane segments in the neighbourhood of the current surfel.

5 RESULT

This section shows the results of applying our algorithm on different 3D surfaces. The first three synthetic objects denoted *Cube*, *Octahedron* and *Sphere Mesh* are all obtained from the digitization of a mesh; the last object denoted *Parametric Sphere* is a parametric sphere voxelized using Gaussian digitization. These four objects are described as follows:

- *Cube*: a cube (6 face, 8 vertices) of size $10 \times 10 \times 10$ voxels.
- *Octahedron*: an octahedron (8 triangle, 6 vertices) of height 16 voxels.
- *Sphere Mesh*: a low polly sphere (320 triangles, 162 vertices) of radius 64 voxels.
- *Parametric Sphere*: a sphere with a radius of 64 voxels.

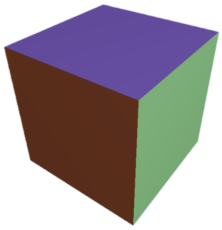
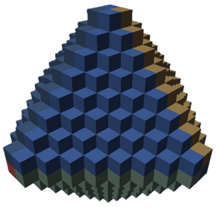
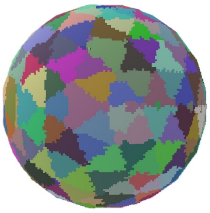
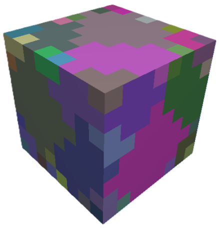
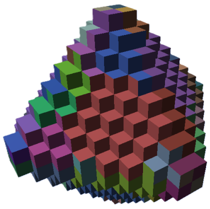
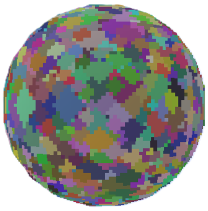
Algorithm	Cube	Octahedron	Sphere Mesh
Our Algorithm			
DGtal Algorithm			

Table 1: The segmentation of the surface of a cube; an octahedron; a digitized sphere mesh using our algorithm and using the DGtal library algorithm.

Table 1 and Figure 7 show the results of applying our algorithm and the algorithm presented in DGtal library, on the *Cube*, *Octahedron*, *Sphere Mesh* and *Parametric Sphere*. Table 2, highlights some details about these results. In this table, we discuss, for each object, the number of faces (triangles or polygons) in the mesh before digitization (Nb Faces) and the number of surfels after digitization (Nb Surfels).

We compare our algorithm to the segmentation method present in DGtal (DGTal is an open-source library where state-of-the-art algorithms in digital geometry are integrated) [DGT]. As this method allows the recognition of thick planes, we limit ourselves to standard planes, in order to maintain a criterion of reversibility. In addition, in order to maintain homogeneity with our method, each surfel is treated as a set of 4 pointels.

The results obtained using this algorithm are listed in Table 2. The results are compared on several metrics: the number of plane segments (Nb PS) generated by each segmentation method, the number of surfels inside the biggest (PS Max Size) and smallest plane segments (PS Min Size), the average (PS Mean Size) and median size (PS Median Size) of a plane segment. We also take the number of plane segments smaller than a defined size, here segments smaller than three and smaller than five in order to see if a method tends to produce small plane segments. These metrics are present in the literature and can be retrieved in several publications [Siv04].

For the *Sphere Mesh*, the number of segments recognized is greater than the number of triangles present in the initial mesh which is what we expected to retrieve. Furthermore, a lot of little segments are present in the segmentation, the presence of these little segments in the segmentation is a well-known problem and appears at the end of the algorithm when segments are strongly constrained and new surfels cannot fit the actual discrete plane recognition. However, $98,7\%$ of the surfels are in the 320 biggest plane segments. A post-processing stage could reduce those cases, as discussed in the perspectives.

For the *Parametric Sphere* (see Figure7) we have a greater number of plane segments for a roughly similar number of surfels. This result was to be expected since the parametric sphere is not derived from a mesh and has no plane in its geometry. However, results are competitive with previously proposed methods even if some surfels are embedded in plane segments. The percentage of small segments (less than 5 surfels) for the *Parametic Sphere* (28%) is just above the one for the *Mesh Sphere* (24%).

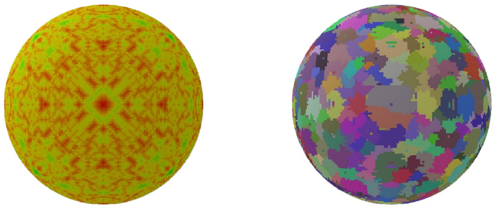


Figure 7: Our algorithm on a parametric sphere (left: first pass, right: second pass).

Overall, our method generates fewer plane segments than the method proposed in the DGtal library, particularly in terms of small pieces of planes.

As noted in section 1, the reconstruction of 3D surfaces can be really useful in medical imaging. Figure 8 shows a head-and-neck tumour semi-automatically segmented from PET image with the algorithm FLAB [HLRT+09] and Figure 9 shows our segmentation and the recon-

https://www.doi.org/10.24132/CSRN.3401.18

Object	Cube		Octahedron		Sphere Mesh		Parametric Sphere	
Nb Faces	6		8		320			
Nb Surfels	600		864		77 270		77 118	
Algorithm	Our	DGtal	Our	DGtal	Our	DGtal	Our	DGtal
Nb PS	6	85	8	81	456	2233	1183	2552
PS Max Size	100	41	108	45	551	354	356	159
PS Min Size	100	1	108	1	1	1	1	1
PS Mean Size	100	7	108	10	169	34	65	30
PS Median Size	100	8	108	9	203	31	52	39
Nb surfels ≤ 3	0	24	0	5	83	313	276	370
Nb surfels ≤ 5	0	39	0	29	100	427	335	554

Table 2: Quantitative information obtained after applying our algorithm and the algorithm presented in DGtal.

struction of the tumour. From the clinical point of view, the tumor shape is important in cancer prognosis and therapy choice analysis [HLRA$^+$21], therefore it is crucial to have its realistic 3D representation, which provides more precise shape descriptors. Such shape information combined with other image-based characteristics (often called "radiomics"), such as intensity and texture features, can be used to create prediction models for therapeutic choices and for survival prediction via machine and deep learning approaches [TTL$^+$22, HLRA$^+$21, BTT$^+$20].

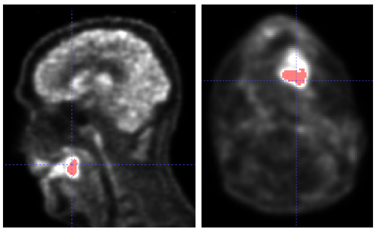


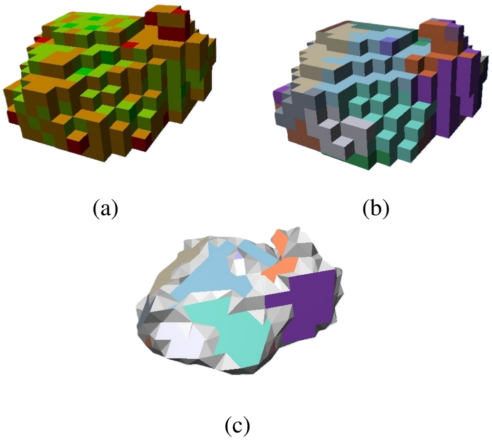
Figure 8: A head-and-neck tumour semi-automatically segmented from PET image with the algorithm FLAB (left : sagittal plane, right : axial plane).

6 CONCLUSION

In this paper, we have presented a new segmentation method for 3D digitized objects. Our method uses critical point detection based on surfel analyses based on a distance map. It retrieves fewer plane segments that are better distributed on the object than the state-of-the-art methods [FP99, KS01, Pap99, Siv04].

Surfel analysis however needs to be improved: some surfels having very different neighborhoods are labelled in the same way which is not accurate. We want to further analyse the topological configuration of surfel neighbourhoods and adjust the weight computation accordingly to better discriminate between cases.

Moreover, we have to study the criteria used to decide, when a surfel can be added to more than one plane and which one is the "best". We may, for those particular surfels, use a post-treatment to swap them from one plane segment to another to optimize the reconstruction. A way to reach our goal may be to define a Machine learning model to optimize our results.


(a) (b)

(c)

Figure 9: Polyhedrization and our algorithm surface decomposition on a head-and-neck cancer tumour. (a) The first pass, (b) the segmentation on discrete plane segment, (c) a reconstruction using Marching cube optimisation algorithm [CGS04] on our segmentation (b).

Other post-treatments can be considered: cutting the large plane segments to add smaller segments to homogenize the size of the polygons for example.

7 REFERENCES

[AAS97] E. Andres, R. Acharya, and C. Sibata. Discrete analytical hyperplanes. *Graphical Models and Image Processing*, 59(5):302–309, 1997.

[And03] E. Andres. Discrete linear objects in dimension n: the standard model. *Graphical Models*, 65(1):92 – 111, 2003.

[BB06] Valentin E. Brimkov and Reneta Barneva. Polyhedrization of discrete convex volumes. In *Advances in Visual Computing*, pages 548–557, 2006.

[BF94] P. Borianne and J. Françon. Reversible polyhedrization of discrete volumes. In *4th Discrete Geometry for Computer Imagery*, pages 157–168, Grenoble, France, 1994.

[BSDA03] Rodolphe Breton, Isabelle Sivignon, Florent Dupont, and Eric Andres. Towards an invertible euclidean reconstruction of a discrete object. In Ingela Nyström, Gabriella Sanniti di Baja, and Stina Svensson, editors, *Discrete Geometry for Computer Imagery*, pages 246–256, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[BTT$^+$20] Moran Berraho, Olena Tankyevych, Gaelle Tachon, HATT Mathieu, Dimitris Visvikis, and Catherine Cheze Le Rest. Use of pet derived features to predict mutational status in lung adenocarcinomas, 2020.

[CB08] Emilie Charrier and Lilian Buzer. An Efficient and Quasi Linear Worst-Case Time Algorithm for Digital Plane Recognition. In *Discrete Geometry for Computer Imagery*, pages 346–357, 2008.

[CDJS06] David Coeurjolly, Florent Dupont, Laurent Jospin, and Isabelle Sivignon. Optimization Schemes for the Reversible Discrete Volume Polyhedrization Using Marching Cubes Simplification. In *Discrete Geometry for Computer Imagery*, pages 413–424, 2006.

[CGS04] David Coeurjolly, Alexis Guillaume, and Isabelle Sivignon. Reversible discrete volume polyhedrization using Marching Cubes simplification. In *Vision Geometry XII*, pages 1 – 11. International Society for Optics and Photonics, SPIE, 2004.

[DA09] Martine Dexet and Eric Andres. A generalized preimage for the digital analytical hyperplane recognition. *Discret. Appl. Math.*, pages 476–489, 2009.

[DCA06] Martine Dexet, David Coeurjolly, and Eric Andres. Invertible polygonalization of 3d planar digital curves and application to volume data reconstruction. In *Advances in Visual Computing, Second International Symposium*, pages 514–523, 2006.

[DGT] DGTAL. Digital geometry tools and algorithms library. `http://libdgtal.org`.

[FP99] Jean Françon and Laurent Papier. Polyhedrization of the Boundary of a Voxel Object. In *Discrete Geometry for Computer Imagery*, pages 425–434, Berlin, Heidelberg, 1999.

[FT05] F. Feschet and L. Tougne. On the min DSS problem of closed discrete curves. *Discrete Applied Mathematics*, pages 138–153, 2005.

[HLRA$^+$21] Mathieu Hatt, Catherine Cheze Le Rest, Nils Antonorsi, Florent Tixier, Olena Tankyevych, Vincent Jaouen, Francois Lucia, Vincent Bourbonne, Ulrike Schick, Bogdan Badic, et al. Radiomics in pet/ct: current status and future ai-based evolutions. *Seminars in Nuclear Medicine*, 51(2):126–133, 2021.

[HLRT$^+$09] Mathieu Hatt, Catherine Cheze Le Rest, Alexandre Turzo, Christian Roux, and Dimitris Visvikis. A fuzzy locally adaptive bayesian segmentation approach for volume determination in pet. *IEEE transactions on medical imaging*, 28(6):881–893, 2009.

[KS01] Reinhard Klette and Hao Jie Sun. Digital Planar Segment Based Polyhedrization for Surface Area Estimation. In *Visual Form 2001*, pages 356–366, Berlin, Heidelberg, 2001.

[LC87] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, pages 163–169, 1987.

[LMR20] Jacques-Olivier Lachaud, Jocelyn Meyron, and Tristan Roussillon. An optimized framework for plane-probing algorithms. *J. Math. Imaging Vis.*, 62(5):718–736, 2020.

[LRC22] Jui-Ting Lu, Tristan Roussillon, and David Coeurjolly. A new lattice-based plane-probing algorithm. In *Discrete Geometry and Mathematical Morphology*, pages 366–381, 2022.

[Pap99] Laurent Papier. *Polyédrisation et visualisation d'objets discrets tridimensionnels*. PhD thesis, Louis Pasteur, Strasbourg, 1999.

[PCD21] F. Pires, C. Costa, and P. Dias. On the use of virtual reality for medical imaging visualization. *J Digit Imaging*, page 1034–1048, 2021.

[Rev91] Jean-Pierre Reveillès. *Géométrie discrète, calcul en nombres entiers et algorithmique*. PhD thesis, Université Louis Paster, 1991.

[RLA08] Marc Rodríguez, Gaëlle Largeteau-Skapin, and Eric Andres. Local non-planarity of three dimensional surfaces for an invertible reconstruction: k-cuspal cells. In *Advances in Visual Computing, 4th International Symposium, ISVC*, Lecture Notes in Computer Science, pages 925–934, 2008.

[SC03] Isabelle Sivignon and David Coeurjolly. From Digital Plane Segmentation to Polyhedral Representation. In *Geometry, Morphology, and Computational Imaging*, pages 356–367, Berlin, Heidelberg, 2003.

[SC09] Isabelle Sivignon and David Coeurjolly. Minimum decomposition of a digital surface into digital plane segments is NP-hard. *Discrete Applied Mathematics*, pages 558–570, 2009.

[SDC04] Isabelle Sivignon, Florent Dupont, and Jean-Marc Chassery. Decomposition of a Three-Dimensional Discrete Object Surface into Discrete Plane Pieces. *Algorithmica*, pages 25–43, 2004.

[Siv04] Isabelle Sivignon. *De la caractérisation à la reconstruction polyédrique de surfaces en géométrie discrète*. PhD thesis, Grenoble, Institut national polytechnique de Grenoble, 2004.

[TTL$^+$22] Olena Tankyevych, Flora Trousset, Claire Latappy, Moran Berraho, Julien Dutilh, Jean Pierre Tasu, Corinne Lamour, and Catherine Cheze Le Rest. Development of radiomic-based model to predict clinical outcomes in non-small cell lung cancer patients treated with immunotherapy. *Cancers*, 14(23):5931, 2022.

[VC00] Joëlle Vittone and Jean-Marc Chassery. Recognition of Digital Naive Planes and Polyhedrization. In *Discrete Geometry for Computer Imagery*, pages 296–307, Berlin, Heidelberg, 2000.

https://www.doi.org/10.24132/CSRN.3401.18

Measuring the Influence of Alcohol Consumption on Presence in Virtual Reality

Fabio Genz[1], Fabian Dreer[1], Florian Krötz[1], Marco D'Amelio[2], and Dieter Kranzlmüller[1]

[1]MNM-Team, LMU München, Oettingenstr. 67, 80538 München, Germany
[2]LMU München, Geschwister-Scholl-Platz 1, 80539 München, Germany
fabio.genz@nm.ifi.lmu.de

ABSTRACT

We examine the influence of alcohol consumption on presence in Virtual Reality (VR) with both subjective and objective data. To measure the level of presence in VR we propose a method using four self-developed indicators, two subjective (*Flow*, *Subjective Behaviour*) and two objective ones (*Objective Behaviour*, *Performance*). To assess the validity of our method, we conducted a user study ($n = 20$). Although results show no significant correlations, we reveal two potential research gaps regarding the general threshold for measurable effects of alcohol consumption in VR and eventual gender differences. Besides the two research gaps, we show three further approaches for future work.

Keywords
Virtual Reality, Presence, Alcohol Consumption, Human-Computer-Interaction, Games

1 INTRODUCTION

Virtual Reality (VR) has reached the mass market, with an ongoing growth in the number of users. In 2023, 14 million Head Mounted Displays (HMD) were sold, leading to estimated 22 million units in active use [Ess23]. Around half of the world's population consumes alcohol as estimated by Hoek et al. [Hoe+22]. Therefore, questions related to the use of VR under the influence of alcohol are increasingly relevant.

To date, there has been little research on this topic [Dur+18]. While previous research e.g. investigated effects on cybersickness [IWB17], questions about presence remain unanswered. To help close this research gap, our paper contributes in two ways:

(i) A method to measure the level of presence in VR in form of four self-developed indicators: two subjective (*Flow*, *Subjective Behaviour*) and two objective (*Objective Behaviour*, *Performance*).
(ii) A user study ($n = 20$) revealing two potential research gaps regarding the general threshold value for measurable effects of alcohol consumption in VR and possible gender differences.

In Chapter 2 we review related work. Our proposed method is presented in Chapter 3. While Chapter 4 describes the conducted user study, we show the results in Chapter 5. After critically reflecting on the results in Chapter 6, we conclude and outline potential future work in Chapter 7.

2 RELATED WORK

Our examination of alcohol's influence on presence in VR comprises three areas of research. In this section we look at literature on: the concept of presence in VR (Section 2.1), ways to measure presence in VR (Section 2.2), and previous research on alcohol and VR (Section 2.3).

2.1 Presence and Immersion

Presence and *Immersion* are two key concepts in VR research, closely linked together and often mistakenly used synonymously even among VR experts [Gen+21a; Sla03]. To better understand and clearly distinguish between them, we are taking a closer look at both.

2.1.1 Presence

With VR technologies developing into consumer-level products in the early 1990s, research looked for approaches "to define VR in terms of the human experience rather than the technological hardware" [Ste92]. Subsequently, the concept of *(Virtual) Presence* emerged as a dependent measure of the individual experience in VR. This human experience-focused view, sparked an ongoing debate about the definition of *Presence* in VR [Bar16].

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.

https://www.doi.org/10.24132/CSRN.3401.19

We follow the definition of Slater [Sla09], one of the most commonly cited definitions of *Presence*, as a "feeling of being there", being in a place or virtual environment (VE) even though one is physically in a different place [Min80; Hee92; WS98].

The term can be further distinguished in *Plausibility Illusion* and *Place Illusion*. *Plausibility Illusion* describes how credible events, that happen in the VE, are for the user. *Place Illusion* describes the deception of being in a place created by the *Immersion* of the senses [Sla09].

Factors influencing *Presence* in VE can be separated in interoceptional and exteroceptional. Interoceptional factors relate to the presented content, e.g. participants perceive a greater *Presence* if they feel emotionally affected. An exteroceptional factor is, for example, the degree of *Immersion* [Mar+20]. Slater [Sla09] considers *Immersion* as an influencing factor on *Presence*, describing it as a "human reaction to immersion".

To shed more light on the link between *Presence* and *Immersion* and to differentiate between the two terms, we will subsequently have a closer look at the concept of *Immersion*.

2.1.2 Immersion

Regarding *Immersion* we follow Slater and Wilbur's [SW97] definition "as a quality of the system's technology, an objective measure of the extent to which the system presents a vivid VE while shutting out physical reality".

Definitions are united by understanding the term as a sensory perception triggered by technical elements. Yet it is difficult to provide a standardized definition [BA20] since the literature offers a number of variations [KB18; Nas+00; WS98].

Selzer and Castro [SC23] list variables influencing *Immersion* and categorize them according to different sensorial perceptions.

We found the following visual variables: field of view [Kim+14; WT13], screen resolution [Kim+14; Ahn+14], stereopsis [Kim+14; Ahn+14], response time or latency [KLP20], brightness, contrast, saturation and sharpness [MG96], the level of detail of 3D models [Vol+20], the lighting of the VE [SSC10] and the use of dynamic shadows [SSC10].

Audio-related variables include the use of sound versus no sound [Zel92; PWD13], ambient sound [Bim11], 3D spatial sound [AJC14; Ber+17], the use of headphones versus speakers [Ber+17] and echo or reverberation [Bim11].

Variables related to the tactile system and tracking of the user include sensory bandwidth [Sno98], level of body tracking [Gor+11], degrees of freedom [Bim11], affordance of controls [Wil13], response time or latency of tracking [ABW93], locomotion [Sel18], temperature and wind [AJC14; SC23].

2.1.3 Drawing the Line between Presence and Immersion

To better distinguish *Presence* and *Immersion*, we refer to an abstract comparison from the field of colour science.

Objectively, a colour can be described by its corresponding wavelength distribution. The perception and emotional reaction to a colour by different individuals can in turn have a wide range of results. Thus, *Immersion* can refer to the wavelength distribution, while *Presence* corresponds to colour perception [Sla+09].

2.2 Measuring Presence in VR

Methods to measure *Presence* in VR can be divided in subjective and objective ones. For subjective methods, questionnaires are currently the predominant form [Sch21]. Schwind et al. [Sch+19] identified 15 different questionnaires. Objective methods are e.g. physiological measurements, behavioral observations and performance measurements [Laa+15].

Regarding physiological measurements the most common are heart rate, heart rate variability and skin conductance [Sch21; Sla+22]. Other approaches include, e.g. eye tracking [Sch21], electrodermal activity [Mee+05], muscular responses measured via electromyography [AS11; Kiv+11], functional magnetic resonance imaging [Hof+03] or the integration of electroencephalography [Bau+06; JP16; KN12; KKN12; Cle+14; Pet+20], although the acquisition and interpretation of these data pose a number of challenges [Gen+21b].

Behavioral observations are much-discussed in research and considered experimental, sometimes seen as the most creative method [Sch21]. Sheridan [She92] introduced the idea early on with his approach to evaluate social behaviour in relation to naturalness. Previous results indicate that *Presence* correlates with behaviour in a highly immersive VE and behaviour depends heavily on the specific circumstances under which data points are measured [BA20].

For performance it has been postulated that greater *Presence* enhances task performance. Nevertheless, there is a long debate about the correlation between performance and *Presence* [Wel99]. While some studies show positive correlations, reports of insignificant or even negative correlations exist as well [BA20]. Nash et al. [Nas+00] proposed several measures for performance measurement linked to *Presence*.

Although a variety of methods exists the use of either subjective or objective is insufficient [Sla+22]. Potential disadvantages of questionnaires include memory impairment for extensive questionnaires after an experience [SC23; IWB17] or a potential disturbed feeling of presence in case of short questions during an experience (e.g. 1-item questionnaire) [Sch21]. A purely objective method in turn is difficult due to the subjective nature of *Presence* [Nas+00; BA20]. Hence, literature recommends to combine subjective and objective measures [SBW18; Sla+22].

In their literature review Skarbez et al. [SBW18] recommend considering at least two different methods. Slater et al. [Sla+22] describe the triangulation of using several approaches, e.g. the combination of subjective reports and objective measures, as ideal.

After defining important terms and examining current approaches to measure *Presence* in VR, we take a closer look at research on alcohol and VR next.

2.3 Alcohol and Virtual Reality

A person's alcohol level is measured as blood alcohol concentration (BAC) in alcohol per mille (‰). This value indicates how many millilitres of pure alcohol are contained in one litre of blood.

Alcohol has a direct influence on the human sensory system [IWB17]. Alongside the amount of alcohol consumed its influence depends on a number of factors such as age, gender, genetics, body weight, type of drink, drinking experience, health status and drug use [IWB17; Roe+94; HKW00; Mum+99]. First impairments occur from 0.2 ‰, e.g. in form of slight concentration and problems to focus, reaction delays and deterioration in movement coordination. With increasing BAC restrictions intensify and expand, e.g. on vision. More than 3 ‰ can lead to death [IWB17; BFV20; Gai+18].

Literature on alcohol and VR can be divided into two major research directions. First, the use of VR in alcohol studies. Second, the use of alcohol in VR studies.

While we were able to identify a variety of studies and approaches for the first research direction (e.g. cue exposure therapy [Ghi+19], driving simulators [BFV20], medical training [Gil+23] or educational applications [Lyk+20]), only one study could be assigned to the second research direction, where we also attribute our work to.

As a study targeting the use of alcohol in VR, Iskenderova et al. [IWB17] investigated how alcohol consumption affects cybersickness in VR by conducting a user study ($n = 31$). Cybersickness is one of the main adverse effects of VE and characterised by symptoms like nausea, eye pain, sweating, disorientation, fatigue, headaches and vomiting [BC03]. The study revealed that a BAC of around 0.07 ‰ significantly reduces symptoms of cybersickness. The results are remarkable, as e.g. Blasiis et al. [BFV20] or Gaibler et al. [Gai+18] indicate first noticeable effects above a BAC of 0.2 ‰.

As alcohol has direct effects on human sensory perception [IWB17], we assume correlations between BAC and *Presence* in VR. Although various approaches to measure *Presence* in VR exist, none seems adequate. We therefore propose a method to measure *Presence* in VR in the next section.

3 METHOD

Our proposed method to measure *Presence* in VR consists of four self-developed indicators: two subjective (*Flow*, *Subjective Behaviour*) and two objective ones (*Objective Behaviour*, *Performance*).

The term *Flow* is based on the flow theory of Csikszentmihalyi [Csi90] and describes a positive mental state where people lose their sense of time when they are completely involved in an activity. We consider this state of involvement as an indication for presence in VR. To measure the level of this flow-like state, and therefore presence, we subtract the subjectively perceived time from the actually measured time under the assumption that the difference determines a potential flow-like state. A higher deviation therefore means a higher level of presence.

Both *Subjective Behaviour* and *Objective Behaviour* are based on previous approaches to derive presence from behavioral observations [BA20]. While the literature categorises behaviour as an objective measurement, our method adds a subjective perspective.

For *Subjective Behaviour* we assume presence is correlated with the perceived ability to better position and interact in the VE. For *Objective Behaviour* we assume presence is correlated with the usage frequency of input devices (e.g. controller) for locomotion. If a person uses such devices less frequently, as a result moving more naturally in the VE, we conclude the "feeling of being there", respectively the level of presence, to be higher.

Performance is based on the assumption that better task performance in VE correlates with a higher level of presence in VR [BA20].

To measure subjective data we suggest using short questionnaires presented directly after an experience. For objective data we suggest manually measuring from direct observations or using tracking data from modern HMDs.

In the next chapter we apply and assess the validity of our method in the context of a conducted user study.

4 USER STUDY

In this chapter we describe details regarding the conducted user study for the evaluation of our method to measure *Presence* in VR. We explain the study design, give an overview of the system used as well as the task to be performed. Furthermore we characterise the participants and detail our application of the method plus the general study procedure.

4.1 Study Design

The user study employs a quantitative research design. We chose a between-group experimental design, measuring and comparing the collected variables in two groups. A non-alcoholised group (control group) and an alcoholised group (alcohol group).

4.2 System Overview

The user study was conducted in a laboratory room at our university, which provides an empty area of $3,30m \times 1,80m$. We used a Meta Quest 3 as HMD. As a standalone device it is wireless with an integrated processor, battery-operated, and offers inside-out tracking, which enables the localisation and demarcation of the VE within the real environment as well as game boundaries in a so-called *guardian*. This HMD has a resolution of 2064×2208 pixels per eye, a refresh rate of up to 120 Hz and a field of view of 110° horizontally and 96° vertically. The contained lenses can be adjusted to the pupillary distance of the user.

The HMD is supplied with two (right and left) *Touch Plus Controllers* as input devices. Each controller contains four buttons, a thumb stick, a thumb rest, and a two-stage trigger.

Streaming the visual content from the HMD to a browser is possible enabling us to observe users from outside as well as inside the VR application. The observer setup consists of a workstation with two screens, a mouse and a keyboard.

As the audiovisual stimulus we presented the commercially available VR game *Walkabout Mini-Golf*, which simulates playing mini-golf from a first-person perspective in various VEs with minimalist comic graphics. The user is able to look around the VE by using the head tracking capabilities of the HMD. Locomotion is possible by using the controller in different ways: teleportation by pull and release of the two-stage trigger, snapturning by turning the thumb rest left or right, or real walking within the previously defined guardian. Depending on whether the user is right-handed or left-handed, the golf club is controlled using the right or left controller.

We chose this application since Mini Golf requires no complicated explanation, offers measurable interaction and spatial aspects and, as one of the most popular games in the Oculus Store, promises a certain motivation and enjoyment to participate in the user study.

4.3 Task

The task consisted of two games. First, participants had to hit 10 golf balls at targets on a virtual driving range. The number of points for hitting one of these targets varied, depending on distance, size and whether the targets were static or moving. Collecting as many points as possible was the goal.

Second, after selecting the level *Blossom Tree*, participants had to play three classic mini-golf courses. The aim was to hit a golf ball, past obstacles, into a hole in as few strokes as possible.

4.4 Participants

All participants ($n = 20$; sex: 70% male, 30% female, age: 22 - 60 years; M = 31.5, SD = 9.7) were recruited at university. Participation was voluntary, and took place outside of working hours or study time. Participants were divided into two groups:

First, a control group ($n = 11$; sex: $54,55\%$ male, $45,45\%$ female; Age: 22 - 60 years; M = 32.5, SD = 13.3) where 55.5 % stated to have experience in VR.

Second, an alcohol group ($n = 9$; sex: $88,89\%$ male, $11,11\%$ female; Age: 27 - 35 years; M = 29.9, SD = 2.4) where 60 % stated to have experience in VR.

4.5 Method Application

We measure presence by applying our method described in Section 3, yet first we detail the variables used for the measurements.

4.5.1 Measured Variables

We measured eleven variables in total, six of those through a self-designed questionnaire.

Three objective variables from the questionnaire by asking for *age*, *gender* and *previous experience with VR*. And three subjective variables by asking the following questions:

Q1) "How would you rate your ball control?"
Q2) "How precisely were you able to take up your desired position?"
Q3) "How many minutes did you spend using the application in total?"

Q1) and Q2) were answered using a five point likert scale (1 = very good, 2 = good, 3 = neutral, 4 = bad, 5 = very bad). Q3) was answered in a free field.

Furthermore, three objective variables by manually measuring the *number of snapturns*, *number of teleportations* and *total time spent*. One objective variable by measuring *BAC*, using the *ACE X Alkoholtester*, a breath alcohol tester labeled *Accuracy Class 1* with a precision of ±0.005 percent [Ros19].

One objective variable by measuring *pupillary distance*, using the app *Dotty EyeMeasure* (Version 1.22) on an iPhone 13 mini (iOS Version 17.2.1), which was used for configuration purposes only.

4.5.2 Measuring Presence

We determine *Flow*, as shown in Equation 1, by subtracting the result of Q3) from the measured time.

$$f = total\ time\ spent - q_3 \tag{1}$$

Subjective Behaviour, as shown in Equation 2, is calculated by halving the sum of the results of Q1) and Q2).

$$u = \frac{(q_1 + q_2)}{2} \tag{2}$$

We determine *Objective Behaviour*, as shown in Equation 3, as the sum of the number of *snapturns* s and the number of teleportations t.

$$o = s + t \tag{3}$$

We determine *Performance*, as shown in Equation 4, with n number of ranges, in our case $n = 3$, $h_i^{\max}$ the number of max shots allowed at the ith hole and h_i number of shots on the ith hole. This evaluates the performance of the holes to a percentage value, where 100% corresponds to clearing each hole in one shot and 0% to the maximum number of attempts required.

$$p = \frac{100}{(\sum_{i=1}^{n} h_i^{\max}) - n} \cdot \left(\sum_{i=1}^{n} h_i - n \right) + 100 \tag{4}$$

4.6 Study Procedure

Before taking part in the user study, potential participants were informed about possible exclusion criteria (e.g. age, health problems, current medication use, current or previous addiction problems and pregnancy).

The HMD was kept charged during the user study. Backup batteries were available for the controllers. If required, we provided support by answering general questions.

The user study consisted of two rounds. Each round contained seven sub-steps, which are described below. Figure 1 illustrates the execution of the user study.

(1) Information: Participants were informed at the beginning of the user study about the purpose, the individual steps and the right to discontinue the study at any time and for any reason. A written declaration of consent was obtained from all participants. In addition, information was provided in accordance with Art. 13 GDPR.

(2) Measurement: Prior to each round, we measured each participant's current BAC and pupillary distance. In the first round, all participants had to be sober. We used this round as the baseline for the measured values. In the second round, we made a distinction between two groups. The control group was asked to remain sober. The alcohol group was asked to bring and consume alcoholic beverages at their own discretion. To prevent falsification of the BAC test, all participants were asked to refrain from smoking, eating or using a mouth spray for at least 15 minutes before each round.

(3) Preparation: Before the HMD was handed over, the application was started, the pupillary distance was set, the streaming of the visual content from the HMD was started, participants were asked whether they are left- or right-handed, and the HMD and corresponding controller were disinfected.

(4) Customisation: After handing over the HMD and the controller to the participant, we explained how the HMD can be adjusted to the respective head size.

(5) Familiarisation: At the beginning of the VR application, the participant is located in a freely walkable VE. We use this mode to explain the controls and letting them familiarise.

(6) Execution: When the participants communicated that they were ready, we explained the task. Afterwards we started our measurements.

(7) Questionnaire: After completing the task, participants were asked to answer our questionnaire.

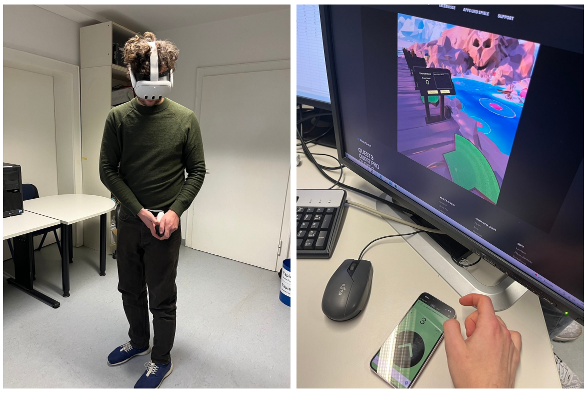


Figure 1. Pictures of a user conducting the user study (left) and the authors' parallel perspective while measuring various parameters (right)

5 RESULTS

We conducted a user study to measure the influence of alcohol on *Presence* in VR. The measurements are based on our method which combines objective as well as subjective indicators. In this section we present the obtained results. A critical discussion of these follows in the next section 6.

The levels of BAC ranged from a minimum of 0.07 ‰ to a maximum of 0.93 ‰. The mean concentration was 0.31 ‰, with a standard deviation of 0.33. Six participants registered values below 0.2 ‰, while three participants exceeded the 0.4 ‰ threshold.

Table 1 shows the descriptive statistics for the measured variables. None of the observed trends reached statistical significance.

		Round 1		Round 2	
Group		mean	sd	mean	sd
Control	f	5.29	7.01	3.93	3.70
	u	2.32	0.78	1.82	0.46
	o	29.4	9.99	27.5	11.0
	p	45.5	17.6	62.6	23.8
Alcohol	f	1.11	2.67	1.17	1.99
	u	1.78	0.36	1.61	0.33
	o	53.0	38.2	51.7	35.7
	p	46.5	15.7	63.0	19.7

Table 1. Descriptive Statistics for the measured variables *Flow* (f), *Subjective Behaviour* (u), *Objective Behaviour* (o) and *Performance* (p)

Distinctions emerged between the groups during round 1 for *Subjective Behaviour*, *Objective Behaviour* and *Flow*.

The alcoholised group had lower scores for *Subjective Behaviour* (1.78 vs. 2.32, 95% CI [-1.11, 0.03]), *Objective Behaviour* (53.0 vs. 29.4, 95% CI[-6.01, 53.28]) as well as *Flow* (1.11 vs. 5.29, 95% CI[-9.13, 0.76]). Despite these differences, both groups demonstrated a comparable *Performance* of 46% (95% CI[-14.67, 16.69]).

There is no remarkable change in the mean value for *Flow*, *Subjective Behaviour* and *Objective Behaviour* between rounds 1 and 2 for both groups.

Notably, there was a *Performance* improvement from 45% to 63% (control 95% CI[-35.89, 1.55], alcohol 95% CI[-34.37, 1.37]) between round 1 and 2 for both groups.

6 DISCUSSION

In this section, we discuss the previously presented results, reveal two potential research gaps and point out limitations.

First, we expected measurable effects of alcohol on *Presence* in terms of different mean values between round 1 and 2 for both groups. Instead, our results show no significant changes for *Flow*, *Subjective Behaviour* and *Objective Behaviour*. By taking a closer look at the dataset, we saw that six participants registered a BAC below 0.2 ‰ and only three exceeded 0.4 ‰. As the literature seems unable to define a clear threshold value for measurable effects of BAC in VR [BFV20; Gai+18; IWB17], this indicates a first potential research gap.

Second, we observe different mean values in the first round for *Flow*, *Subjective Behaviour* and *Objective Behaviour* between both groups. We noticed that there is only one woman in the group that consumed alcohol in contrast to the control group, which is balanced. Since the measurements between the groups in round 1 already differ more than the standard deviation, this indicates a second potential research gap namely possible gender differences on *Presence* in VR.

Since *Performance* improved for both groups in the same manner, alcohol appears to have no influence on attributed learning effects in VR.

There are several shortcomings in our study. First and foremost, none of the observed trends reached statistical significance. We mainly attribute this to the limited sample size of the user study. The data set also suffers from an unbalanced distribution of BAC within the intoxicated group, as well as an imbalanced gender composition.

7 CONCLUSION AND FUTURE WORK

Questions related to the use of VR under the influence of alcohol are of increasing relevance. Yet there has been little research on this topic. As alcohol has influence on the human sensory system we expected a measurable influence on presence in VR. Although literature offers various approaches to measure presence in VR none seems adequate.

We presented a method to measure presence in VR in form of four self-developed indicators. To assess the validity of our approach, we conducted a user study ($n = 20$). Even though the results show no significant results, we revealed two potential research gaps: an unclear BAC threshold value for measurable effects on presence in VR and possible gender differences on presence in general.

Besides these two, we see three more approaches for future work. First, the proposed method of this work could be evaluated in a larger user study with more participants, a balanced BAC distribution in the alcoholised group and a balanced gender composition.

Second, the proposed method could be modified by changing the questionnaire (e.g. questions, timing of questions), integrating physiological measures (e.g. heart rate, electrodermal activity or electroencephalograph [Laa+15; Sla+22]), or using more data sources of modern HMDs like eye tracking.

Third, the user study could be replicated in the real world, on a real mini-golf course, to compare the results and examine to what extent VR reflects reality here and how this shows in the chosen metrics.

ACKNOWLEDGMENTS

We would like to thank the MNM-Team members, especially Vitalian Danciu, who supported this work with discussions and feedback, as well as the participants of the study.

REFERENCES

[ABW93] K. W. Arthur, K. S. Booth, and C. Ware. “Evaluating 3D task performance for fish tank virtual worlds”. In: *ACM Trans. Inf. Syst.* 11.3 (July 1993), pp. 239–265.

[Ahn+14] D. Ahn et al. “The effects of actual human size display and stereoscopic presentation on users’ sense of being together with and of psychological immersion in a virtual character”. In: *Cyberpsychology, behavior and social networking* 17.7 (2014), pp. 483–487.

[AJC14] A. S. Azevedo, J. Jorge, and P. Campos. “Combining EEG Data with Place and Plausibility Responses as an Approach to Measuring Presence in Outdoor Virtual Environments”. In: *Presence: Teleoperators and Virtual Environments* 23.4 (2014), pp. 354–368.

[AS11] A. Antley and M. Slater. “The effect on lower spine muscle activation of walking on a narrow beam in virtual reality”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.2 (2011), pp. 255–259.

[BA20] M. I. Berkman and E. Akan. “Presence and Immersion in Virtual Reality”. In: *Encyclopedia of Computer Graphics and Games*. Ed. by N. Lee. Springer eBook Collection. Cham: Springer International Publishing, 2020, pp. 1–10.

[Bar16] W. Barfield. “Musings on Presence Twenty-Five Years after “Being There“”. In: *Presence: Teleoperators and Virtual Environments* 25.2 (2016), pp. 148–150.

[Bau+06] T. Baumgartner et al. “Neural correlate of spatial presence in an arousing and noninteractive virtual reality: an EEG and psychophysiology study”. In: *CyberPsychology & Behavior* 9.1 (2006), pp. 30–45.

[BC03] G. C. Burdea and P. Coiffet. *Virtual reality technology*. 2. ed. New York: John Wiley and Sons Ltd, 2003.

[Ber+17] I. Bergstrom et al. “The Plausibility of a String Quartet Performance in Virtual Reality”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.4 (2017), pp. 1352–1359.

[BFV20] M. R. de Blasiis, C. Ferrante, and V. Veraldi. “Driving Risk Assessment Under the Effect of Alcohol Through an Eye Tracking System in Virtual Reality”. In: *Advances in Safety Management and Human Factors*. Ed. by P. M. Arezes. Vol. 969. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2020, pp. 329–341.

[Bim11] Bimal, Balakrishnan. S. Shyam Sundar. “Where Am I? How Can I Get There? Impact of Navigability and Narrative Transportation on Spatial Presence”. In: *Human–Computer Interaction* 26:3 (2011), pp. 161–204.

[Cle+14] M. Clemente et al. “Assessment of the influence of navigation control and screen size on the sense of presence in virtual reality using EEG”. In: *Expert Systems with Applications* 41.4 (2014), pp. 1584–1592.

[Csi90] M. Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper & Row., Jan. 1990.

[Dur+18] J. Durl et al. “Utilising virtual reality in alcohol studies: A systematic review”. In: *Health Education Journal* 77.2 (2018), pp. 212–225.

[Ess23] R. Esser. *Technology Industry Briefing: Neue Perspektiven für Virtual Reality | Deloitte Deutschland*. Ed. by Deloitte Deutschland. 2023.

[Gai+18] F. Gaibler et al. “Drink & Drive: A Serious but Fun Game on Alcohol-Induced Impairments in Road Traffic”. In: *7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games)* (2018).

[Gen+21a] F. Genz et al. “Evaluation of Proprietary Social VR Platforms for Use in Distance Learning”. In: *Augmented reality, virtual reality, and computer graphics : 8th international conference, AVR 2021, virtual event, September 7-10, 2021, proceedings*. Springer, 2021, pp. 462–480.

[Gen+21b] F. Genz et al. “Replacing EEG Sensors by AI Based Emulation”. In: *Augmented Reality, Virtual Reality, and Computer Graphics*. Ed. by L. T. de Paolis, P. Arpaia, and P. Bourdot. Vol. 12980. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 66–80.

[Ghi+19] A. Ghiţă et al. “Cue-Elicited Anxiety and Alcohol Craving as Indicators of the Validity of ALCO-VR Software: A Virtual Reality Study”. In: *Journal of clinical medicine* 8.8 (2019).

[Gil+23] C. Gilmour et al. “Pain and alcohol consumption in virtual reality”. In: *Experimental and clinical psychopharmacology* 31.2 (2023), pp. 433–442.

[Gor+11] A. Gorini et al. “The role of immersion and narrative in mediated presence: the virtual hospital experience”. In: *Cyberpsychology, behavior and social networking* 14.3 (2011), pp. 99–105.

[Hee92] C. Heeter. *Being there: The subjective experience of presence: Presence: Teleoperators and Virtual Environments*. MIT Press, 1992.

[HKW00] L. Holdstock, A. C. King, and H. de Wit. “Subjective and Objective Responses to Ethanol in Moderate/Heavy and Light Social Drinkers”. In: *Alcoholism: Clinical and Experimental Research* 24.6 (2000), pp. 789–794.

[Hoe+22] A. G. Hoek et al. “Alcohol Consumption and Cardiovascular Disease Risk: Placing New Data in Context”. In: *Current atherosclerosis reports* 24.1 (2022), pp. 51–59.

[Hof+03] H. G. Hoffman et al. “The illusion of presence in immersive virtual reality during an fMRI brain scan”. In: *Cyberpsychology & behavior : the impact of the Internet, multimedia and virtual reality on behavior and society* 6.2 (2003), pp. 127–131.

[IWB17] A. Iskenderova, F. Weidner, and W. Broll. “Drunk Virtual Reality Gaming”. In: *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. Ed. by B. Schouten et al. New York, NY, USA: ACM, 2017, pp. 561–572.

[JP16] S. Jeon and W. Park. *Computing Presence in Nonfigurative Virtual Environment under Color and Motion Stimulus: 6-7 December 2016, San Francisco : FTC*. Proceedings of Future Technologies Conference. Piscataway, NJ: IEEE, 2016.

[KB18] G. Kim and F. Biocca. “Immersion in Virtual Reality Can Increase Exercise Motivation and Physical Performance”. In: *Virtual, Augmented and Mixed Reality: Applications in Health, Cultural Heritage, and Industry*. Ed. by J. Y. Chen and G. Fragomeni. Vol. 10910. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 94–102.

[Kim+14] K. Kim et al. “Effects of virtual environment platforms on emotional responses”. In: *Computer methods and programs in biomedicine* 113.3 (2014), pp. 882–893.

[Kiv+11] J. M. Kivikangas et al. “A review of the use of psychophysiological methods in game research”. In: *Journal of Gaming & Virtual Worlds* 3.3 (2011), pp. 181–199.

[KKN12] S. E. Kober, J. Kurzmann, and C. Neuper. “Cortical correlate of spatial presence in 2D and 3D interactive virtual reality: an EEG study”. In: *International journal of psychophysiology : official journal of the International Organization of Psychophysiology* 83.3 (2012), pp. 365–374.

[KLP20] J. Kim, W. Luu, and S. Palmisano. “Multisensory integration and the experience of scene instability, presence and cybersickness in virtual environments”. In: *Computers in Human Behavior* 113 (2020), p. 106484.

[KN12] S. E. Kober and C. Neuper. “Using auditory event-related EEG potentials to assess presence in virtual reality”. In: *International Journal of Human-Computer Studies* 70.9 (2012), pp. 577–587.

[Laa+15] J. Laarni et al. “Ways to Measure Spatial Presence: Review and Future Directions”. In: *Immersed in Media*. Ed. by M. Lombard et al. Cham: Springer International Publishing, 2015, pp. 139–185.

[Lyk+20] P. B. Lyk et al. “Co-Designing and Learning in Virtual Reality: Development of Tool for Alcohol Resistance Training”. In: *Electronic Journal of e-Learning* 18.3 (2020), pp. 219–234.

[Mar+20] J. Marín-Morales et al. “Emotion Recognition in Immersive Virtual Reality: From Statistics to Affective Computing”. In: *Sensors (Basel, Switzerland)* 20.18 (2020).

[Mee+05] M. Meehan et al. “Review of four studies on the use of physiological reaction as a measure of presence in stressful virtual environments”. In: *Applied psychophysiology and biofeedback* 30.3 (2005), pp. 239–258.

[MG96] T. Mazuryk and M. Gervautz. “History, applications, technology and future”. In: *Virtual Reality* 72.4 (1996), pp. 486–497.

[Min80] M. Minsky. *Telepresence*. OMNI magazine, 1980.

[Mum+99] M. S. Mumenthaler et al. “Gender differences in moderate drinking effects”. In: *Alcohol Research & Health* 23.1 (1999), p. 55.

[Nas+00] E. B. Nash et al. “A Review of Presence and Performance in Virtual Environments”. In: *International Journal of Human-Computer Interaction* 12.1 (2000), pp. 1–41.

[Pet+20] I. V. Petukhov et al. “Being present in a real or virtual world: A EEG study”. In: *International journal of medical informatics* 136 (2020), p. 103977.

[PWD13] S. Poeschl, K. Wall, and N. Doering. “Integration of spatial sound in immersive virtual environments an experimental study on effects of spatial sound on presence”. In: *2013 IEEE Virtual Reality (VR)*. IEEE, 2013, pp. 129–130.

[Roe+94] T. Roehrs et al. “Sleepiness and ethanol effects on simulated driving”. In: *Alcoholism: Clinical and Experimental Research* 18.1 (1994), pp. 154–158.

[Ros19] E. Rosenberg. *Untersuchung der Genauigkeit (= Richtigkeit und Präzision ausgewählter Atemalkohol-Messgeräte(‘Alkotester’)*. Bericht TUW CTA 2019/01 DE. TU Wien, 2019.

[SBW18] R. Skarbez, J. F. P. Brooks, and M. C. Whitton. “A Survey of Presence and Related Concepts”. In: *ACM Computing Surveys* 50.6 (2018), pp. 1–39.

[SC23] M. Selzer and S. M. Castro. “A Methodology for Generating Virtual Reality Immersion Metrics based on System Variables”. In: *Journal of Computer Science & Technology* 23.2 (2023), e08.

[Sch+19] V. Schwind et al. “Using Presence Questionnaires in Virtual Reality”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Ed. by S. Brewster et al. New York, NY, USA: ACM, 2019, pp. 1–12.

[Sch21] J. Schirm. “Standardizing Objective Measures of Presence in Immersive Virtual Environments". In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. Ed. by Y. Kitamura et al. New York, NY, USA: ACM, 2021, pp. 1–4.

[Sel18] M. N. Selzer. *Interacción humano computadora en ambientes virtuales*. 2018.

[She92] T. B. Sheridan. "Musings on Telepresence and Virtual Presence". In: 1, 1 (1992), pp. 120–126.

[Sla+09] M. Slater et al. *How we experience immersive virtual environments: the concept of presence and its measurement*. 2009.

[Sla+22] M. Slater et al. "A Separate Reality: An Update on Place Illusion and Plausibility in Virtual Reality". In: *Frontiers in Virtual Reality* 3 (2022).

[Sla03] M. Slater. "A Note on Presence Terminology". In: (2003).

[Sla09] M. Slater. "Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments". In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 364.1535 (2009), pp. 3549–3557.

[Sno98] M. P. Snow. *Charting Presence in Virtual Environments and its Effects on Performance*. Virginia Tech, 1998.

[SSC10] M. Slater, B. Spanlang, and D. Corominas. "Simulating virtual environments within virtual environments as the basis for a psychophysics of presence". In: *ACM Transactions on Graphics* 29.4 (2010), pp. 1–9.

[Ste92] J. Steuer. "Defining Virtual Reality: Dimensions Determining Telepresence". In: *Journal of Communication* 42.4 (1992), pp. 73–93.

[SW97] M. Slater and S. Wilbur. "A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments". In: *Presence: Teleoperators and Virtual Environments* 6.6 (1997), pp. 603–616.

[Vol+20] T. Volkmann et al. "What you see isn't necessarily what you get". In: *Proceedings of Mensch und Computer 2020*. Ed. by B. Preim, A. Nürnberger, and C. Hansen. New York, NY, USA: ACM, 2020, pp. 119–128.

[Wel99] R. B. Welch. "How can we determine if the sense of presence affects task performance?" In: *Presence: Teleoperators and Virtual Environments* 8.5 (1999), pp. 574–577.

[Wil13] K. D. Williams. "The Effects of Video Game Controls on Hostility, Identification, and Presence". In: *Mass Communication and Society* 16.1 (2013), pp. 26–48.

[WS98] B. G. Witmer and M. J. Singer. "Measuring Presence in Virtual Environments: A Presence Questionnaire". In: *Presence: Teleoperators and Virtual Environments* 7.3 (1998), pp. 225–240.

[WT13] G. Wallis and J. Tichon. "Predicting the Efficacy of Simulator-based Training Using a Perceptual Judgment Task Versus Questionnaire-based Measures of Presence". In: *Presence: Teleoperators and Virtual Environments* 22.1 (2013), pp. 67–85.

[Zel92] D. Zeltzer. "Autonomy, Interaction, and Presence". In: *Presence: Teleoperators and Virtual Environments* 1.1 (1992), pp. 127–132.

Automatic Data Generation of Incorrect Image-Text Pairs for Effective Contrastive Learning of CLIP Model

Rina Tagami
Chukyo University
Yagoto Honmachi
Showa-ku
101-2, Nagoya,
Japan
tagami@isl.sist.chukyo-u.ac.jp

Hiroki Kobayashi
Chukyo University
Yagoto Honmachi
Showa-ku
101-2, Nagoya, Japan
kobayashi@isl.sist.chukyo-u.ac.jp

Shuichi Akizuki
Chukyo University
Yagoto Honmachi
Showa-ku
101-2, Nagoya, Japan
s-akizuki@sist.chukyo-u.ac.jp

Manabu Hashimoto
Chukyo University
Yagoto Honmachi
Showa-ku
101-2, Nagoya, Japan
mana@isl.sist.chukyo-u.ac.jp

ABSTRACT

In this study, we proposed a method for automatically generating high-quality CLIP(Contrastive Language Image Pre-training) training data to improve the performance of text-based image retrieval using CLIP. In general, two types of image-text pair data are used in CLIP training: correct pairs and incorrect pairs. correct pairs are pairs in which the image and text content are compatible, and are created by scraping or other methods. incorrect pairs are incompatible image-text pairs, which are created by changing the combination of the correct pairs. CLIP is completed by contrastive training to increase the similarity between the image and text in correct pairs and decrease the similarity in incorrect pairs. However, when there are multiple images in the training data that are similar to each other, the text attached to them is also considered to be similar to each other, and although it is preferable to treat them as correct pairs, changed pairs are treated as incorrect pairs. In other words, incorrect pairs with high relevance between image texts are learned as having low relevance between image texts, and this inconsistency has a negative impact on the CLIP model. Therefore, if two images taken from the training data are not similar, then the similarity between texts assigned to them should also be low, so that a highly reliable incorrect pair can be created by exchanging the assigned text with each other. We applied this idea to the results of clustering the images and texts in the training data, respectively, and used the similarity between the clusters to generate an incorrect pair, then learned to increase the negative effect as the similarity between images was lower. The results of an experiment using the Amazon review dataset, which is commonly used in this field, showed a 21.0% improvement in Rank@1 score compared to vanilla CLIP.

Keywords

Large language Models, Image Retrieval, Image-Text Dataset, CLIP, Contrastive Learning, K-means Clustering

1 INTRODUCTION

The use of online shopping has increased in recent years due to the ease and convenience of purchase. Online shopping applications are equipped with a function that allows users to search for desired products by entering keywords or sentences into a search system. As the usage rate of the search system increases in proportion to the usage rate of the application, there is a need to improve the search accuracy. However, when keywords or text are input, images unintended by the user are sometimes output as search results. This study proposes a methodology for a search system that outputs images when text is input to output images that meet the user's intention.

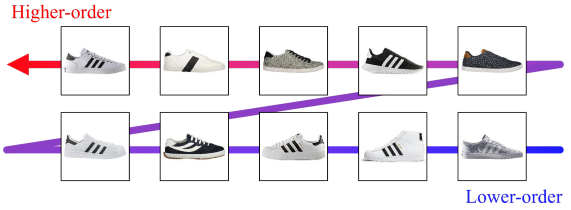


Figure 1: Output results when the text "Black and white striped sneaker." is entered into the vanilla CLIP (top 10).

Various image retrieval methods have been proposed, the most widely used of which [SIG08] assign keywords or phrases related to images as "tags" match the tags with text queries, and output images contain-

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.

https://www.doi.org/10.24132/CSRN.3401.20

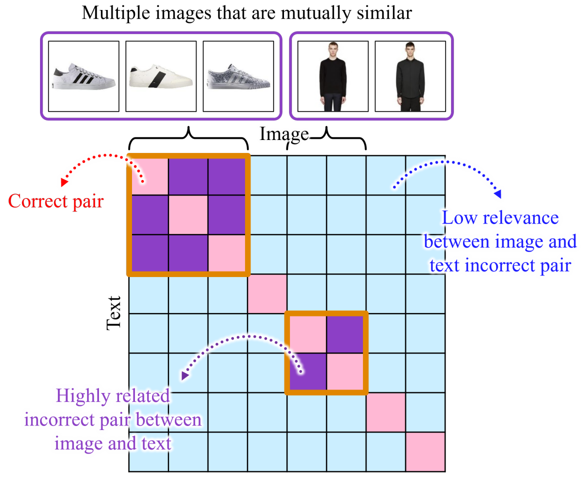

Multiple images that are mutually similar
Image
Text
Correct pair
Low relevance between image and text incorrect pair
Highly related incorrect pair between image and text

Figure 2: Relationship between correct and incorrect pairs.

ing tags with high similarity in a ranking format. The risk here that if the annotator tags an image incorrectly, or if the tagging is highly ambiguous, the result will be unintended. There are also automatic captioning techniques [KIL16] using deep learning and image retrieval [GIA15] based on the relationship between images and hashtags, but problems remain in terms of the cost of labeling training data and the reduced retrieval accuracy for unknown data. Recently, multimodal approaches [QI20][LI23] have become more available, and in particular, image retrieval using Contrastive Language Image Pre-training (CLIP) [RAD21] is becoming established [BAL22][HEN22]. Because of its zero-shot learning capability, CLIP can output relevant images even for text that is not in the training data. However, the problem of outputting images with low relevance to the input text, as shown in the Figure1, remains unsolved. A previous study [AGA21] on this issue suggested that CLIP performs well on image retrieval for general categories, but may perform poorly on certain tasks due to inherited biases. Other previous study [SHA23] suggests that the linguistic representation of images in a particular category or text describing that category is not well learned if what the image represents does not match the text prompt.

In the proposed method, the CLIP model is improved by modifying the current training data and adjusting the number of data to improve the retrieval accuracy. First, CLIP performs contrastive training using correct pairs (diagonal components in Figure2) and incorrect pairs (off-diagonal components in Figure2), which have high relevance between image and text content. When the training data consists of K pairs of images and texts, we assume that “all of them are correct pairs" and treat all the $K^2 - K$ generated by changing the combination quite of K pairs as incorrect pairs. This is reasonable when the pairs are independent, i.e., the images are not similar to each other. However, when there are multiple images in the training data that are similar to each other, the text attached to the images is also considered to be similar to each other, so all pairs are treated as incorrect pairs, even though it is preferable to treat the pairs with different combinations as correct pairs. Therefore, incorrect pairs (purple area in Figure2), whose contents are highly related to each other, are learned to be less related, which has a negative impact on the CLIP model. In addition, since there is a large difference in the number of data between correct and incorrect pairs, the learning is biased toward a large number of data.

In this study, we propose a method to solve both problems simultaneously. The main idea is to carefully select only the incorrect pairs that are expected to have low relevance between image-text pairs (blue region in Figure2), and adjust the number of data so that the number of incorrect pairs is the same as the number of correct pairs. Specifically, image features are extracted from Vision Transformer (ViT) [DOS20] and text features are extracted from BERT [DEV18], and are clustered together. If the clusters of two arbitrarily selected image features are different and the similarity between images calculated by ViT is low, and if the clusters between the given texts are also different, then the texts are exchanged to generate an incorrect pair. By repeating this process until the number of data is the same as that of the incorrect pair, only incorrect pairs can be generated without bias in the number of data and with low relevance of content between image texts. If the proposed method can successfully increase the accuracy of image retrieval, it will contribute to improving the purchasing effectiveness of online shopping by eliminating the need to filter out unwanted products.

This paper is organized as follows: Section 2 describes related work and their problems. Section 3 describes the proposed method, and Section 4 describes the experimental results of the proposed method and a comparative method. Section 5 provides a conclusion of the proposed method.

2 RELATED WORK

Various methods for image search have been proposed over time, traditionally utilizing tags attached to images or surrounding text content. Recently, multimodal approaches have become feasible, with methods proposed for searching images from input text using the embedding representations of BERT as metadata attached to images [QI20][LI23], as well as using CLIP and ALIGN [JIA21] for image search [BAL22][HEN22]. In fact, numerous image search methods have been proposed for CLIP thanks to its ability to learn semantic relationships between natural language text and image content.

For example, e-CLIP [SHI22] was designed for practical use in online shopping, proposing an image search framework that utilizes CLIP for learning. It is aimed for use in downstream tasks such as category classification, attribute extraction, product matching, product clustering, and adult product recognition, allowing for the reduction of redundant information learning through the deletion of duplicate images using ResNet-34 [TAN19] and hash values, thereby enabling more efficient learning processes. By collecting similar images based on categories for contrastive learning, it has shown high performance in tasks related to the images used for learning, although it faces issues with decreased accuracy in zero-shot tasks.

The EI-CLIP [MA22] method improves the discriminative performance of images in texts containing proper nouns (e.g., Burberry, GUCCI) within the CLIP framework. It vectorizes proper nouns using an entity encoder and associates these vectors with textual embedding representation through an Entity-Aware module while contrastively learning with images. While it shows high discriminative performance for images in texts containing proper nouns, it faces issues with decreased accuracy in texts with high ambiguity or without metadata.

OpenFashionCLIP [CAR23], another method, seeks to enhance image search performance in the fashion domain not by modifying the learning method of CLIP, but by automatically performing prompt engineering on fashion-related texts to improve the quality of input queries. It prepares multiple template prompts and combines randomly chosen template prompts with input texts before feeding them to the text encoder for contrastive learning with images. Although it shows a higher discriminative performance than the baseline CLIP, selecting prompts randomly can result in prompts with low consistency with images, potentially leading to decreased discriminative performance depending on the dataset used.

There is also the RA-CLIP [XIE23], which, like the proposed method, improves the dataset based on the similarity between images. To enrich the information in the image data used to train the CLIP model, this method uses a module called RAM to extend the feature set of images that are similar to the input images. The training using these expanded image features significantly improves the zero-shot accuracy in image retrieval. Similar images play the role of a cheat sheet in CLIP, and the process of the proposed method can be regarded as an open-book test that does not require memorization of all visual information in the training data. Therefore, many image-text relationships can be learned with limited training. However, the real-time collection of similar images is computationally expensive, and retrieval performance degrades when the dataset is not domain- or task-specific.

Target learning by collecting similar images or by improving the quality of images and texts can improve retrieval accuracy to a certain extent. However, when images in the data are similar to each other, they are learned as low similarity of incorrect pairs with high relevance between image-text, and thus do not solve the essential problem. This method proposes a new method for generating good incorrect pairs.

3 PROPOSED METHOD

This section provides an overview of the proposed method and then describes the detailed method procedure.

3.1 Overview of proposed method

The proposed method generates effective incorrect pairs to improve the performance of image retrieval using CLIP. First, a new database (DB) is created from an original DB of image-text pairs (Figure3). The new DB contains correct pairs with strong semantic relevance and incorrect pairs with weak semantic relevance between images and texts, which are used for fine-tuning based on contrastive learning in CLIP. In vanilla CLIP, for K image-text pairs, the corresponding pair in a $K \times K$ matrix is treated as the correct pair, and the rest are learned as incorrect pairs. In contrast, the proposed method generates K incorrect pairs for each K correct pairs and performs contrast learning using these incorrect pairs. The method of generating incorrect pairs is described in Section 3.2, and the learning method using incorrect pairs is described in Section 3.3.

3.2 How to generate incorrect pairs

Unlike vanilla CLIP, the proposed method uses only incorrect pairs with low relevance between image-text pairs for training. First, features are extracted from the images and texts in the original DB shown in the Figure4 (left), using ViT and BERT, respectively, and clustered. In this case, we use K-means clustering [MAC67], which is reportedly effective for sentence modeling and topic modeling in previous studies [ASK21]. Next, one cluster that differs from the cluster of a certain image in the original DB is randomly selected, and the image features that are least similar to the image features in the original DB by Cosine similarity are output from the selected cluster. If the clusters between text features attached to the image are also different, the texts are exchanged. If the clusters between the texts are the same, the second and subsequent dissimilar image features are used to compare the text clusters. In this example, after the exchange, the image representing a jacket is assigned the text “shirt" and the image representing a shirt is assigned the text “jacket", as shown on the right in the

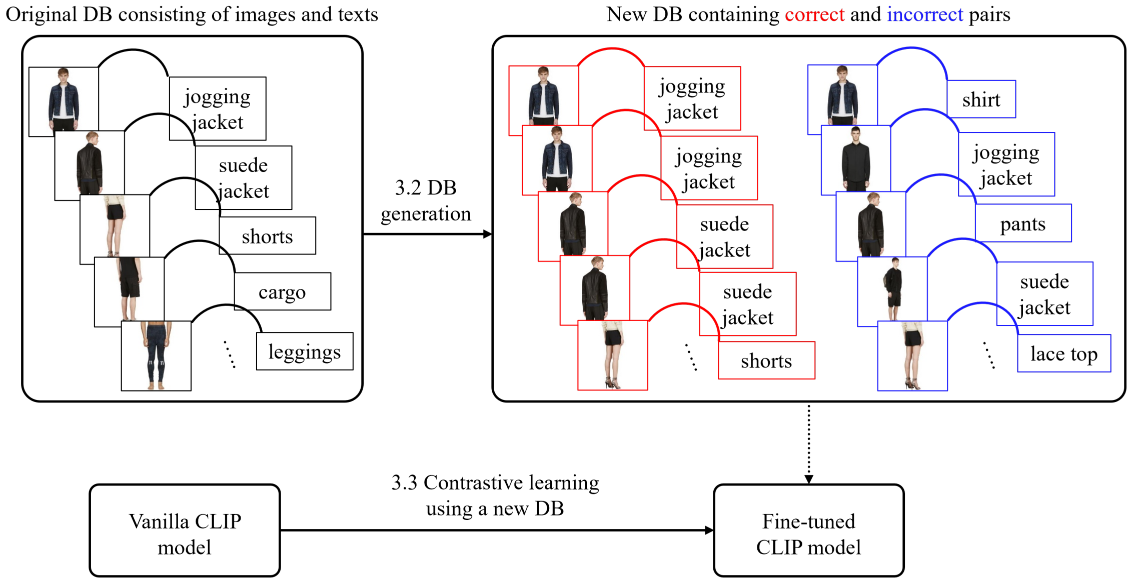


Figure 3: Flow of proposed method.

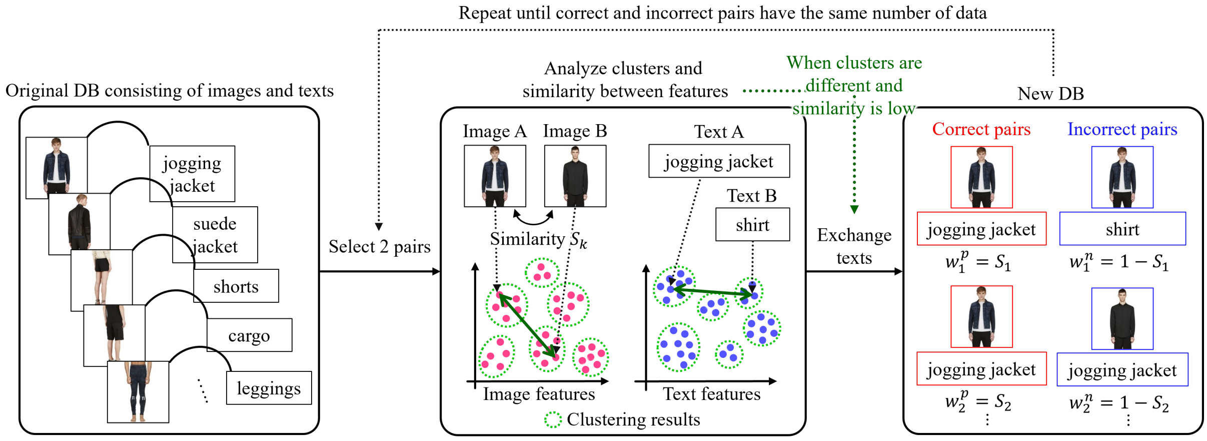


Figure 4: Flow of incorrect pair generation.

Figure. This is repeated until the number of correct pairs of data is reached.

Based on the assumption that "when there are multiple similar images, the texts assigned to the images are also similar to each other," we assign the weights w_k^p and w_k^n to each pair according to the similarity S_k between the two images for each pair (Eq.1, Eq.2). If the images are similar, the learning of correct pairs is enhanced by w_k^p during contrast learning, and if they are dissimilar, the learning of incorrect pairs is enhanced by w_k^n. In this way, the proposed method can separate only irrelevant images and text in the feature space without separating relevant images and text by learning with incorrect pairs generated by the proposed method.

$$w_k^p = S_k \tag{1}$$

$$w_k^n = 1 - S_k \tag{2}$$

3.3 Learning with incorrect pairs

The proposed method performs contrastive training utilizing correct and incorrect pairs with the same number of data. First, image and text features are extracted through encoders. Next, the similarity of the correct pair is trained to increase, while the similarity of the incorrect pair is trained to decrease (Figure 5). Utilizing the similarity between the correct and incorrect pairs of images as a basis, the training weights of the correct pair are balanced with the training weights of the incorrect pair. Specifically, when the similarity between the images of a pair is high, the training weight of the correct pair is increased; when the similarity is low, the training weight of the incorrect pair is increased. In this way, the model is expected to more effectively identify relevant images based on the input text and reduce the output of irrelevant images. The loss is calculated from two terms, as in Eq.3. The first term L_p is the cross-entropy error between the cosine similarity s^p of the correct pair and the label y_k representing the category of the kth image, which is calculated by Eq.4.

The second term L_n is the cross-entropy error between the cosine similarity $1 - s^n$ of the incorrect pair and the label y_k representing the category of the kth image, which is calculated by Eq.5. The similarity between the image and text features is subtracted from 1, so the less similar the features are, the smaller the loss. The lower the similarity between the correct pair of images and the incorrect pair of images, the smaller the value of w_k^p and the larger the value of w_k^n. α and β in Eq.3 are hyperparameters, which are weights that balance the terms. Thus, it can be seen that Learning with weights w_k^p, w_k^n can improve the discriminative performance of images because correct and incorrect pairs can be more distinctly separated in the feature space.

$$L = \alpha L_p + \beta L_n \tag{3}$$

$$L_p = -\sum_{k=1}^{K} y_k \cdot \log(w_k^p s_k^p) \tag{4}$$

$$L_n = -\sum_{k=1}^{K} y_k \cdot \log(w_k^n (1 - s_k^n)) \tag{5}$$

4 EXPERIMENTS AND DISCUSSION

In this section, the purpose and conditions of the experiment are described, followed by the results and discussion of the experiment.

4.1 Experimental conditions

To evaluate the image retrieval performance of the proposed method, we conducted experiments on the task of searching for images from text. The ability to search and output the matching item (GT) from a group of images when a text query is provided is assessed. Following the experimental method of a prior study [GAO20], we utilized 100 randomly selected images from the dataset, conducting image searches using the text as a pair for the image. The experimental results are the average of 100 trials. The evaluation metrics are Rank@1, Rank@5, Rank@10, and Mean Average Precision (mAP). Rank@K indicates the percentage of correct images among the top K search results, and mAP is the average of how often the correct item appears at the top of the search results (AP) for each query. The datasets used were Fashion-gen [ROS18] and Amazon Review Data 2018 [NI19] (Figure 6). The Fashion-gen dataset contains 67,666 image-text pairs, of which 32,213 were used for the experiments. This dataset is characterized by its long and specific texts. The Amazon Review dataset contains 431,492 image-text pairs, with 30% of these being used in the experiments. This dataset provides a more challenging image search task due to its brief and more ambiguous texts compared to Fashion-gen. The number of clusters used for k-means clustering was set to 10, since the dataset is broadly classified into 10 categories.

As comparative methods, we used: (1) the CLIP model released by OpenAI [RAD21], (2) the CLIP model fine-tuned only with correct pairs, (3) the CLIP model fine-tuned with both correct and incorrect pairs (incorrect pairs were randomly selected from the DB without weighting w_k^p, w_k^n during training), (4) the CLIP model fine-tuned with both correct and incorrect pairs (incorrect pairs were generated from those with a similarity of 0.8 or less, without weighting w_k^p, w_k^n during training), (5) the CLIP model fine-tuned with both correct and incorrect pairs (incorrect pairs were generated from those with a similarity of 0.2 or less, without weighting w_k^p, w_k^n during training), (6) the CLIP model fine-tuned with both correct and incorrect pairs (incorrect pairs were generated from those with a similarity of 0.2 or less, with weighting w_k^p, w_k^n during training), (7) EI-CLIP (results cited from the paper [MA22]), and (8) OpenFashionCLIP [CAR23].

4.2 Quantitative experimental results

Table 1 lists the experimental results, with the vertical axis representing the evaluation metrics and the horizontal axis representing the methods, with the highest values for each metric highlighted in red. First, as a result common to both datasets, (1) had low generality for the data, resulting in low scores for all metrics, and (2) showed improved mAP scores compared to (1), but Rank@1 was low. For (3) and (4), there were cases where the images in the correct and incorrect pairs were similar, and there was no improvement in accuracy, but for (5) and (6), where the images in the correct and incorrect pairs were not similar, an improvement in mAP was observed. The proposed method generally showed high scores for Rank@1 and mAP, indicating that the clustering of image features and text features based on incorrect pairs is effective for learning. Additionally, adjusting the learning weights based on the similarity of images in correct and incorrect pairs, enhancing the similarity of correct pairs, and reducing the similarity of incorrect pairs all contributed to a clearer separation in the feature space and improved the image identification performance. As for (7), it was suggested in the original paper [MA22] that accuracy may decrease in cases without metadata or with high ambiguity of text, and it

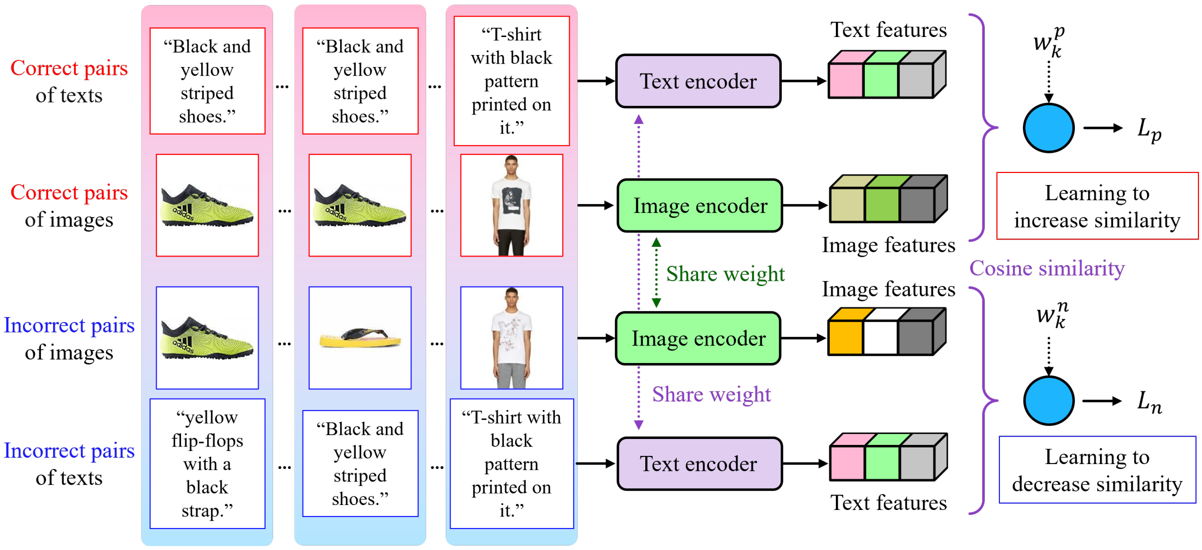

Correct pairs of texts
Correct pairs of images
Incorrect pairs of images
Incorrect pairs of texts
"Black and yellow striped shoes."
"T-shirt with black pattern printed on it."
"yellow flip-flops with a black strap."
Text encoder
Image encoder
Share weight
Text features
Image features
Cosine similarity
w_k^p
L_p
Learning to increase similarity
w_k^n
L_n
Learning to decrease similarity

Figure 5: Learning flow.

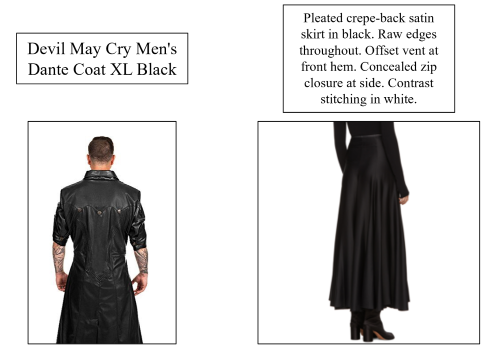
Fashion-gen dataset

	(1)	(2)	(3)	(4)	(5)	(6)	Ours	(7)	(8)
Rank@1	44.0	60.0	47.0	65.0	64.0	65.0	71.0	40.0	54.0
Rank@5	81.0	93.0	87.0	92.0	95.0	96.0	95.0	71.0	83.0
Rank@10	94.0	100	95.0	99.0	99.0	100	97.0	84.0	88.0
mAP	0.60	0.73	0.64	0.77	0.78	0.78	0.81		0.65

Amazon Review Data 2018 dataset

	(1)	(2)	(3)	(4)	(5)	(6)	Ours	(7)	(8)
Rank@1	59.0	59.0	62.0	64.0	67.0	71.0	72.0	23.7	56.0
Rank@5	80.0	81.0	83.0	83.0	82.0	84.0	84.0	49.4	78.0
Rank@10	85.0	89.0	89.0	89.0	87.0	89.0	90.0	61.6	84.0
mAP	0.67	0.68	0.70	0.72	0.73	0.77	0.78		0.66

Table 1: Results using Fashion-gen dataset (above) and results using Amazon Review Data 2018 dataset (below).


Devil May Cry Men's Dante Coat XL Black
Pleated crepe-back satin skirt in black. Raw edges throughout. Offset vent at front hem. Concealed zip closure at side. Contrast stitching in white.

(a) Example image and text from Amazon Review Data 2018 dataset
(b) Example image and text from Fashion-Gen dataset

Figure 6: Example images and texts from datasets.

was observed that Rank@1 was low for the Amazon review data, which is composed of highly ambiguous text. (8) was considered to have not high identification accuracy due to the potential loss of the model's generalization capability for the data, which could be impaired by prompt engineering dependent on the method.

4.3 Qualitative experimental results

To qualitatively evaluate the image identification performance of the proposed method, the top 10 search images for each method were output in response to text queries. The dataset used was Digikala Products Color Classification [MAS21], which includes various similarly colored and shaped product images for online shopping, created for product identification. It consists solely of images, with no correct texts provided; therefore, the determination of whether the output images for the text queries were correct was based on subjective judgment and the judgment of ChatGPT-4. The comparative methods were all the same ones as used in Section 4.2 except for (7).

Figure 7 shows the results, with the vertical axis representing each method and the horizontal axis representing the output image. The red box in the image shows the image judged to be the correct image by ChatGPT-4 and subjectivity, and the black box shows the image not judged to be correct by ChatGPT-4. In this experiment, the text "Black and white striped shoes." was input and

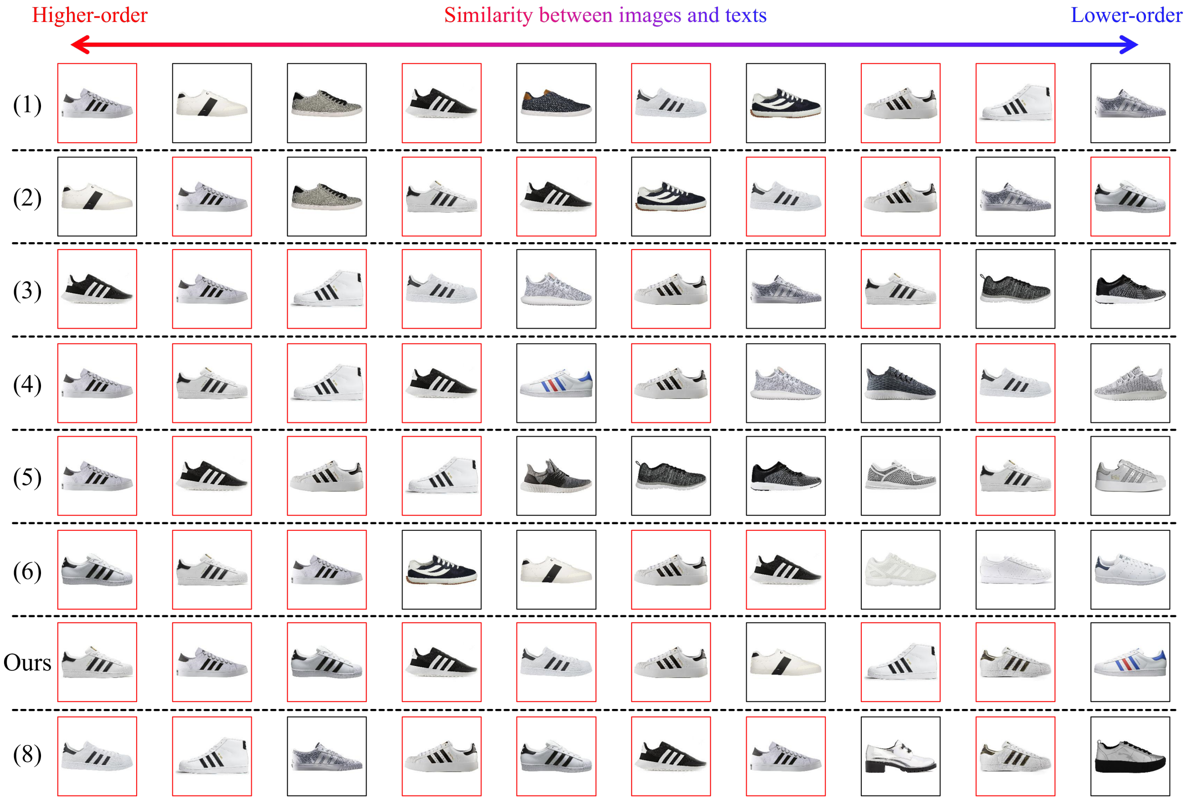


Figure 7: Output results (top 10 images) when the text “Black and white striped sneaker." was input to each method.

the image was output. The experimental results suggest that the image identification performance was low for (1) and (2) because the training may have used incorrect pairs in which the image and text were highly related. In the case of (3) and (4), where the correct pair images may be similar or identical to the incorrect pair images, the training was not effective because the similarity between the image and text was trained to be low, even though the images and text were related. In the case of (5) and (6), the images with low similarity between the incorrect and correct pairs were selected, so we conclude that the intended image for the input text was output. In particular, since the proposed method clusters images and text independently, it is easy to understand the structure of the entire dataset and to select pairs that are not particularly similar, and as a result, we believe that the intended images can be output.

5 CONCLUSION

In this work, we proposed an incorrect pair generation method based on image clustering and demonstrated through experiments that it achieves a higher search accuracy compared to other incorrect pair generation methods and image search techniques using CLIP. By generating suitable incorrect pairs from the clustering results of image and text features and learning from them, the image identification capability was enhanced. Experimental results from the Amazon Review Data 2018 dataset, a commonly used dataset in this field, showed a 27.0% improvement in Rank@1 score compared to vanilla CLIP, and a 11.0% improvement compared to a random reduction of incorrect pairs. An image search system utilizing the proposed method would be able to save users the trouble of filtering out undesired products, thus improving usability and potentially enhancing the purchasing effect in online shopping. The proposed method only generates incorrect pairs of images and text that are too poorly related, which may lead to over-learning. In the future, we plan to increase the accuracy of image retrieval by learning the relationships among detailed features of objects from the relationships among image texts at a rough category level by increasing the learning hierarchy of the CLIP model. We will also expand the comparison method, encoders used, and datasets to further demonstrate the effectiveness of the proposed method.

6 REFERENCES

[SIG08] B. Sigurbjornsson and R. Van Zwol, “Flickr tag recommendation based on collective knowledge", In Proceedings of the 17th International World Wide Web Conference, pp. 327–336, Apr. 2008.

[KIL16] M. Kilickaya, A. Erdem, N. Ikizler-Cinbis, and E. Erdem, “Re-evaluating automatic metrics for image captioning", arXiv preprint arXiv:1612.07600., 2016.

[GIA15] S. Giannoulakis and N. Tsapatsoulis, “Instagram hashtags as image annotation metadata", In Artificial Intelligence Applications and Innovations: 11th IFIP WG 12.5 International Conference, AIAI 2015, Proceedings 11, pp.206–220, Springer International Publishing, Sep. 2015.

[QI20] D. Qi, L. Su, J. Song, E. Cui, T. Bharti and A. Sacheti, “Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data", arXiv preprint arXiv:2001.07966, 2020.

[LI23] J. Li, D. Li, S. Savarese, and S. Hoi, “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models", In International conference on machine learning, pp.19730–19742, PMLR, 2023.

[JIA21] C. Jia, Y. Yang, Y. Xia, Y. T. Chen, Z. Parekh, H. Pham, et al., “Scaling up visual and vision-language representation learning with noisy text supervision", In International conference on machine learning, pp.4904–4916, PMLR, 2021.

[RAD21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, ... and I. Sutskever, “Learning transferable visual models from natural language supervision", In International Conference on Machine Learning, pp.8748–8763, PMLR, July 2021.

[BAL22] A. Baldrati, M. Bertini, T. Uricchio, and A. Del Bimbo, “Effective conditioned and composed image retrieval combining clip-based features", In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.21466–21474, 2022.

[HEN22] M. Hendriksen, M. Bleeker, S. Vakulenko, N. van Noord, E. Kuiper and M. de Rijke, “Extending CLIP for Category-to-image Retrieval in E-commerce", In European Conference on Information Retrieval, pp.289–303, April 2022.

[AGA21] S. Agarwal, G. Krueger, J. Clark, A. Radford, J. W. Kim, and M. Brundage, “Evaluating CLIP: Towards characterization of broader capabilities and downstream implications", arXiv preprint arXiv:2108.02818., 2021.

[SHA23] J. J. Shao, J. X. Shi, X. W. Yang, L. Z. Guo, and Y. F. Li, “Investigating the limitation of clip models: The worst-performing categories", arXiv preprint arXiv:2310.03324., 2023.

[DOS20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16×16 words: Transformers for image recognition at scale", arXiv preprint arXiv:2010.11929, 2020.

[DEV18] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding", arXiv preprint arXiv:1810.04805, 2018.

[SHI22] W. Shin, J. Park, T. Woo, Y. Cho, K. Oh, and H. Song, “e-clip: Large-scale vision-language representation learning in e-commerce", In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pp.3484–3494, Oct. 2022.

[TAN19] Y. Tang, F. Borisyuk, S. Malreddy, Y. Li, Y. Liu, and S. Kirshner, “MSURU: Large scale e-commerce image classification with weakly supervised search data", In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp.2518–2526, July 2019.

[MA22] H. Ma, H. Zhao, Z. Lin, a. Kale, Z. Wang, T. Yu, et al., “Ei-clip: Entity-aware interventional contrastive learning for e-commerce cross-modal retrieval", In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.18051–18061, 2022.

[CAR23] G. Cartella, A. Baldrati, D. Morelli, M. Cornia., M. Bertini, and R. Cucchiara, “OpenFashionCLIP: Vision-and-Language Contrastive Learning with Open-Source Fashion Data", In International Conference on Image Analysis and Processing, pp.245–256, Sep. 2023.

[XIE23] C. W. Xie, S. Sun, X. Xiong., Y. Zheng, D. Zhao, and J. Zhou, “Ra-clip: Retrieval augmented contrastive language-image pre-training", In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.19265–19274, 2023.

[MAC67] J. MacQueen, “Some methods for classification and analysis of multivariate observations", In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol.1, No.14, pp.281-297, 1967.

[ASK21] P. M. A. Kumar, T. S. M. Rao, L. A. Raj, and E. Pugazhendi, “An efficient text-based image retrieval using natural language processing (NLP) techniques", In Intelligent System Design: Proceedings of Intelligent System Design: INDIA 2019, pp.505–519, 2021.

[GAO20] D. Gao, L. Jin, B. Chen, M. Qiu, P. Li, Y. Wei, ... and H. Wang, “Fashionbert: Text and image matching with adaptive loss for cross-modal retrieval", In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.2251–2260, July 2020.

[ROS18] N. Rostamzadeh, S. Hosseini, T. Boquet, W. Stokowiec, Y. Zhang, C. Jauvin, and C. Pal, “Fashion-gen: The generative fashion dataset and challenge", arXiv preprint arXiv:1806.08317, 2018.

[NI19] J. Ni, J. Li, and J. McAuley, “Justifying recommendations using distantly-labeled reviews and fine-grained aspects", In Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp.188–197, 2019.

[MAS21] masouduut94, 2021, “Digikala Products Color Classification | Kaggle", https://www.kaggle.com/datasets/masouduut94/digikala-color-classification/data

Color Quality Comparison in Spectrally (*Un*)Correlated Random Field Models

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

ABSTRACT

We inspect the ability to reproduce spectral (color) composition in random field-based texture models, test when it can neglect spectral correlation, and simplify these random models without visibly depreciating their visual quality. These probabilistic models present essential two or three-dimensional factors for modeling seven-dimensional Bidirectional Texture Function (BTF) - the most advanced representation currently used in real-world material visual properties modeling. They can seamlessly approximate original measured massive data and extend them to an arbitrary size or simulate unmeasured ones. Using extensive test data sets and a small set of setup control parameters, these models reach a vast compression ratio while maintaining the visual quality of measurements, and thus, they are the only viable alternative for BTF practical usage.

Keywords

Texture Analysis, Texture Synthesis, Texture Color Quality Comparison, Texture Modeling, BTF.

1 INTRODUCTION

Enhancing Photorealism in Virtual Reality (VR) scenes necessitates the meticulous application of textures that accurately depict natural material surfaces, ensuring a seamless fusion of visual elements. In the realm of computer graphics, achieving lifelike material appearances relies on visual textures, conceptualized as manifestations of a mathematically-defined random field (RF) possessing spatially uniform attributes. This texture model manifests as a discrete RF, comprising random vectors predominantly arranged on a rectangular lattice grid. The vector space's dimensionality corresponds to the spectral planes within the texture, delineating the richness and complexity of the visual representation.

Real-world materials exhibit intricate physical characteristics, with their micro-structures intricately influencing both reflectance and overall visual presentation. While imperceptible to the naked eye, these micro-structures significantly impact how materials reflect light under various viewing and illumination conditions. The most advanced representation currently available for quantifying and modeling these complexities is the Bidirectional Texture Function (BTF), introduced in [Dan97a]. BTF data serves as the most precise digital emulation of real-world material visual properties, offering analytical insights that elude alternative, less comprehensive visual measurement techniques. The BTF is a seven-dimensional function that encompasses four parameters related to illumination and viewing angles, including azimuthal and elevation angles, one parameter indexing spectral channels, and two parameters representing planar coordinates. This comprehensive model accurately preserves all visual effects inherent to natural materials, including self-occlusions, self-shadowing, inter-reflections, and subsurface scattering, ensuring a faithful representation of real-world surfaces.

Numerous applications (automotive, airspace, safety, architecture, interior design, entertainment industry, movies, computer games, advertising, material recognition) [Hai06a, Sca09a, Vac09a, Vac10a] require to analyze or visualize real-world material visual properties. BTF data are the most advanced and accurate digital representation of real-world material visual properties to date, and their analysis provides information about the measured materials that mainly cannot be attained using any alternative visual measure-

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.

https://www.doi.org/10.24132/CSRN.3401.21

ments or representations. Results of multidimensional textures-related research are also applicable to medical applications [Cul04a, Hai07a, Hai08b, Gri09a] and varied image processing problems, such as image restoration [Hai02a], cultural heritage preservation [Mal01a, Has11a]. Moreover, psychophysical studies of these data [Fil08a] have shown that analyzing of different BTF samples can help understand human perception of real-world materials.

BTF is represented by thousands of given material surface images taken in different combinations of light sources and observation positions during measurement. BTF data size can reach up to several terabytes [HFV12a] even for a limited number of combinations of illumination and viewing angles and small planar size of the measured material, usually several square centimeters [Hai13a]. These restrictions exclude the direct use of measured BTF in applications, and accordingly, some compression and enlargement are necessary. Two enlargement alternatives exist - sampling methods and mathematical models. Sampling methods require storing parts of the original measurements, cannot avoid seams, and offer a minimal compression ratio. Mathematical models generate unlimited enlarged versions of the original texture directly from their small fixed set of parameters, thus offering extreme data compression without seams but may compromise visual quality.

Such quality compromise is hard to express as fully automatic texture quality assessment and mutual similarity evaluation of two or more of them present a significant but complex problem that needs to be solved. Validation of the state-of-the-art texture fidelity criteria [Hai14a] based on the online benchmark [1] demonstrated that none of already published ones, i.e., CW-SSIM [Wan09a], (STSIM-1, STSIM-2, STSIM-M) [Zuj13a] can be reliably used for such task at all. Psycho-physical evaluations are a trustworthy alternative, but they are extremely impractical, expensive, and generally demanding. A pressing need is for a reliable criterion to support texture model development, which would be able to compare the original texture with synthesized or reconstructed ones and identify the best result and, therefore, corresponding optimal model parameter settings for specific models. Such criterion also plays a vital role in efficient content-based image retrieval, e.g., from digital libraries or multimedia databases.

This paper answers whether we can model a BTF texture using mathematical models on spectrally decorrelated data. We compare random fields-based texture models using their spectral quality, i.e., we investigate how individual models can represent color information of the original data. We propose a novel criterion that allows us to predict when monospectral decorrelated channels can be modeled by a set of simpler 2D random fields and when fully spectrally correlated 3D models with more than $no\_spectral\_channels\times$ parameters are required compared to 2D models.

2 BTF RANDOM FIELD MODELS

The size of BTF data prohibits its direct integration into graphic applications, necessitating compression for practical usage. Additionally, BTF data is typically acquired under a limited set of illumination and viewing conditions, mandating reconstruction of the BTF space for real-world application. Furthermore, given the relatively small planar size of the measured sample from which BTF data originates, seamless texture enlargement becomes imperative. These essential considerations collectively fall under the umbrella of BTF modeling, which encompasses techniques to address the challenges posed by data size, reconstruction, and texture enlargement, ensuring effective utilization of BTF data in graphics applications. In addition to probabilistic BTF models, there is an alternative approach to approximate BTF data using pixel-wise generalizations of existing BRDF models, known as SVBRDF. However, this method comes with trade-offs, as it sacrifices visual quality by omitting critical features such as self-occlusions, self-shadowing, inter-reflections, and subsurface scattering. Furthermore, it cannot achieve the same level of compression efficiency as fully probabilistic BTF models, which offer significantly higher compression ratios and arbitrary size texture generation without visible discontinuities.

Modeling BTF based on probabilistic models necessitates the utilization of multi-dimensional models. However, such models are not prevalent and encounter various unresolved theoretical challenges, as noted in [Hai13a]. One potential workaround involves spectrally and spatially factorizing the BTF space, enabling its representation through a series of lower-dimensional models. Unfortunately, real data are correlated and can be spectrally factorized only approximately, which can lead to a loss of spectral information.

Using 2D models [Hai23a] for multi-spectral material data requires the input to be spectrally de-correlated using the Karhunen-Loève Transformation (KLT) so that resulting mono-spectral factors can be modeled independently. The 2D - 3D models compression ratio for the measured BTF space in our representation (6 561 measurements) is 1 : 19 683. However, this step leads to an inevitable loss of information. The approximation error is proportional to the extent of the color space contained in the modeled texture.

The original data space $\tilde{Y}$ is transformed into the new one with coordinate axes Y. The new basis consists

---

[1] http://tfa.utia.cas.cz

of the eigenvectors of the second-order statistical moments matrix $\Sigma = E\{\tilde{Y}_r \tilde{Y}_r^T\}$ where r denotes a multi-index $r = (r_1, r_2)$, $r \in I$, with the row and column indices, $\bullet$ all possible values of the corresponding index, I is a finite discrete 2D rectangular $M \times N$ index lattice, and d is the number of spectral bands. The projection of a $d \times 1$ random vector $\tilde{Y}_r$ onto the KLT coordinate system uses transformation matrix T, which consists of eigenvectors of Σ. If the measured data are Gaussian, then the transformed data are independent, and thus, each mono spectral factor can be modelled independently [Hai23a]. Although this assumption generally does not hold, practical results show that it may be taken into account without a significant impact on the visual quality of achieved results.

The texture pixels are defined as intensity values (2D) or intensity vectors (3D) on multiple finite $M \times N$ 2D lattice. The 3D multiindex is $r = \{r_1, r_2, r_3\}$ with spatial (r_1, r_2) and spectral (r_3) indices. For mathematical simplicity, all lattices are defined as double toroidal [Hai13a]. Markovian neighboring lattice locations are the set of relative lattice locations called Contextual Neighbourhood (CN) I_r. The selection of an appropriate CN influences the overall model performance: CN containing too few elements cannot capture all texture details. On the other hand, including the unnecessary elements adds to the computational burden and can degrade the model's performance as an additional noise source.

2.1 2D Causal Auto-Regressive Model

The Causal Auto-Regressive (CAR) RF is a collection of random variables with a joint probability density on the set of all possible realizations Y of the $M \times N$ lattice I [Hai23a]:

$$p(Y|\gamma, \sigma^{-2}) = (2\pi\sigma^2)^{-\frac{(MN-1)}{2}} \qquad (1)$$
$$exp\left\{ -\frac{1}{2} tr \left\{ \sigma^{-2} \begin{pmatrix} -\alpha \\ \gamma^T \end{pmatrix}^T V_{MN-1} \begin{pmatrix} -\alpha \\ \gamma^T \end{pmatrix} \right\} \right\} ,$$

where α is a vector of unities, $\gamma = [a_1, \ldots, a_\eta]$ is a parameter vector, η denotes the cardinality of a causal contextual neighbourhood I_r^C, σ is variance of Gaussian distribution and

$$V_{t-1} = \tilde{V}_{t-1} + V_0 = \left(\begin{array}{cc} \tilde{V}_{yy(t-1)} & \tilde{V}^T_{xy(t-1)} \\ \tilde{V}_{xy(t-1)} & \tilde{V}_{xx(t-1)} \end{array} \right) + V_0 \ , \quad (2)$$

$$\begin{aligned} \tilde{V}_{xx(t-1)} &= \sum_{k=1}^{t-1} X_k X_k^T & \tilde{V}_{xy(t-1)} &= \sum_{k=1}^{t-1} X_k Y_k^T \ , \\ \tilde{V}_{yy(t-1)} &= \sum_{k=1}^{t-1} Y_k Y_k^T & X_k &= [Y_{k-s}^T : \forall s \in I_k^C]^T \ . \end{aligned}$$

Simplified notation $t, t-1, \ldots$ denotes the process position in I, i.e., t is the index of the sequence of multi-indices $((r_1, r_2)_t)_{t=1}^{MN}$. The order of indices $(r_1, r_2) = r$ depends on the order in which the analyzed texture pixels are processed. The data from model history obtained during adaptation are denoted as $Y_{(t-1)}$. For the sake of proper model adaptation, the standard exponential forgetting factor technique can be utilized [Hai23a]. The 2D CAR model can be expressed as a stationary causal uncorrelated noise driven 2D auto-regressive process with correlation structure [Hai23a]:

$$E\{e_r e_s\} = \left\{ \begin{array}{ll} \sigma_i^2 & s = r, \\ 0 & otherwise. \end{array} \right. \qquad (3)$$

Unlike 2D Gaussian Markov random field model, the parameters of 2D CAR model can be estimated analytically without simplifying approximations using the maximum likelihood, the LS or Bayesian methods [Hai02c]. The Bayesian parameter estimations of the model with the normal-Wishart parameter prior which maximizes the posterior density are [Hai23a]:

$$\begin{aligned} \hat{\gamma}^T_{t-1} &= V^{-1}_{xx(t-1)} V_{xy(t-1)} \ , & (4) \\ \hat{\sigma}^2_{t-1} &= \frac{\lambda_{(t-1)}}{\beta(t)} \ , & (5) \\ \lambda_{(t-1)} &= V_{yy(t-1)} - V^T_{xy(t-1)} V^{-1}_{xx(t-1)} V_{xy(t-1)} \ , & (6) \\ \beta(t) &= \beta(0) + t - 1, \ \beta(0) > 1 \ , & (7) \end{aligned}$$

and $V_{xx(0)}$, $V_{xy(0)}$, $V_{yy(0)}$ are from normal-gamma parameter prior. The 2D CAR synthesis can be easily performed [Hai23a].

2.2 3D Causal Auto-Regressive Model

The 3D CAR RF is a collection of random variables with a joint probability density on the set of all possible realizations Y of the $M \times N \times d$ lattice I, subject to the condition [Hai23a]:

$$p(Y|\Theta, \Sigma^{-1}) = 2\pi^{-\frac{d(MN-1)}{2}} |\Sigma^{-1}|^{\frac{MN-1}{2}} \qquad (8)$$
$$exp\left\{ -\frac{1}{2} tr \left\{ \Sigma^{-1} \begin{pmatrix} -I \\ \Theta^T \end{pmatrix}^T V_{MN-1} \begin{pmatrix} -I \\ \Theta^T \end{pmatrix} \right\} \right\} ,$$

where I is identity matrix, Σ is covariance matrix of Gaussian distribution and Θ is $d \times d\eta$ parameter matrix $\Theta = (A_1, \ldots, A_\eta)$ where $\forall i \in \{1, \ldots, \eta\}$ [Hai23a]:

$$A_i = \begin{pmatrix} a^i_{1,1} & \cdots & a^i_{1,d} \\ \vdots & \ddots & \vdots \\ a^i_{d,1} & \cdots & a^i_{d,d} \end{pmatrix} . \qquad (9)$$

The 3D CAR model can be expressed as a stationary causal uncorrelated noise driven 3D auto-regressive process [Hai23a]:

$$Y_r = \Theta X_r + e_r \ , \qquad (10)$$

with correlation structure [Hai13a]:

$$E\{e_r e_s\} = \begin{cases} \Sigma & s = r, \\ 0 & otherwise, \end{cases} \tag{11}$$

The estimates of model parameters are [Hai23a]:

$$\hat{\Theta}^T_{t-1} = V^{-1}_{xx(t-1)} V_{xy(t-1)} \ , \tag{12}$$

$$\hat{\Sigma}_{t-1} = \frac{\lambda_{(t-1)}}{\beta(t)} \ . \tag{13}$$

where $V_{xx(0)}$, $V_{xy(0)}$, $V_{yy(0)}$ are from normal-Wishart parameter prior. The model can be directly generated from the model equation (10).

2.3 Moving Average Model

2D MA texture model assumes that the modeled mono spectral factor is an output of a specific underlying system that completely characterizes it in response to a 2D uncorrelated random input, and that is a sample from 2D RF defined on an infinite 2D lattice. The impulse response of a linear 2D filter can represent this system. Therefore, a convolution of an uncorrelated 2D RF can generate the mono spectral factor with this 2D filter.

Supposing that the mono spectral factor Y is the output of the underlying linear system which completely characterizes it in response to the 2D uncorrelated random input e_r, then $Y_{r,i}$ is determined by the following difference equation [Li92a]:

$$Y_{r,i} = \sum_{s \in I_r} a_{s,i} \, e_{r-s,i} \ , \tag{14}$$

where a_s are constant coefficients. I_r determines the causality or non-causality of the model.

To fit the model to a given mono spectral factor Y, the parameters a_s have to be estimated. This may be done using a method [Li92a] similar to the one-dimensional (1D) Random Decrement Technique [Col73a]. The estimation procedure begins by arbitrarily selecting a threshold ξ usually chosen as some percentage of the standard deviation of the intensity values of the mono spectral factor. Generally, higher value of ξ leads to the synthesized textures with higher contrast and vice versa. Unfortunately, there does not exist any method for an automatic determination of the optimal value of ξ, i.e., the value at which the result that is visually the most similar to the original is achieved.

During the model parameter estimation process every pixel of the analyzed mono spectral factor is examined. If the intensity value of the examined pixel Y_r is higher than the value of the threshold ξ and among the four adjacent pixels to the pixel Y_r at least one in the same row and at least one in the same column are lower than ξ, i.e., $Y_{(r_1,r_2)} > \xi$ and ($Y_{(r_1,r_2-1)} < \xi$ or $Y_{(r_1,r_2+1)} < \xi$ and $Y_{(r_1-1,r_2)} < \xi$ or $Y_{(r_1+1,r_2)} < \xi$) holds then the pixel Y_r is referred to as significant. For each significant pixel Y_r, the intensity values of pixels whose pixel-relative position to Y_r is defined by I_r form a vector Υ of length which equals the number of elements of I_r. It is assumed that there is a unambiguous correspondence between components of Υ, parameters a_s and the elements of I_r. Vectors Υ_υ of all significant pixels of Y are summed and divided by the number of significant pixels (g) to obtain the estimate of the model parameters, i.e., [Li92a]:

$$\hat{a}_s = \frac{1}{g} \sum_{\upsilon=1}^{g} \Upsilon_\upsilon \ . \tag{15}$$

The model assumes that the modelled texture is the RF realization with zero mean therefore, it is necessary to estimate the mean value of the pixel intensity levels. Estimated mean value is saved together with estimated model parameters for the texture synthesis purposes.

2D MA texture model is able to generate synthetic mono spectral factor of arbitrary size from the model parameters a_s according to the model equation (14). It has been proven that the result of the synthesis closely approximates the first and second order statistics of the original when e_r is the white noise [Li92a].

2.4 Extended Moving Average Model

It is possible to extend the approach described in Sec. 2.3 to obtain 3D MA texture model as suggested in [Hav15a]. Using a 3D model avoids spectral de-correlation of analyzed multi-spectral texture with possible loss of certain portion of the information which is a certain advantage over the 2D MA texture model. The 3D MA texture model assumes that the input is a stochastic multi-spectral texture and the output of certain underlying system which completely characterizes it in response to a 3D uncorrelated random input $e_{r,i}$. 3D MA model equation represents an extended version of (14) allowing simultaneous modelling of all texture spectral planes. $Y_{r,i}$ is determined by the following difference equation [Hav15a]:

$$Y_{r,i} = \sum_{j=1}^{d} \sum_{s \in I_r} a_{s,i,j} \, e_{r-s,i} \ , \tag{16}$$

where $a_{s,i,j}$ are constant coefficients. The geometry of I_r determines the causality or non-causality of the model.

Parameters $a_{s,i,j}$ have to be estimated to fit the model equation (16) to certain multi-spectral texture Y performing extended variant of the method used for 2D MA texture model parameter estimation. The estimation procedure begins by selecting thresholds ξ_k, $k \in$

$\{1,\dots,d\}$, usually chosen as some percentage of the standard deviation of the pixel intensity values of the corresponding spectral planes. Generally, higher values of ξ_k leads to the synthesized textures with higher contrast and vice versa. For simplicity and possible comparison with 2D MA, all ξ_k were set equal and denoted ξ there. There does not exist any method for an automatic determination of the optimal values of ξ_k, i.e., the value at which the result that is visually the most similar to the original is achieved.

During the model parameter estimation process every pixel of the analyzed multi-spectral texture is examined. Individual parameters $a_{s,i,j}$ are estimated independently but at the same time. If $Y_{(r_1,r_2),i} > \xi_i$ and ($Y_{(r_1,r_2-1),j} < \xi_j$ or $Y_{(r_1,r_2+1),j} < \xi_j$ and $Y_{(r_1-1,r_2),j} < \xi_j$ or $Y_{(r_1+1,r_2),j} < \xi_j$) holds then the pixel Y_r is referred to as significant for $a_{s,i,j}$. For each significant pixel Y_r for $a_{s,i,j}$, the intensity values in j-th spectral plane of pixels whose pixel-relative position to Y_r is defined by I_r form a vector Υ^{ij} of length which equals the number of elements of I_r. It is assumed that there is a unambiguous correspondence between components of Υ^{ij}, parameters $a_{s,i,j}$ and the elements of I_r. Vectors Υ^{ij}_{υ} of all pixels significant for $a_{s,i,j}$ are summed and divided by their number (denoted as g^{ij}) to obtain the estimate of the model parameters, i.e., [Hav15a]:

$$\hat{a}_{s,i,j} = \frac{1}{g^{ij}} \sum_{\upsilon=1}^{g^{ij}} \Upsilon^{ij}_{\upsilon} \ . \tag{17}$$

The model assumes that the modelled multi-spectral texture is the RF realization with zero mean therefore, it is necessary to estimate the mean value of the pixel intensity levels of individual spectral planes.

The 3D MA model is able to generate synthetic multi-spectral texture of arbitrary size from the model parameters $a_{s,i,j}$ according to (16).

2.5 Multi-Spectral Simultaneous Auto-Regressive Model

The Multi-Spectral Simultaneous Auto-Regressive (MSAR) model [Hai12a] is based on the (MSAR) texture model [Ben98a]. The MSAR model can be expressed as a stationary, non-causal correlated noise-driven 3D auto-regressive process [Ben98a]:

$$Y_{r,i} = \sum_{j=1}^{d} \sum_{s \in I_r^{ij}} a_{s,i,j} Y_{r \oplus s,j} + \sqrt{\sigma_i}\varepsilon_{r,i} \ , \tag{18}$$

where I_r^{ij} denotes the CN relating intensity values in the i-th spectral plane to the neighbouring ones in the j-th ($j \in \{1,\dots,d\}$) spectral plane, $a_{s,i,j}$ are the corresponding parameters which define the dependence of $Y_{r,i}$ on its neighbours defined by I_r^{ij}. Symbol $\oplus$ denotes modulo addition in each index of the multi-indices r, i.e., modulo addition of M for r_1 and modulo addition of M for r_2 and $s = (s_1, s_2)$. The driving noise $\varepsilon_{r,i}$ are i.i.d. random variables distributed normally with zero mean and constant but unknown variance σ_i. Rewriting (18) in matrix form the MSAR model equations become [Ben98a]:

$$\Psi Y = \varepsilon \ , \tag{19}$$

$$\Psi = \begin{pmatrix} \Psi_{11} & \Psi_{12} & \dots & \Psi_{1d} \\ \Psi_{21} & \Psi_{22} & \dots & \Psi_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_{d1} & \Psi_{d2} & \dots & \Psi_{dd} \end{pmatrix} , \tag{20}$$

$$Y = \{Y_1, Y_2, \dots, Y_d\}^T \ , \tag{21}$$

$$\varepsilon = \{\sqrt{\sigma_1}\varepsilon_1, \sqrt{\sigma_2}\varepsilon_2, \dots, \sqrt{\sigma_d}\varepsilon_d\}^T \ , \tag{22}$$

where both Y_i and ε_i are vectors of length $M \times N$ of lexicographic ordered arrays of $Y_{r,i}$ and $\varepsilon_{r,i}$, respectively, i.e., $Y_i = \{Y_{(1,1),i}, Y_{(1,2),i}, \dots, Y_{(1,M),i}, Y_{(2,1),i}, Y_{(2,2),i}, \dots, Y_{(M,M),i}\}$ and similarly for ε_i. The transformation matrix Ψ is composed of $M^2 \times M^2$ block circulant sub-matrices [Ben98a]:

$$\Psi ij = \begin{pmatrix} \Psi^1_{ij} & \Psi^2_{ij} & \dots & \Psi^M_{ij} \\ \Psi^M_{ij} & \Psi^1_{ij} & \dots & \Psi^{M-1}_{ij} \\ \vdots & \vdots & \ddots & \vdots \\ \Psi^2_{ij} & \Psi^3_{ij} & \dots & \Psi^1_{ij} \end{pmatrix} , \tag{23}$$

where each element Ψ^k_{ij}, $k \in \{1,\dots,M\}$, is an $M \times M$ circulant matrix whose (m,n)-th element is given by [Ben98a]:

$$\Psi^k_{i,j}(m,n) = \begin{cases} 1, & i = j, m = n, k = 1, \\ -a_{s,i,j}, & s_1 = k-1, \\ & s_2 = ((n-m) \bmod M), \\ & (s_1, s_2) \in I_r^{ij}, \\ 0, & otherwise. \end{cases} \tag{24}$$

Rewriting (19) as $Y = \Psi^{-1}\varepsilon$, the covariance matrix is obtained as [Ben98a]:

$$\Sigma_Y = E\{\ \Psi^{-1}\ \varepsilon\varepsilon^T\ \Psi^{-T}\ \} = \Psi^{-1}\Sigma_\varepsilon\Psi^{-T} \tag{25}$$

$$\Sigma_\varepsilon = E\{\varepsilon\varepsilon^T\} = \begin{pmatrix} \sigma_1 I & 0 & \dots & 0 \\ 0 & \sigma_2 I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_d I \end{pmatrix} \tag{26}$$

where I is the identity matrix. The positive definite property required of Σ_Y is ensured if Ψ is nonsingular. MSAR model parameters can be estimated using the Least Square (LS) approach. The estimate of the MSAR model parameters is obtained by equating the intensity values in individual pixel spectral planes of the analyzed texture to the expected ones predicted by the model, leading to the independent systems of equations [Ben98a]:

$$Y_{r,i} = E\{Y_{r,i}|\gamma_i\} = X_{r,i}^T\gamma_i \ , \tag{27}$$

$$\gamma_i = [\gamma_{i1}, \gamma_{i2}, \dots, \gamma_{id}]^T \ , \tag{28}$$

$$X_{r,i} = \Big[\{Y_{r\oplus s,1} : s \in I_r^{i1}\}, \dots, \{Y_{r\oplus s,d} : s \in I_r^{id}\}\Big]^T \tag{29}$$

where $\gamma_{ij} = [a_{s,i,j} : \forall s \in I_r^{ij}]$. The LS solution $\hat{\gamma}_i$ and $\hat{\sigma}_i$ can be found as [Ben98a]:

$$\hat{\gamma}_i = \left(\sum_{r\in I} X_{r,i}X_{r,i}^T\right)^{-1}\left(\sum_{r\in I} X_{r,i}Y_{r,i}\right) \ , \tag{30}$$

$$\hat{\sigma}_i = \frac{1}{M^2}\sum_{r\in I}(Y_{r,i} - \hat{\gamma}_i{}^T X_{r,i})^2 \ . \tag{31}$$

The model assumes that the modeled texture is the RF realization with zero means; therefore, estimating the mean value of the pixel intensity levels of individual spectral planes is necessary. There are several possibilities existing for the MSAR texture model synthesis. Considering the double toroidal boundary conditions, the Discrete Fast Fourier Transform (DFFT) is the most effective method. The MSAR model equations (18) may be expressed in terms of the Discrete Fourier Transform (DFT) of each spectral plane as [Ben98a]:

$$\tilde{Y}_{t,i} = \sum_{j=1}^{d}\sum_{s\in I_r^{ij}} a_{s,i,j}\tilde{Y}_{t,j}e^{\sqrt{-1}\omega_{st}} + \sqrt{\sigma_i}\tilde{\varepsilon}_{t,i} \ , \tag{32}$$

where $\tilde{Y}_{t,i}$ and $\tilde{\varepsilon}_{t,i}$ are the 2D DFT coefficients of $Y_{r,i}$ and $\varepsilon_{r,i}$, respectively, at the discrete frequency index $t = (m,n)$ and $\omega_{st} = \frac{2\pi(ms_1+ns_2)}{M}$. The equations (32) can be written in matrix form as [Ben98a]:

$$\tilde{Y}_t = \Lambda_t^{-1}\,\Sigma^{\frac{1}{2}}\,\tilde{\varepsilon}_t, \ , \tag{33}$$

$$\tilde{Y}_t = \{\tilde{Y}_{t,1}, \tilde{Y}_{t,2}, \dots, \tilde{Y}_{t,d}\}^T \ , \tag{34}$$

$$\tilde{\varepsilon}_t = \{\tilde{\varepsilon}_{t,1}, \tilde{\varepsilon}_{t,2}, \dots, \tilde{\varepsilon}_{t,d}\}^T \ , \tag{35}$$

where the matrices Σ and Λ_t are defined as [Ben98a]:

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_d \end{pmatrix} , \tag{36}$$

$$\Lambda_t = \begin{pmatrix} \lambda_{t,11} & \lambda_{t,12} & \cdots & \lambda_{t,1d} \\ \lambda_{t,21} & \lambda_{t,22} & \cdots & \lambda_{t,2d} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{t,d1} & \lambda_{t,d2} & \cdots & \lambda_{t,dd} \end{pmatrix} , \tag{37}$$

$$\lambda_{t,ij} = \begin{cases} 1 - \sum_{s\in I_r^{ij}} a_{s,i,j}\, e^{\sqrt{-1}\omega_{st}} & i = j \, , \\ -\sum_{s\in I_r^{ij}} a_{s,i,j}\, e^{\sqrt{-1}\omega_{st}} & i \neq j \, . \end{cases} \tag{38}$$

The MSAR model will be stable and valid if Λ_t is a non-singular matrix $for t \in I$. A $M \times M$ texture can be synthesized from the estimated model parameters according to the following algorithm [Ben98a]:

1) Generate d noise 2D arrays $\varepsilon_{r,i}$ using a pseudo random number generator.

2) Calculate the 2D DFFT of each noise array separately.

3) For each discrete frequency index t compute: $\tilde{Y}_t = \Lambda_t^{-1}\,\Sigma^{\frac{1}{2}}\,\tilde{\varepsilon}_t$.

4) Perform the 2D inverse DFFT of each $\tilde{Y}_{t,i}$ separately.

2.6 Multi-Spectral Markov Random Field Model

The BTF Multi-Spectral Markov Random Field (BTF-MMRF) model is based on (MMRF) factor texture model [Ben98a]. A multi-spectral texture can be considered Markovian with respect to I_r^{ij} if it has following property [Ben98a]:

$$\begin{aligned} &p(Y_{r,i}\,|\,Y_{s,j}, \forall j \in \{1,\dots,d\} : j \neq i, \forall s \in I_r^{ij} : s \neq r) \\ &\qquad = p(Y_{r,i}\,|\,Y_{s,j}, \forall j \in \{1,\dots,d\}, \forall s \in I_r^{ij}) \ . \end{aligned} \tag{39}$$

As the conditional distributions of $Y_{r,i}$ given $\{Y_{s,j}, \forall j \in \{1,\dots,d\} : j \neq i, \forall s \in I_r^{ij} : s \neq r\}$ and $Y_{r,i}$ given $\{Y_{s,j}, \forall j \in \{1,\dots,d\}, \forall s \in I_r^{ij}\}$ are the same, the best linear estimator of Y can be written as [Ben98a]:

$$Y_{r,i} = \sum_{j=1}^{d}\sum_{s\in N_r^{ij}} a_{s,i,j}Y_{r\oplus s,j} + \varepsilon_{r,i}, \ , \tag{40}$$

where $a_{s,i,j}$ are taken as the coefficients of the Minimum Mean Square Error (MMSE) estimate of $Y_{r,i}$ and $\varepsilon_{r,i}$ denotes the estimation error. From the orthogonality property of the MMSE estimates [Ben98a]:

$$E\{Y_{r,i}\varepsilon_{s,j}\} = \begin{cases} \sigma_i & i = j, s = r \ , \\ 0 & \text{otherwise} \ , \end{cases} \tag{41}$$

the correlation structure of the stationary noise $\varepsilon_{r,i}$ is [Ben98a]:

$$\Psi_{ij}^s = E\{\varepsilon_{r,i}\varepsilon_{r\oplus s,j}\} = \begin{cases} -a_{s,i,j}\sigma_j & s \in I_r^{ij} \ , \\ \sigma_j & s = 0, i = j \, , \\ 0 & \text{otherwise} \ . \end{cases} \tag{42}$$

Because the correlation functions have the symmetry property $\Psi_{ij}^s = \Psi_{ji}^{-s}$, there is an implicit requirement that I_r^{ij} and the associated coefficients are symmetric, i.e., $s \in I_r^{ij} \iff -s \in I_{-r}^{ji}$ and $\sigma_j\Psi_{ij}^s = \sigma_i\Psi_{ji}^{-s}$. Equation (40) can be rewritten in matrix form as: $\Psi Y = \varepsilon$ and equations (41) and (42) can be expressed using matrix notations as [Ben98a]:

$$\Sigma_\varepsilon = E\{Y\varepsilon^T\} = \begin{pmatrix} \sigma_1 I & 0 & \dots & 0 \\ 0 & \sigma_2 I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_d I \end{pmatrix} \tag{43}$$

$$\Sigma_e = E\{\varepsilon\varepsilon^T\} = E\{\Psi Y\varepsilon^T\} = \Psi\Sigma_\varepsilon \ . \tag{44}$$

The model parameters must be chosen such that Σ_e is positive definite. Using (19) and (44), the MMRF covariance matrix can be found as [Ben98a]:

$$\begin{aligned}\Sigma_Y &= E\{YY^T\} = E\{\Psi^{-1}\varepsilon\varepsilon^T\Psi^{-T}\} \\ &= \Psi^{-1}\Sigma_e\Psi^{-T} = \Psi^{-1}\Sigma_\varepsilon \ . \end{aligned} \tag{45}$$

Matrix Σ_Y will be admissible covariance matrix if all model parameters are chosen such that Σ_Y is positive definite which is consistent with the requirement that matrix Σ_e be positive definite, since $\Sigma_Y = \Sigma_\varepsilon \Sigma_e^{-1} \Sigma_\varepsilon$. MMRF model parameters can be estimated using an approach based on the LS method similar to the MSAR model parameter estimation. Due to the symmetry property of (42), the LS estimates are inherently nonlinear, and it is necessary to solve for all model parameters simultaneously. As in the case of the MSAR model parameter estimation, LS estimates are obtained by equating each pixel spectral intensity value to the expected value of one of the model equations. An iterative approach can be used to obtain the LS solutions [Ben98a]:

$$\widehat{\gamma}_{i,t+1} = \left(\sum_{r \in I} Q_{r,i,t} Q_{r,i,t}^T \right)^{-1} \left(\sum_{r \in I} Q_{r,i,t} Y_{r,i} \right) \ , \tag{46}$$

$$\widehat{\sigma}_{i,t+1} = \frac{1}{M^2} \sum_{r \in I} (Y_{r,i} - Q_{r,i,t}^T \hat{\gamma}_{i,t})^2 \ , \tag{47}$$

$$Q_{r,i,t} = \begin{pmatrix} q_{r,11} & q_{r,12} & \dots & 0 & 0 \\ 0 & \frac{\widehat{\sigma}_{2,t} q_{s,21}}{\widehat{\sigma}_{1,t}} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \frac{\widehat{\sigma}_{d,t} q_{r,dd-1}}{\widehat{\sigma}_{d-1,t}} & q_{r,dd} \end{pmatrix}^T ,$$

$$q_{r,ij} = \begin{cases} \{Y_{(r\oplus s),j} + Y_{(r\oplus -s),j} : s \in N_r^{ij}\} & i = j \,, \\ \{Y_{(r\oplus s),j} : s \in I_r^{ij}\} & i < j \,, \\ \{Y_{(r\oplus -s),j} : s \in I_r^{ij}\} & i > j \,, \end{cases}$$

with $\frac{\widehat{\sigma}_{z,1}}{\widehat{\sigma}_{z-1,1}} = 1, z \in (2,\dots,d)$, where $t \in \{1,\dots,+\infty\}$ denoting the order of the iteration. Although no proof of convergence exists for this procedure, experimental results indicate that a sufficiently accurate estimate is achieved in less than ten iterations. If the results of two consecutive iterations are the same within the achievable accuracy of the rounding on the used system, then such results are considered sufficiently accurate estimates. It should be noted that the LS estimate represents only a very rough approximation of the MMRF model parameter estimate. The algorithm for the synthesis of the MMRF model is identical to that one for the MSAR model, except the calculations in the third step which replaces citeBen97a: $\tilde{Y}_t = \Lambda_t^{-1}(\Lambda_t\Sigma)^{\frac{1}{2}}\tilde{\varepsilon}_t$. The MMRF model will exist and be stable if $\Lambda_t\Sigma$ is positive definite $\forall t \in I$.

2.7 Pseudo Markov Random Field Model

The BTF Pseudo Markov Random Field (BTF-PMRF) model [Hav12a] is based on the (PMRF) texture model [Ben98a]. The PMRF model represents an approximation of the MMRF model. The apparent motivation for deriving this approximation was to avoid an iterative parameter estimation method. The PMRF model was derived from the MMRF model by subtly modifying the correlation structure given in (42). The PMRF and the MMRF model equations are identical with one exception: the PMRF stationary noise $e_{r,i}$ is assumed to have the following correlation structure [Ben98a]:

$$E\{\varepsilon_{r,i}\varepsilon_{s\oplus r,j}\} = \begin{cases} -a_{s,i,j}\sqrt{\sigma_i\sigma_j} & s \in I_r^{ij} \,, \\ \sigma_j & s = 0, i = j \,, \\ 0 & otherwise \,. \end{cases} \tag{48}$$

The main difference between the MMRF model and the PMRF model and the significant advantage of the PMRF model is the fact that the estimate $\widehat{\gamma}$ is linear and independent of the estimate $\widehat{\sigma}$. The LS estimates of the PMRF model parameters are the same as those of the MMRF model, for $t = 1$. Therefore, the model parameter estimation does not require an iterative process, unlike the case of the MMRF model parameter estimation, which reduces the computational burden. Employing the same method used for the MMRF model synthesis in the case of the PMRF model, the calculations in the third step of the synthesis procedure become [Ben98a]: $\tilde{Y}_t = \Lambda_t^{-1}\Sigma^{\frac{1}{2}}\Lambda_t^{\frac{1}{2}}\tilde{\varepsilon}_t$. The PMRF model will exist and be stable if Λ_t is positive definite $\forall t \in I$.

3 COLOR QUALITY CRITERIA

3.1 Spectral Decorrelation Criterion

$$\kappa(c_{\max}) = \frac{1}{c_{\max}^n} \det(\Sigma) \ , \tag{49}$$

where Σ is a $n \times n$ material texture spectral covariance matrix, $c_{\max}$ is the maximal possible spectral value per channel. If the criterion $\kappa(c_{\max}) \leq 3$, we can replace a 3D random field model with its Karhunen-Loeve decorrelated version and model each decorrelated single spectral band with a 2D random field model without significant MEMD error (50), i.e., with negligible color loss.

3.2 Color Composition Comparison

The Mean Exhaustive Minimum Distance (MEMD) [Hav19a] is used to compare the spectral composition of two textures and the cardinalities of the same colors but ignoring the locations of individual pixels. The comparison is performed by individually taking pixels from the first image and searching for the most similar, i.e., the closest in certain vector metric sense, to the ones in the second image. The pixel from the second image identified as the most similar is removed from the stack representing the second image, and the local spectral error is recorded. The MEMD criterion is as follows:

$$\zeta(A,B) = \frac{1}{M} \sum_{(r_1,r_2)\in\langle A\rangle} \min_{(\acute{r}_1,\acute{r}_2)\in N} \left\{ \rho\left(Y^A_{r_1,r_2,\bullet}, Y^B_{\acute{r}_1,\acute{r}_2,\bullet}\right)\right\} \ , \tag{50}$$

where $M = \min\{\sharp\{A\},\sharp\{B\}\}$, $\sharp\{A\}$ is the number of pixels in A and similarly for $\sharp\{B\}$, $\min\{\emptyset\} = 0$, (r_1,r_2) denotes the location in A, $\langle A\rangle$ represents the set of all pixel indices of A, $(\acute{r}_1,\acute{r}_2)$ is the location in B, N is the set of unprocessed pixel indices of B, ρ is an arbitrary vector metric and $Y^A_{r_1,r_2,\bullet}$ represents the pixel at (r_1,r_2) in A, where $\bullet$ denotes all corresponding spectral indices, similarly for $Y^B_{\acute{r}_1,\acute{r}_2,\bullet}$. The term $\zeta(A,B)$ is evaluated using raster scanning of A. The algorithm stops when all pixels of A are scanned, or N becomes an empty set (see details in [Hav19a, Hav21a, Hav23a]).

4 TEST DATA

We used BTF MAM2014 Dataset [2] [Fil18a]. This dataset consists of 16 BTF of materials presented at the Workshop on Material Appearance Modelling (MAM) 2014. The BTF datasets consist of a collection of material images taken for lighting and viewing directions with an angular sampling of 81 × 81. The uncompressed HDR data (32-bit float) are represented in CIE XYZ colorspace. The spatial resolution of the datasets is 353.3 dpi. They included widely different natural, artificial, matte or glossy materials are mica, sand-fine, sand-coarse, burlap, cork, towel, green cloth, green felt, basketball, flocked paper, silver gold, brown tile, glass tile, blue-black-gold tile, crinkle paper, and basketweave. This texture set is illustrated in Fig. 1.

5 RESULTS

We used the same setup for all tested models, i.e., the number and type of used neighborhood sets. Using 16 BTF materials, each consisting of 6,561 images, testing ten random field models, and trying eight different neighborhood sets, each with three different Gaussian-Laplacean pyramid setups, we created 20,155,392 synthesized images, which were all compared with corresponding original data in total. The best result, i.e., the synthesized image most similar to the original using the MEMD criterion, was identified, and its corresponding MEMD value was used for the presented statistics. The mean MEMD of the best results and standard deviation for each tested model are counted for individual used materials and the whole test set. The results are the mean difference from original data in % and its corresponding standard deviation, summarized in Table 1 ($MEMD_{\max} = 255$), that shows significantly larger spectral error for large $\kappa(256) \geq 3$ values for silver gold, flocked paper, mica, and basketweave. 2D random field models can approximate all other materials after the KLT decorrelation with an acceptably minor spectral error.

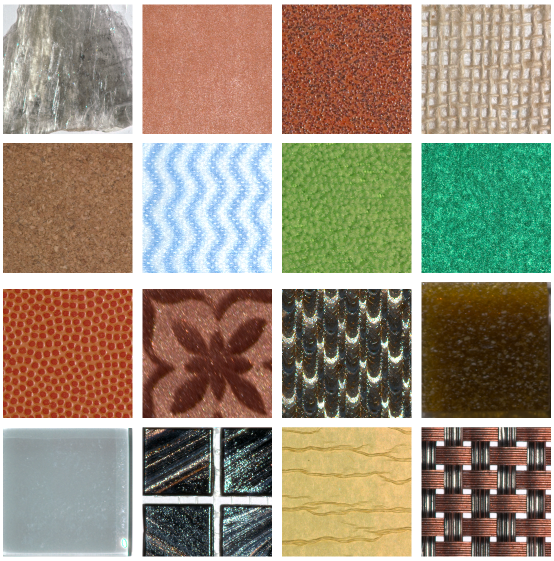


Figure 1: BTF MAM204 Dataset presentation (from upper-left to bottom-right): mica, sand-fine, sand-coarse, burlap, cork, towel, green cloth, green felt, basketball, flocked paper, silver gold, brown tile, glass tile, blue-black-gold tile, crinkle paper, and basketweave.

6 CONCLUSIONS

We presented a criterion allowing us to predict when a vast BTF data space can be modeled using a set of 2D random field models $(\kappa(\cdot) < 3)$ instead of more demanding 3D random field models without significant loss of spectral quality. This quality prediction allows avoiding demanding computing experiments with both types of models, spectrally decorrelated 2D and fully correlated 3D as well as circumvent the main problem with still non-existent reliable texture quality criteria. These mathematical models represent attractive modeling alternatives offering extreme data compression as only tens parameters must be stored instead of the original acquired data. They can reconstruct BTF space, i.e., predict the material's visual appearance under unmeasured conditions and synthesize textures of arbitrary size without disturbing visual artifacts and with preserved overall visual impressions. On the other hand, they can only approximate original data, which may result in visual quality compromise.

We performed robust experiments involving random field BTF models, analyzing and synthesizing under

---

[2] http://btf.utia.cas.cz

	mica		fine sand		coarse sand		burlap		cork		towel		green cloth		green felt		basketball	
Model	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
2D CAR	3%	9%	5%	2%	5%	4%	5%	2%	2%	1%	3%	1%	3%	1%	5%	2%	5%	4%
2D MA	10%	27%	11%	6%	10%	13%	8%	3%	4%	1%	6%	17%	5%	2%	11%	4%	10%	13%
2D MSAR	4%	10%	5%	2%	5%	2%	6%	2%	3%	1%	2%	1%	3%	1%	4%	1%	5%	2%
2D MMRF	4%	10%	5%	2%	5%	2%	6%	2%	3%	1%	2%	1%	3%	1%	4%	1%	5%	2%
2D PMRF	4%	10%	5%	2%	5%	2%	6%	2%	3%	1%	2%	1%	3%	1%	4%	1%	5%	2%
3D CAR	3%	9%	4%	2%	4%	4%	4%	2%	2%	1%	2%	1%	2%	1%	3%	1%	4%	4%
3D MA	4%	10%	5%	3%	5%	4%	4%	1%	2%	1%	3%	4%	2%	1%	5%	3%	5%	4%
3D MSAR	4%	10%	7%	3%	6%	5%	7%	4%	4%	2%	4%	2%	5%	2%	5%	1%	6%	5%
3D MMRF	4%	9%	5%	2%	5%	4%	6%	2%	3%	1%	3%	1%	4%	2%	5%	1%	5%	4%
3D PMRF	4%	9%	5%	2%	5%	4%	6%	2%	3%	1%	3%	1%	4%	2%	5%	1%	5%	4%
$\kappa(256)$	3.18		2.42		1.14		4.3e-02		1.87e-03		7.3e-03		8.9e-04		6.3e-02		1.14	
	flocked paper		silver gold		brown tile		glass tile		blue-black -gold tile		crinkle paper		basketweave				all materials	
2D CAR	5%	4%	8%	8%	4%	5%	1%	4%	3%	5%	1%	4%	9%	6%			4%	5%
2D MA	9%	6%	23%	22%	10%	19%	4%	19%	10%	19%	2%	12%	17%	10%			9%	15%
2D MSAR	5%	4%	10%	7%	4%	3%	1%	2%	4%	4%	1%	0%	11%	7%			4%	5%
2D MMRF	5%	4%	10%	8%	4%	2%	1%	2%	4%	4%	1%	0%	11%	7%			4%	5%
2D PMRF	5%	4%	10%	9%	4%	2%	1%	1%	4%	4%	1%	0%	11%	7%			5%	5%
3D CAR	4%	4%	8%	8%	3%	5%	1%	4%	3%	5%	1%	4%	8%	5%			4%	5%
3D MA	4%	3%	9%	9%	4%	5%	2%	6%	4%	6%	1%	6%	9%	6%			4%	6%
3D MSAR	6%	4%	9%	8%	4%	6%	2%	6%	4%	6%	1%	4%	11%	7%			5%	6%
3D MMRF	6%	5%	9%	9%	4%	5%	1%	4%	4%	5%	1%	4%	10%	7%			5%	5%
3D PMRF	6%	5%	9%	9%	4%	5%	1%	4%	4%	5%	1%	4%	10%	7%			5%	5%
$\kappa(256)$	18.8		91.9		7.33e-01		5.04e-07		2.07e-02		1.07e-05		3.70					

Table 1: Mean values and standard deviations percentual $MEMD^* = 100 * MEMD/MEMD_{\max}$ values error corresponding to the best-achieved results for individual models and materials.

various conditions. All synthesized data were compared with corresponding original ones using texture color composition criterion MEMD. Achieved results confirmed that BTF measurements of materials with a limited range of colors predicted with the $\kappa()$ criterion from one perpendicular illuminated and observed angle can be reliably reconstructed using a set of simpler 2D random field models. The only exception with tested models is the oversimplified 2D MA model, which has mostly poor spectral performance.

Results of multidimensional textures related research are also applicable in medical applications and varied image processing problems, such as image restoration, cultural heritage preservation. BTF data are useful in a study of shadow casting by material structure and the analysis of material dimensionality. Analysis of different BTF samples can help understand human perception of real-world materials.

7 REFERENCES

[Ben97a] Bennett, J. Modeling and Analysis of Gray Tone, Color, and Multispectral Texture Images by Random Field Models and Their Generalizations, Ph.D. thesis, Southern Methodist University, Dallas, 1997.

[Ben98a] Bennett, J., Khotanzad, A. Multispectral Random Field Models for Synthesis and Analysis of Color Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, (3), pp. 327–332, 1998.

[Col73a] Cole Jr., H. A. On-line Failure Detection and Damping Measurement of Aerospace Structures by Random Decrement Signatures. Technical Report TMX-62.041, NASA, 1973.

[Cul04a] Cula, O., Dana, K., Murphy, F., Rao, B. Bidirectional Imaging and Modeling of Skin Texture. IEEE Transactions on Biomedical Engineering, 51, 12, pp.2148-2159, 2004.

[Dan97a] Dana, K., Nayar, S., van Ginneken, B., Koenderink, J. Reflectance and Texture of Real-World Surfaces. Proceedings of IEEE Conference Computer Vision and Pattern Recognition, pp.151-157, 1997.

[Fil18a] Filip, J., Kolafová, M., Havlíček, M., Vávra, R., Haindl, M., Rushmeier, H. Evaluating Physical and Rendered Material Appearance. The Visual Computer, 34, Springer (Computer Graphics International 2018), pp.805-816, 2018.

[Fil08a] Filip, J., Chantler, M., Green, P., Haindl, M. A Psychophysically Validated Metric for Bidirectional Texture Data Reduction. ACM Transactions on Graphics, 27, (5), 2008.

[Gri09a] Grim, J., Somol, P., Haindl, M., Daneš, J. Computer-Aided Evaluation of Screening Mammograms Based on Local Texture Models. IEEE Transactions on Image Processing, 18, (4), pp.765-773, 2009.

[Hai14a] 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.

[Hai91a] Haindl, M. Texture Synthesis. CWI Quarterly, 4, (4), pp.305-331, 1991.

[Hai13a] Haindl, M., Filip, J. Visual Texture. Advances in Computer Vision and Pattern Recognition, Springer-Verlag, London, (2013).

[Hai02a] Haindl, M., Filip, J. Fast Restoration of color Movies Scratches. Proceedings of the 16th International Conference on Pattern Recognition, IEEE Computer Society, pp.269-272, 2002.

[Hai12a] Haindl, M., Havlíček, M. Bidirectional Texture Function Simultaneous Autoregressive Model. Computational Intelligence for Multimedia Understanding, pp.149-159, 2012.

[Hai02c] Haindl, M., Havlíček, V. A Multiscale color Texture Model. Proceedings of the 16th International Conference on Pattern Recognition, IEEE Computer Society, Los Alamitos, pp.255-258, 2002.

[Hai06a] Haindl, M., Vácha, P. Illumination Invariant Texture Retrieval. Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006, 3, IEEE Computer Society, pp.276-279, 2006.

[Hai07a] Haindl, M., Mikeš, S., Scarpa, G. Unsupervised Detection of Mammogram Regions of Interest. Knowledge-Based Intelligent Information and Engineering Systems. LNAI, 4694, Springer Berlin, pp.33-40, 2007.

[Hai08b] Haindl M., Mikeš, S. Unsupervised Mammograms Segmentation. Proceedings of the 19th International Conference on Pattern Recognition, ICPR 2008, IEEE Computer Society, Los Alamitos, pp.1-4, 2008.

[Has11a] Hasegawa, T., Tsumura, N., Nakaguchi, T., Iino, K. Photometric Approach to Surface Reconstruction of Artist Paitings. Journal of Electronic Imaging, 20, (1), 2011.

[HFV12a] Haindl, M., Filip, J., Vávra, R. Digital Material Appearance: the Curse of Tera-Bytes. ERCIM News, 90, pp.49-50, 2012.

[Hai23a] Haindl M. Bidirectional Texture Function Modeling, Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging, Springer International Publishing, pp. 1023–1064, 2023.

[Hav12a] Havlíček, M. Bidirectional Texture Function Three Dimensional Pseudo Gaussian Markov Random Field Model. Doktorandské dny 2012, ČVUT, pp.53-62, 2012.

[Hav15a] Havlíček, M. Extended Bidirectional Texture Function Moving Average Model. Doktorandské dny 2015, ČVUT, pp.37-43, 2015.

[Hav19a] Havlíček, M., Haindl, M. Texture spectral similarity criteria. IET Image Processing, 13(11), pp.1998-2007, 2019.

[Hav21a] Havlíček, M., Haindl, M. Optimized Texture Spectral Similarity Criteria. Advances in Computational Collective Intelligence, Springer International Publishing, Cham, pp.644-655, 2021.

[Hav23a] Havlíček, M., Haindl, M. Texture Spectral Similarity Criteria Comparison. WSCG 2023 Proceedings - 31. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG 2023, University of West Bohemia, Plzeň, pp.100-106, 2023.

[Li92a] Li, X., Cadzow, J., Wilkes, D., Peters, R., Bodruzzaman II, M. An Efficient Two Dimensional Moving Average Model for Texture Analysis and Synthesis. Proceedings IEEE Southeastcon 1992, 1, pp.392-395, 1992.

[Mal01a] Malzbender, T., Gelb, D., Wolters, H. Polynomial Texture Maps. ACM SIGGRAPH 2001, ACM Press, pp.519-528, 2001.

[Sca09a] Scarpa, G., Haindl, M., Zerubia, J. Hierarchical Multiple Markov Chain Model for Unsupervised Texture Segmentation. IEEE Trans. on Image Processing, 18, (8), pp.1830-1843, 2009.

[Vac10a] Vácha, P., Haindl, M. Content-Based Tile Retrieval System. Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science, 6218, Springer Berlin/Heidelberg, pp.434-443, 2010.

[Vac09a] Vácha, P., Haindl, M. Illumination Invariant and Rotational Insensitive Textural Representation. Proceedings of IEEE Int. Conf. on Image Processing, ICIP, pp.1333-1336, 2009.

[Wan09a] Wang, Z., Simoncelli, E. P. Translation Insensitive Image Similarity in Complex Wavelet Domain. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp.573-576, 2009.

[Zuj13a] Zujovic, J., Pappas, T., Neuhoff, D. Structural Texture Similarity Metrics for Image Analysis and Retrieval. IEEE Transactions on Image Processing, 22, (7), pp.2545-2558, 2013.

3D Multi-Criteria Design Generation and Optimization of an Engine Mount for an Unmanned Air Vehicle Using a Conditional Variational Autoencoder

Sebastian Eilermann
Helmut-Schmidt University
Germany, 22043 Hamburg
sebastian.eilermann@hsu-hh.de

Christoph Petroll
University of the Bundeswehr Munich
Germany, 85579 Neubiberg, BY
christophpetroll@bundeswehr.org

Philipp Hoefer
University of the Bundeswehr Munich
Germany, 85579 Neubiberg, BY
philipp.hoefer@unibw.de

Oliver Niggemann
Helmut-Schmidt University
Germany, 22043 Hamburg
oliver.niggemann@hsu-hh.de

ABSTRACT

One of the most promising developments in computer vision in recent years is the use of generative neural networks for functionality condition-based 3D design reconstruction and generation. Here, neural networks learn dependencies between functionalities and a geometry in a very effective way. For a neural network the functionalities are translated in conditions to a certain geometry. But the more conditions the design generation needs to reflect, the more difficult it is to learn clear dependencies. This leads to a multi criteria design problem due various conditions, which are not considered in the neural network structure so far. In this paper, we address this multi-criteria challenge for a 3D design use case related to an unmanned aerial vehicle (UAV) motor mount. We generate 10,000 abstract 3D designs and subject them all to simulations for three physical disciplines: mechanics, thermodynamics, and aerodynamics. Then, we train a Conditional Variational Autoencoder (CVAE) using the geometry and corresponding multicriteria functional constraints as input. We use our trained CVAE as well as the Marching cubes algorithm to generate meshes for simulation based evaluation. The results are then evaluated with the generated UAV designs. Subsequently, we demonstrate the ability to generate optimized designs under self-defined functionality conditions using the trained neural network.

Keywords

3D Generation, Multi-Criteria, Optimization, Engine Mount, Coditional Variational Autoencoder, Simulation based Evaluation

1 INTRODUCTION

The potential of using neural networks (NN) for computer aided design (CAD) generation shows new possibilities in the fields e.g. medicine, engineering as well as product development. Algorithms iteratively generate a variety of solutions in the shortest possible time for higher-performance designs [Seo22]. Only functionality requirements with some boundary conditions are needed. This is achieved by NN as they connect the functionality requirements, as conditions, directly to generated geometry features. The weights of the neural networks are adjusted based on the conditions for optimal material distribution during training. Trained neural networks are showing excellent results to learn these dependencies [Du21]. Thus, the NN can be trained to find new design variations which only consider functionality requirements. For this generative design processes, generative neural networks like Variational Autoencoder (VAE) [Kin13] and Generative Adversarial Networks (GAN) [Goo14] are often successfully used. For generative neural networks based approaches functionality design requirements are assigned as conditions for a specific geometry. Newer approaches use Conditional Variational Autoencoder (CVAE)[Soh15] to generate objects under specific conditions. Training such a model with an increasing number of conditions is a major challenge. For this purpose, a low dimensional representation like a latent space is mostly used to represent multi functionality dependencies. It gives the opportunity to compare designs due to their similarities [Shu20]. In this way, the chance is given to balance

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.

https://www.doi.org/10.24132/CSRN.3401.22

multi-criteria conditions for a higher-performance design. However, these latent spaces are difficult to interpret and analyze with a growing number of conditions. This is one reason why design problems solved so far with generative neural networks are mostly limited to a two-dimensional design space and a low number of considered conditions [Che22, Oh19a]. Further, existing approaches show lack of adaptation in the NN structure with the complexity of multiple conditions to learn the relationship between conditions and geometry. In particular, when conditions are strongly dependent, in the same geometry. Moreover, there are hardly any examples as a data basis for such 3D CAD generative neural networks use cases with major conditions. From this we derive our main research questions for this work:

1. Can a generative neural networks be trained on 3D synthetic data with multi physics based conditions to generate a 3D design for a real design task?
2. How must the structure of a generative neural network be extended to associate multi-criteria constraints during the training process with geometry features on a design?
3. How to train a generative neural network to predict more powerful designs and find an optimum under multiphysics conditions?

To answer our research questions, we have a multi-criteria 3D design use case regarding an engine mount for an unmanned air vehicle. We generate 10,000 3D generative designs, which is based on our earlier work [Pet21]. The designs are generated with respect to the functionalities and geometries features from our use case. With a total of 30,000 executed simulations we evaluate and label the generated generative designs in terms of its physics-based functionalities. The physics-based functionalities are mechanics, thermodynamics as well as aerodynamics. In addition, we introduce conditions, which enable an assessment of manufacturability with additive manufacturing [Bik19]. After, we train a CVAE with our labeled generative designs [Soh15]. A major challenge on our regression problem are the continuous values of our conditions. Moreover, the physical quantities values are numerically very different in scale. The introduction of more than one condition caused a large divergence in the latent space. It is an ambiguous learning behavior to concrete geometry features. Therefore, we semantically partition our conditions and extend the input NN structure of the CVAE. The trained model gives us the probability for a material prediction for all material areas in the design space. Finally, we use our trained model to generate an optimized design which fulfills best our multi-criteria functionalities. To the best of our knowledge, this is the first work that addresses the problem of a 3D design problem with regression multi-criteria conditions with a generative neural networks. Our contribution in this paper is threefold:

- An extended conditional variational autoencoder approach to open up a three-dimensional solution space with a geometrically parameter-free description of a component under multiple physics based conditions.
- An approach for a higher performance design generation, from a multi-condition learned relationship between latent representations and the generated designs.
- An evaluation of our presented approach on a 3D use case, with an interpretation of the latent space of a successfully trained generative neural networks for an optimal component design.

2 RELATED WORK

2.1 Deep Learning for 3D Data

In the field of computer vision, there has been a significant development of different deep networks for a variety of different tasks in recent years. For this reason, a variety of methods based on VAEs [Bro16, QY20], GANs[Gao22, Shu20], diffusion models [Ye22, Zhe22] as well as normalising flows [Klo20] have been explored to generate 3D objects as mesh, point cloud or voxel representations. So in the field of 3D object recognition to implement a joint embedding of 3D shapes and synthesised images approaches are shown in [Li15, Su15a]. Another approach is presented in [Sha16] where the researchers used voxel-based models with an autoencoder to represent 3D objects. A more effective approach is used in [Qi17a] where a point-like representation is used to explore 3D objects. Other approaches like in [Yan16a] use 2D images together with a 3D to 2D projection layer to generate 3D objects. Besides the classical use of the presented approaches for classification tasks [Qi17a, Sha16], the approaches can also be used for completing full shapes [Che19b, Tch19a] or for single-view reconstruction [Man18]. Furthermore, [Che19a, Fu22] are exploring text-based 3D object generating approaches.

2.2 Conditional Variational Autoencoder

Based on the concept of a VAE [Kin13] a Conditional Variational Autoencoder [Soh15] (CVAE) is considered good to represent the high-dimensional joint distributions of features [Kim21a, Soh15, Yon21].

The main target of VAEs is the estimation of the relation between the input x_i and the corresponding latent representation z_i. In variational inference, the posterior $p(z|x)$ is approximated by a parameterized distribution

$q_\theta(z|x)$ called the variational distribution. The lower bound for $p(x)$ can be written as follows [Kin13]:

$$L_{\theta,\phi,x} = E_{q_\theta(z|x)}\ [log\ p_\phi(x|z)] - KL(q_\theta(z|x)||p_\phi(z)) \tag{1}$$

The two fundamental parts of the VAE are the Encoder $E = q_\theta(x|z)$ with parameters θ and the Decoder $D = p_{\phi_D}(x|z)$ with parameters ϕ. They represent functions which map the input x_i to a latent space z_i and vice versa. The reconstruction from x_i is $\hat{x}_i$. Here, the represented optimization is an minimization of the reconstruction loss under consideration of the KL divergence as an regularizer. E has two outputs μ_i and σ_i that correspond to the mean and the standard deviation of the Gaussian latent variable z_i. For this, the reparameterization trick [Kin13] is normally used with $\mu_i + \sigma_i * \varepsilon$ under consideration of $\varepsilon_i \sim N(0,1)$ to calculate z_i. It helps the network to shift not to much from the true distribution.

In contrast to the VAE, a CVAE approach based on the maximisation from the variational lower bound of the conditional likelihood $p(x|c)$ which supports to generate designs under multiple specified conditions $\mathbf{c} = \{c_1 \dots c_n\}$ where n is the number of conditions [Soh15].

$$\begin{aligned} L_{\theta,\phi,x,\mathbf{c}} = & E_{q_\theta(z|x,\mathbf{c})}\ [log\ p_\phi(x|z,\mathbf{c})] \\ & - KL(q_\theta(z|x,\mathbf{c})||p_\phi(z|\mathbf{c})) \end{aligned} \tag{2}$$

The trained CVAE is usable to reconstruct an input x_i under a set of conditions $\mathbf{c}$ to match the target outputs $\hat{x}_i$. In contrast to the VAE, the main parts of the CVAE E and D are conditioned by $\mathbf{c}$. It follows that $E = q_\theta(z|x,\mathbf{c})$ with parameters θ and $D = p_{\phi_D}(x|z,\mathbf{c})$ with parameters ϕ which represents functions are used to map the input x_i under consideration of $\mathbf{c}$ to a latent space z_i and vice versa. In this context, a core problem is when working with multiple conditions in a CVAE is how to bring them into the network. Also the weighting or balancing problem of the reconstruction error and the Kullback-Leibler divergence shows this. It has been object of several investigations [Asp20].

2.3 Deep Learning for Engineering Tasks

For iterative design generation, in [Shu20] a GAN based approach is shown for direct 3D modeling of an aircraft. The work followed the idea of a physics-based generated dataset. Thereby, the aerodynamics are considered primarily and the shape as the single condition. The goal is to minimize the aerodynamic drag. Furthermore, in [Hey21] an approach is developed for generating 3D models with more constraints. The researchers add a range loss, so design constraints are additionally taken into account based on parameter specifications using the example of 3D aircraft models. A slightly different approach is presented in [Zha19] for the optimization of 3D models. After successfully training of a variational autoencoder, a genetic algorithm is used to optimize the latent space design embeddings. Further, an approach to consider continuous conditions in the generation process with Conditional GANs is shown in [Nob21]. They use a singular vicinal loss in combination with a loss function based on determinant point processes. In doing so, the researchers add a new self-amplifying Lambert Log Exponential Transition Score, which is used for improved conditioning. They successfully demonstrate the approach on an 2D airfoil generation task with diverse results. Similarly, a Free-Form Deformation Generative Adversarial Networks which provides efficient parameterization for 3D shapes is presented in [Che21]. Hereby, they achieve high representation compactness and capacity. A VAE to select an optimal material strength for their 2D optimization approach to retrieve a result from a latent space is shown in [QY20]. They take a structure optimization and determines the optimal material from the latent space of their trained VAE. A two-dimensional shape optimization based on the Bezier GAN, where the approach is based on a parameterized representation of a 2D shape is introduced in [Che22].

The work presented shows the difficulty of available data for design problems. Data for more complex solutions for multiple conditions isn't published. 3D data and corresponding physics-based labels are missing. GAN approaches are available in detail mostly with one considered condition. Multiphysics problems are missing in the context of direct 3D design creation completely or don't deal with real physics-based designs [Ugu19]. Further, it is recognisable that generative neural networks are often used for classification problems, which have not been further discussed here. In summary, an approach which allows to incorporate three-dimensional multi-criteria designs with regression conditions into a generative neural network is missing. Therefore, no extended NN approaches which have a change in their architecture in favor of multi-criteria conditions do exist.

3 METHOD

In this section we propose our method to effectively bring continuous multi-criteria conditions into a new design of an UAV. In doing so, we solve a multi-physics design problem with a CVAE. The use case gives concrete functionalities and geometry features. Finally, a multi-physics and functionally generative optimal design is presented. Optimal with regard to the physical conditions. To achieve this target, we developed a four-step approach to use a generative neural networks for a new design of a component (Figure: 1).

First, we generate 10,000 designs. We do it with a pseudo random noise based on [Bae18] and our earlier

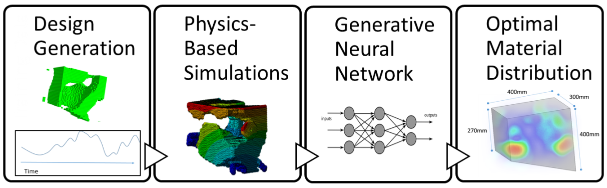

Design Generation
Time
Physics-Based Simulations
Generative Neural Network
inputs
outputs
Optimal Material Distribution
400mm
300mm
270mm
400mm

Figure 1: Multi-criteria generative neural networks design approach. Four steps of using a Deep Input CVAE for functionalities based design generation.

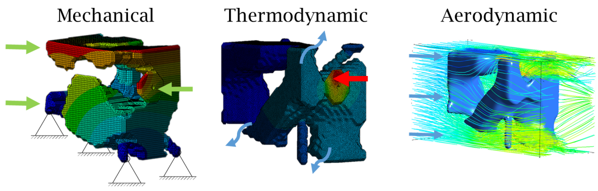


Figure 2: ANSYS simulations for mechanics, thermodynamics and aerodynamics with generated designs and specific load cases. The arrows represent the direction of it.

work [Pet21]. The special random function should ensure that the design space is covered completely and evenly [Pet21]. For the neural network each voxel should occur equally often in training. This is done to teach the neural network a parameter-free geometric generative description of design variants. In this way a geometric solution appears as unrestricted as possible in the design space (generative design). Second, we use a physics-based simulation to evaluate our designs with respect to the functionalities. Each generated generative design is labeled with its physical performance data. Third, we train a generative neural networks with an extended architecture and our generated designs as well as simulation based labels as input. It learns where material in the design space is important or unimportant for the physics-based functionalities. In a last step the trained NN is used to generate an optimal design. At this point the lower dimensional latent representation is used for a prediction of a new design under regression multi-criteria conditions. The main differences of our approach to existing approaches is as follows. We show a generative neural networks based approach in which not only single criterion requirements for a design problem are solved. The design problem is three-dimensional, multi-physical and considers geometric requirements and interfaces. A concrete use case and additive manufacturing are also addressed. The extension of a CVAE is developed, demonstrated and improved for multi criteria conditions.

3.1 Training Data Generation

To create the 10,000 generative designs $\mathbf{X} = \{x_1 \dots x_i\}$, $i \in \{1 \dots 10{,}000\}$ we use a noise based generation method. We define our design space A with 50,400 voxels (Figure: 3)

$$\mathrm{A} = \{(\mathrm{a}_{j,k,l}) | \forall\, j = 1,2,\dots 30,\ k = 1\dots 40,\ l = 1\dots 42\} \tag{3}$$

where one voxel $\mathrm{a}_{j,k,l}$ per cm^3 is used. This comparatively rough representation is chosen due to the expected long computation and power calculation time. Next a three-dimensional Perlin Noise (noise) is used to generate a basic material distribution $\mathrm{Md}_{\mathrm{AM},i}$ in the design space A

$$\mathrm{Md}_{\mathrm{AM},i} = \sum_{n=0}^{M-1} \hat{u} * \mathrm{noise}(\nu_n * \mathrm{a}_{j,k,l}) \tag{4}$$

with amplitude modulation AM, frequency ν and amplitudes $\hat{u}$. Here, $\hat{u}_{n+1} = \hat{u}_n * \phi_{\text{noise}}$ is guilty where a combination of the frequency and amplitude modulation with different frequencies is used. At this point, ϕ_{noise} is a special constant which links the amplitude with the amplitude of the previous step. This creates uniform coverage of the design space. At the correct scale it produces organic-looking designs due to the basis of locally contiguous duration's.

After, where the engine mount needs interfaces to the engine and to aircraft structure, material is used per design (Figure: 3). Through repetitive areas, the NN learns where in any case must be material for addon parts. Algorithms are used to ensure that the designs can be use for a physic-based simulation [Pet21]. So the design consists of only one body and can be flowed through by air[Pet21]. As a final step, the designs are transformed into a surface description for simulations. The described steps from Eq. 4 are repeated until a quantity of 10,000 generated designs $\mathbf{X}$ is achieved.

3.2 Physics Based Simulations

The simulation based label generation is done with automated simulations in ANSYS [Mad15] FEM and CFD (Figure: 2). For this purpose, one mechanical, one thermal and one aerodynamic simulation for each generated design x_i is performed. We take these as the basis for our considered physic-based parameters, which are most expressive for our use case. So our conditions where $n = 9$ are the following:

For the mechanics, we evaluate the mean residual stress c_1 and mean total deformation c_2 for all voxels. For thermodynamic, mean temperature c_3 and heat density c_4. In aerodynamics we consider the mean outlet pressure c_5 and the resistance to air c_6 in the direction of flow. For the previously mentioned conditions, we don't use maxima values cause of bad training tests.

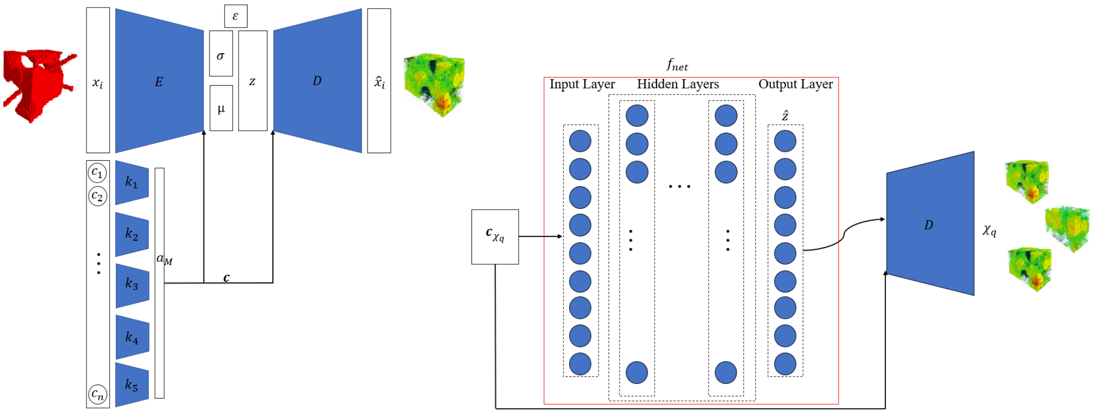


Figure 3: On the left our developed Deep Input CVAE approach for reconstruction of 3D object with multiple conditions c_n and $k_{1,\ldots,5}$ is illustrated. On the right our approach for design generation with a FNN as well as the trained decoder D is shown. The data flow as well as the neural network structure is shown to generate new designs χ_q with self selected values $c_{\{\chi_q\}}$. In red, f_{net} is highlighted.

Instead, we use mean values, so that the value distribution corresponds more closely to a Gaussian distribution. The assessment with regard to additive manufacturing is based on the heat distribution in the printing process c_7 and build-up angle of number to surfaces c_8. A lightweight design criterion c_9 is introduced as a classic optimization factor. For this consideration we use the number of voxels per design. Finally we have a set 10,000 value pairs with one value for each condition per generated design $\mathbf{C}_x = \{\mathbf{c}_{x_1}, \ldots, \mathbf{c}_{x_{10000}}\}$. So, all values are normalized per condition for faster and better training. Values in the set of conditions $c_n = \{c_{n,1}, \ldots, c_{n,10000}\}$ which are not in range of $\pm\, 2\sigma$ are dropped out. This leads to a more uniform training process, which does not focus on maximums.

3.3 Extended CVAE for Multi-Physics Based Design Generation

In our approach we use a CVAE-approach like in 2.2 explained with the generated designs $\mathbf{X}$ and the values of $\mathbf{C}_x$ as input. With data augmentation like in [Kar22] we support our training. Further we use seven dense layers to reduce the size as follows $(50400, 1024, 512, ..., 16)$ as well as in reverse order in D. After each layer we use a Rectified Linear Unit (ReLU) activation function The latent size z is defined as 32. Further we apply an Adam optimizer [Kin14]. For the generation of $q(z)$ as close as possible to the standard normal distribution we are using a two-part loss function with the reconstructions loss $E[\cdot]$ as well as a KL-divergence loss $KL[\cdot]$ like in Eq.2.

In comparison to the state of art, we divide $\mathbf{c}$ given the physic discipline of each c_n into five categories $\mathbf{K} = \{k_1 \ldots k_b\}, b \in \{1 \ldots, 5\}$. In doing so, we have used the following allocations: $k_1 = \{c_1, c_2\}$, $k_2 = \{c_3, c_4\}$, $k_3 = \{c_5, c_6\}$, $k_4 = \{c_7, c_8\}$ and $k_5 = \{c_9\}$. This extension supports the combination of values which differ significantly in their dimensions. In this context, we use complementary simple feedforward neural network (FNN) structure extension for each category in the input of our Deep Input CVAE (D-CVAE). The five extensions are added in one layer $a_{\mathbf{M}}$ after seven hidden layers size $(4, 8, ..., 256)$ per extension k_b and concatenated with last layer of E and the first layer of D. This semantically separated and more complex representation of our input improves the representation of the complex data strongly. After training the D-CVAE, trained D represents a parametric model where z_i and $\mathbf{c}_i$ are input parameters to generate new designs. So, an opportunity is given with the trained D-CVAE to generate a new design χ_q with the desired performance maximization across all conditions. The D-CVAE architecture previously described is shown in Figure 3.

3.4 Design Optimization

An approach for an optimal design generation χ_{opt} follows on. First to generate a new design χ_q with self selected values for each condition in $\mathbf{c}$, the relationship between the latent representation per design z_{x_i} and $\mathbf{c}_{x_1}$ is trained. For this purpose a FNN (f_{net}) is used to learn this relationship to predict $\hat{z}_{x_i}$ as a new representation:

$$\hat{z}_i = f_{\text{net}}(\mathbf{c}_i) \tag{5}$$

In- and output variables to train the FNN $f_{\text{net}}(\mathbf{c})$ with eight hidden layers and a ReLU activation function to predict a latent representation per generated design $\hat{z}_{x_i}$ are $\mathbf{c}_{x_1}$ as well as z_{x_i}. The trained f_{net} allows with trained D and self selected values $\mathbf{c}_{\chi_q}$ for $\mathbf{c}$ to predict a permissible quantity q (e.g. $q = 100$) of new individual models $\mathrm{Q} = \{\chi_1, \ldots, \chi_q\}$. Therefore the following applies under consideration of $\mathbf{c}_{\chi_q}$

$$\chi_i = D(\hat{z}_i, \mathbf{c}_{\chi_q}). \tag{6}$$

The explained approach to generate χ_q is shown on the right in Figure: 3.

Once f_{net} is successfully trained, $\hat{z}_i$ can be used in a three step way to determine an optimal design χ_{opt} in performance across all conditions.

First, the values of every physical property $\mathbf{c}_{n,x_i}$ are ordered per condition from the user's point of view from the lowest to the maximum performance. This is given by the minimum and maximum performance by the value range per condition. This creates new value pairs $\mathbf{C}_\chi = \{\mathbf{c}_{\chi_q}, \ldots, \mathbf{c}_{\chi_q}\}$. These value pairs are are not previously present in $\mathbf{C}_x$. The new value pairs are the basis for more powerful designs. Thereby, χ_1 has the lowest performance for all c_{n,χ_q} while χ_q has the highest performance per c_{n,χ_q} so the following is guilty

$$\begin{aligned}\chi_1 = D(\hat{z}_1, \mathbf{c}_{\chi_1}) < \chi_2 = D(\hat{z}_2, \mathbf{c}_{\chi_2}) < \ldots \\ < \chi_q = D(\hat{z}_q, \mathbf{c}_{\chi_q}).\end{aligned} \tag{7}$$

Next, to see if we can push our self-selected values even further to a higher performance design, we look at the variety which our D-CVAE can provide. For this we use the material change rate ΔM of each design point per design χ_q to the next χ_{q+1} calculated by

$$\Delta M = \sum_{1}^{q-1} \frac{\sum_1^j \sum_1^k \sum_1^l (A_{j*k*l})_{\chi_{q+1}} - (A_{j*k*l})_{\chi_q})}{j*k*l}. \tag{8}$$

We use Eq.8 to define the range where the new values for our defined conditions can be set. We assume that a new value per condition can only be set in the area where trained D has enough diversity in the design. This defines the limit for our trained model to retrieve a design with maximum performance from the latent space. Finally, we analyze the point where the material change rate is maximum while considering maximum performance. This results in the best possible design Eq. (9) with the presented optimization approach.

$$\chi_{\text{opt}} : max\ f(\Delta M)\ \text{for}\ min\ D(\hat{z}_i, \mathbf{c}_{\chi_q}) \tag{9}$$

Here, χ_{opt} defines the optimal material distribution for a higher performance design. For a simulation based validation χ_{opt} has to be transformed manually via [Lor98] to a CAD file.

4 EXPERIMENTS

In this section, we report the details of our experiments and the qualitative and quantitative validation. We compare our approach with a 3D Convolutional Neural Network in conjunction with a CVAE (CNN-CVAE) presented by [Na18] and a fully connected layer (FC-CVAE) presented in [Can19a]. In addition, we show the results in terms of an optimal design generation.

4.1 Use Case

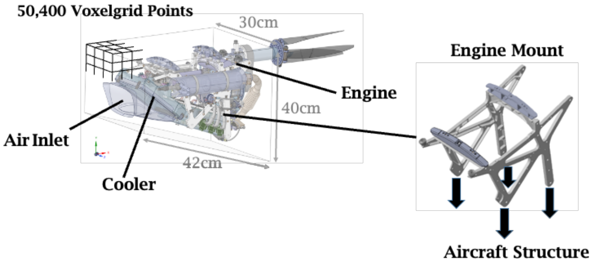


Figure 4: On the right is the complete UAV drive unit and on the left the engine Mount. We use a voxel-based geometric description.

Our use case is the design of an engine mount for an unmanned air vehicle (UAV) displayed in Figure 4). The idea is to reduce the number of components as far as possible to one central design with add-on parts (e.g. electronic, engine). For this purpose, the new possibilities of additive manufacturing are considered.

To achieve the target, we analyze the engine in terms of its main functionalities. In this case, the Wankel engine is attached to an engine mount that transmits the thrust to the aircraft structure. For operation, there is a radiator at the beginning of the engine, which cools the engine through coolant pipes located on the engine mount. In the particular case of the launch phase on a catapult, much heat is transported from the engine in the engine mount.

The main functionality of the engine mount can be described as the static stability to hold and sufficient heat dissipation to cool the engine. To ensure these functionalities, air must flow freely through the engine mount. Our goal is derived from this to design with a CVAE a holder which can withstand the mechanical and thermal load case, and has a favorable aerodynamic design. In addition, conditions which make metal additive manufacturing feasible must be considered.

4.2 Training Settings

The Training is done on a Xeon 4108 with 64GB RAM and 1 GPU NVIDIA P5000. Training results for the mentioned types generative neural networks are shown in Figure (5). It can be seen for multiple conditions the reconstruction result for our designs becomes more and more fuzzy. First, when using a CNN-CVAE compared to the FC-CVAE the core body of the design is presented well. However, fine details and the edges in the designs are not taken into account. Also, the training time is 8h for 200 epochs. Therefore, hyperparameter tuning is very time consuming. Compared to a training time of 40 minutes, the FC-CVAE is much faster, but it shows a very noisy design. Interesting is the observation of areas where material is very unlikely which is displayed numeric negatively (dark blue areas in Figure 5).

In the following Table 1 our final loss values are pre-

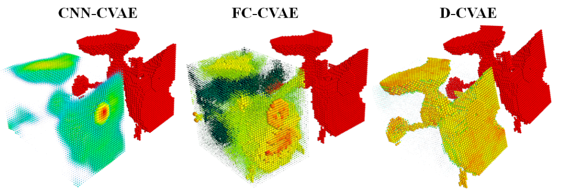


Figure 5: Reconstruction of the models depending on the condition. On the right side in red the original and on the left in colors representing the probability of material.

Model	KL-Loss	Rec. loss	Total
CNN-CVAE	90	6,050	6,140
FC-CVAE	2,500	30,000	32,500
D-CVAE	**80**	**520**	**600**

Table 1: Absolute final values of the loss functions as well as the total loss after training per considered generative neural network-approach. The smaller the error value, the better the model.

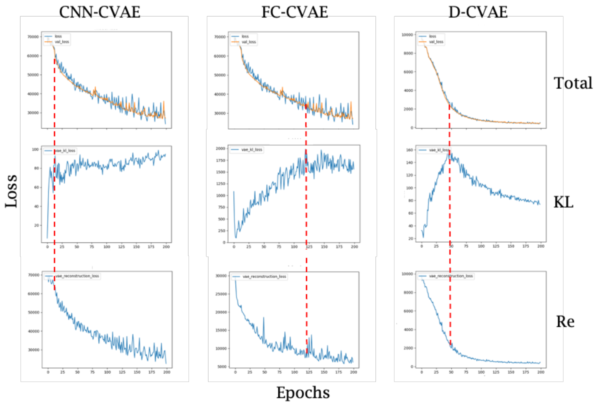


Figure 6: Representation of the loss functions of the 3 trained models. Top total loss, mid KL loss and bottom reconstruction loss. The red line points to the maximum in the KL loss and the adjustment of the reconstruction loss. It is recognizable that our developed D-CVAE has the best training curves.

sented. The reconstruction loss (Rec. loss) can be understood as the number of misrepresented voxels. The KL Loss is a measure for the quality of the conditions learned. Our D-CVAE shows a natural balancing of our label to learn the latent space. This is shown in Figure 6. The total loss with of the D-CVAE improves significantly compared to the CNN-CVAE and FC-CVAE and the two loss components. KL Loss and Reconstruction Loss, converge (redline) by themselves in such a way that the conditions have a sufficient influence.

Model	Abs. Error Design Space
CNN-CVAE	1,762
FC-CVAE	2,625
D-CVAE	**60**

Table 2: Abs. error in design space of design predictions $\hat{x}_i$.

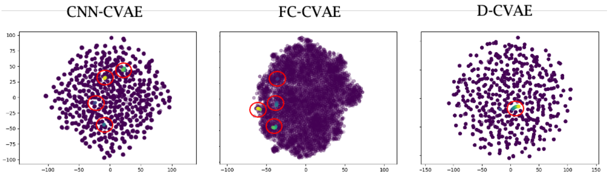


Figure 7: Comparison of the latent space of the t-dispersed stochastic neighborhood implantation of the 100 user wanted performance values. The latent space of our D-CVAE has the best spread data representation.

On the basis of the representation of the learned design (Figure 5 and Table 2) our D-CVAE approach generates qualitatively and quantitatively better results than the CNN-CVAE and FC-CVAE approaches. The D-CVAE shows the highest accuracy when it comes to mapping the contour. By adding up the probabilities of the predictions, the reliability of the predictions of a geometry from the condition can be determined by variance in Table 3.

4.3 Multi-Criteria Optimization of a 3D Design

Next we specify our own values per condition $c_{n,\chi}$ to generate new models in the interest of design optimization. We create $\chi = 100$ values per condition $c_{n,\chi}$ from good to bad in the sense of our used case and the performance. The values are selected as follows: Thermals and mechanics should withstand the loads as much as possible and are demanded as constant conditions. Aerodynamics and manufacturability should improve over the 100 labels from 0-100. The ninth condition (c_9), which should ensure that less material is used, as a classic optimization requirement. $Min.$ and $Max.$ from the simulated conditions are used as upper and lower limits. The challenge here, is that condition combinations are now required which are not previously learned in the latent space. In total these are 100 new values pairs uzk of unknown designs. These hundred conditions are used to retrieve the desired designs in the form of material distributions from the latent space with the decoder $D(\hat{z}_q, \mathbf{c}_{\chi_q})$.

In the following, we use the term material distribution instead of design proposal, because the strongly competing nine conditions lead to the fact that no distinct design for arbitrary condition combinations can emerge clearly. Unfortunately, for our validation with simulations, each material prediction with the new conditions has to be reconstructed manually. Thus, 4 examples each are chosen and simulated evenly split between 0-100 to look at the variance. From this, the variance σ to

Modell	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
VAR CNN-CVAE	0.210	0.30	261	0.07	1,202	569	9.07	1,047	0.07
VAR FC-CVAE	0.150	0.31	169	0.09	2,015	2,000	14.05	476	0.10
VAR D-CVAE	0.077	0.27	62	0.03	939	2,030	7.22	501	0.13
CNN-CVAE	→	↓	→	→	↗	↑	↗	↗	↑
FC-CVAE	→	→	↘	↘	→	↗	→	↑	↑
D-CVAE	→	→	→	→	↗	↗	↑	↑	↑
Target performance	→	→	→	→	↑	↑	↑	↑	↑

Table 3: Variance (VAR) from predicted and simulated designs in comparison to the used condition. Together qualitatively presented with the desired performance that should be achieved

the expected value c_{χ_i} is calculated from the given 100, $q \in Q$ conditions with $\sigma^2 = \frac{1}{q}\sum_1^q (c - c_{\chi_i})^2$.

In addition to Table 3, we evaluate how the value proceed from good to bad qualitatively. In comparison how it should develop according to the wanted performance per condition. Here, ↑ represents a qualitative improvement, ↓ on the other hand a degradation, → a preserve of the condition. The target performance defines how a better performing design should behave. The D-CVAE shows the best results in terms of the qualitative consideration of the conditions. It can be seen that the D-CVAE tends to have lower variance in its simulated predictions than the other models (Table: 3).

The 100 desired labels can be seen in the latent space of the models in Figure (7). It is recognizable that a clearer range in the D-CVAE appears. The area in which the 100 conditions are retrieved is contiguous (red rings). The 100 points are marked from blue (poor performance) to yellow (good performance). Also, the distribution of the data shows more clearly distributed and separated points, which is indicative of a more diverse learned latent space. Finally, we want to find the best possible solution for our 3D multi-criteria design problem. The goal is to find a material distribution in the design space that maximizes performance considering the conditions. But there are natural limits to retrieving better and better design proposals from our model. To find them we look at the range in which our model still shows sufficient diversity material prediction with respect to the conditions. For this we use the material change gradient from one design point to the next for our 100 created conditions (Eq. 8). The range in which significant material change can still be predicted for high performance is of interest. We searched in that manner for the best design with our trained model. We chose one recognizable maximum in the material change rate close to the maximum performance. The results are illustrated in Figure 8, it shows the product of a condition point. We simulate our optimum in all physical disciplines cf. Table 4 and compare it to the best in our previously generated training data. From a qualitative point of view the results makes sense. There is a lot of material around the engine to remove the heat, there is a solid connection to the mounting points. In addition, the engine is directly surrounded by cooling air from two sides. The optimal model we generated is much better compared to the best model in the training data set.

Condition	Training	Our Opt.	Dev.[%]
$c_1[MPa]$	0.062	0.10	-48
$c_2[mm]$	0.0032	0.0072	-56
$c_3[K]$	216	291	+26
$c_4[\frac{kW}{s}]$	0.047	0.095	-51
$c_5[Pa]$	198	26	+716
$c_6[N]$	116	52	+223
$c_7[\frac{mm^2}{layer}]$	7.175	2.300	+311
$c_8[surfaces]$	1,317	795	+165
$c_9[voxel]$	0.234	0.160	+146

Table 4: Our optimum compared in percent to the model with the best performance in our training's data set. The results for the mechanical and thermal load case remain the same as intended and keep the conditions. The other conditions improve significantly.

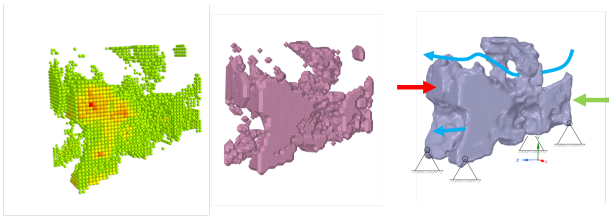


Figure 8: Optimal material distribution and reconstructed CAD design. Arrows indicate the simulated load case for validation.

5 CONCLUSION

The presented approach is one way to generate a 3D parameter-free geometry for a real multi-physics design problem with a D-CVAE. The main problem of determining a material distribution and linking geometry features to multiple regression conditions is met. However, the automated evaluation of the generated designs with D-CVAE is still a major obstacle for such a complex use case. It must be inferred repeatedly from to the material distribution to the design, similar to a classical topology optimization result. In this work one best design is shown in terms of our conditions as an optimum. It can be achieved without ground truth with the help of synthetic data. We have no comparison to an optimized component for all criteria with another method yet. Simpler data would not adequately address the complex challenge of multi-criteria design generation. Therefore, in further work, we concentrate on completely different conditional generative neural network approaches and new ways to clearly generate designs with multiple conditions. So, a faster and automated evaluation can be done with our data. The data and code are available upon reasonable request.

6 ACKNOWLEDGMENTS

This research as part of the project LaiLa is funded by dtec.bw - Digitalization and Technology Reasearch Center of the Bundeswehr which we gratefully acknowledge. dtec.bw is funded by the European Union - NextGenerationEU.

7 REFERENCES

[And01a] Anderson, R.E. Social impacts of computing: Codes of professional ethics. Social Science, pp.453-469, 2001.

[Con00a] Conger., S., and Loch, K.D. (eds.). Ethics and computer use. Com.of ACM 38, No.12, 2000.

[Con00b] Mackay, W.E. Ethics, lies and videotape, in Conf.proc. CHI'00, Denver CO, ACM Press, pp.138-145, 2000.

[Asp20] Asperti, A., and Trentin, M. Balancing reconstruction error Kullback Leibler divergence in Variational Autoencoders. IEEE Access 8, pp.199440-199448, 2020

[Bae18] Bae, H.J. et al. A Perlin noise based augmentation strategy for deep learning with small data samples of HRCT images. In: Scientific reports 8.1, pp. 1-7, 201.

[Bik19] H. Bikas, A. K. Lianos, and P. Stavropoulos, A Design Framework for Additive Manufacturing. *International Journal of Advanced Manufacturing Technology*, vol. 103, pp. 9–12, 2019. `https://doi.org/10.1007/s00170-019-03627-z`

[Bro16] Brock, A. et al. Generative and discriminative voxel modeling with convolutional neural networks. In: arXiv preprint arXiv:1608.04236, 20.

[Can19a] Canchumuni, S., Emerich, A., and Pacheco, M. Towards a robust parameterization for conditioning facies models using deep variational autoencoders and ensemble smoother. In: Computers & Geosciences 128, pp. 87-102, 2019.

[Che19a] Chen, K. et al. Text2shaape: Generating shapes from natural language by learning joint embeddings. In: Computer-Vision-ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2-6, 2018, Revised Selected Papers, Part III 14. Springer, pp. 100-116, 2019.

[Che22] Chen, Q. et al. Inverse design of two-dimensional airfoils using conditional generative models and surrogate log-likelihoods. In: Journal of Mechanical Design 144.2, 2022.

[Che21] Chen, W., and Ramamurthy, A. Deep generative model for efficient 3D airfoil parameterization and generation. In: AIAA Scitech 2021 Forum, p. 1690, 2021.

[Che19b] Chen, X., Chen, B., and Mitra, N. J. Unpaired point cloud completion on real scans using adversarial training. In: arXiv preprint arXiv:1904.00069, 2019.

[Du21] Du, X., He, P., and Martins, J. Rapid airfoil design optimization via neural networks-based parameterization and surrogate modeling. In: Aerospace Science and Technology 113, p. 106701, 2021.

[Fu22] Fu, R. et al. Shapecrafter: A recursive text-conditioned 3d shape generation model. In: Advances in Neural Information Processing Systems 35, pp. 8882-8895. 2022.

[Gao22] Gao, J. et al. Get3d: A generative model of high quality 3d textured shapes learned from images. In: Advances In Neural Information Processing Systems 35, pp. 31841-31854, 2022.

[Goo14] Goodfellow, I. et al. Generative adversarial nets. In: Advances in neural information processing systems 27, 2014.

[Hey21] Heyrani Nobari, A., Chen, W., and Ahmed, F. Range GAN: Range-Constrained Generative Adversarial Network for Conditioned Design Synthesis. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Vol. 85390. American Society of Mechanical Engineers, 2021.

[Seo22] Jang, S., Yoo,S., and Kang, N. Generative design by reinforcement learning: Enhancing the diversity of topology optimization designs. Computer-Aided Design, 146:103225, 2022

[Kar22] Kar, O. F. et al. 3D Common Corruptions and Data Augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18963-18974, 2022.

[Kim21a] Kim, J., Kong, J., and Son, J. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In: International Conference on Machine Learning. PMLR, pp. 5530-5540, 2021.

[Kin14] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. In: arXiv preprint arXiv:1412.6980, 2014.

[Kin13] Kingma, D. P., and Welling, M. Auto-encoding variational bayes. In: arXiv preprint arXiv:1312.6114, 2013.

[Klo20] Klokov, R., Boyer, E., and Verbeek, J. Discrete point flow networks for efficient point cloud generation. In: European Conference on Computer Vision. Springer, pp. 694-710, 2020.

[Li15] Li, Y. et al. Joint embeddings of shapes and images via cnn image purification. In: ACM transactions on graphics (TOG) 34.6, pp. 1-12, 2015.

[Lor98] Lorensen, W. E., and Cline, H. E. Marching cubes: A high resolution 3d surface construction algorithm. In Seminal graphics: pioneering efforts that shaped the field, pages 347-353, 1998.

[Mad15] Madenci, E., and Guven, I. The finite element method and applications in engineering using ANSYSÂ®. Springer, 2015.

[Man18] Mandikal, P. et al. 3D-LMNet: Latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image. In: arXiv preprint arXiv:1807.07796, 2018.

[Na18] Na, J., Jeon, K., and Lee, W. B. Toxic gas release modeling for real-time analysis using variational autoencoder with convolutional neural networks. In: Chemical Engineering Science 181, pp. 68-78, 2018.

[Nob21] Nobari, A. H., Chen, W., and Ahmed, F. Pcdgan: A continuous conditional diverse generative adversarial network for inverse design. In: arXiv preprint arXiv:2106.03620, 2021.

[Oh19a] Oh, S. et al. Deep generative design: Integration of topology optimization and generative models. In: Journal of Mechanical Design 141.11, 2019.

[Pet21] Petroll, C. et al. Synthetic Data Generation for Deep Learning Models. In: DS 111: Proceedings of the 32nd Symposium Design for X (DFX2021), pp. 1-10, 2021.

[Qi17a] Qi, C. R. et al. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 652-660, 2017.

[QY20] Qian, C., and Ye, W. Accelerating gradient-based topology optimization design with dual-model neural networks. In: arXiv preprint arXiv:2009.06245, 2020.

[Sha16] Sharma, A., Grau, O., and Fritz, M. Vconv-dae: Deep volumetric shape learning without object labels. In: Computer Vision-ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14. Springer, pp. 236-250, 2016.

[Shu20] Shu, D. et al. 3D Design Using Generative Adversarial Networks and Physics-Based Validation. In: Journal of Mechanical Design 142.7, issn: 1050-0472. doi: 10.1115/1.4045419, 2020.

[Soh15] Sohn,K., Yan, X., and Lee, H. Learning structured output representation using deep conditional generative models. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS'15, page 3483-3491, Cambridge, MA, USA, MIT Press, 2015.

[Su15a] Su, H. et al. Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE international conference on computer vision, pp. 945-953, 2015.

[Tch19a] Tchapmi, L. et al. Topnet: Structural point cloud decoder. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 383-392, 2019.

[Ugu19] Ugur, M., and Keysan, O. Multi-physics design optimisation of a GaN-based integrated modular motor drive system. In: The Journal of Engineering 2019.17, pp. 3900-3905, 2019.

[Yan16a] Yan, X. et al. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In: Advances in neural information processing systems 29, 2016.

[Ye22] Ye, M., Wu, L., and Liu, Q. First hitting diffusion models. In: arXiv preprint arXiv:2209.01170 (2022).

[Yon21] Yonekura, K., and Suzuki, K. Data-driven design exploration method using conditional variational autoencoder for airfoil design. In: Structural and Multidisciplinary Optimization 64.2, pp. 613-624, 2021.

[Zha19] Zhang, W. et al. 3D shape synthesis for conceptual design and optimization using variational autoencoders. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Vol. 59186. American Society of Mechanical Engineers, 2019.

[Zhe22] Zheng, Y. et al. Neural volumetric mesh generator. In: arXiv preprint arXiv:2210.03158, 2022.

Adapting Virtual Reality Training Applications by Dynamically Adjusting Visual Aspects

Fabio Genz and Dieter Kranzlmüller

MNM-Team, LMU München, Oettingenstr. 67, 80538 München, Germany

fabio.genz@nm.ifi.lmu.de

ABSTRACT

The present work addresses the question how to design a Virtual Reality (VR) training application for a class of search and navigation tasks that dynamically adapts to users by adjusting visual aspects for visual guidance. We present a theoretical concept that dynamically adjusts visual aspects (lighting, colour) in virtual environments (VE) based on a combination of measuring user behaviour (head position, head orientation) and training performance (training time, error rate). The concept is build on derived requirements for a VR application that trains search and navigation tasks and a combination of previous approaches to meet them. A proof-of-concept (PoC) application was implemented, training order picking of parcels in a warehouse. Though the presented concept is sound as it is based on previous research, future work should conduct user studies to validate the concept with quantitative data.

Keywords

Virtual Reality Training, Adaptive Training, Visual Guidance, Human-Computer-Interaction

1 INTRODUCTION

Trainees' needs naturally differ from each other based on different knowledge, skills, demographic, socio-cultural background and learning rates [SZ12; Get84]. Static and predefined trainings are potentially too easy resulting in boredom, or too difficult resulting in anxiety and bad performance [ZA20]. Both affect training negatively [Pek+10; YLC18; Cha+08]. Adaptive training approaches counter these problems and enable more efficient training [Kel69; SZ12].

Finding the right balance between the difficulty of training tasks and skills of trainees has historically been the responsibility of human teachers and tutors [PL04]. Much research has been done to replicate the benefits of human's adaptive teaching capabilities in automatic, computer-based adaptive training [Lan+12].

One way to adapt training difficulty is by providing targeted assistance in form of visual guidance [Van13; GM20]. The extent of this targeted assistance is subsequently referred to as the level of assistance (LOA). Literature provides different approaches by adding visual indicators (e.g. arrows [Gru+18a; Lin+17], attention tunnels [Bio+07; SK08] halos and wedges [Gru+18a]) or changing visual aspects of a VE (e.g. lighting [RH18], blurring [ST13; HKS16]).

VR seems particularly suitable for adaptive training approaches and visual guidance for two reasons. First, modern Head Mounted Displays (HMD) are often equipped with a range of different sensors measuring data serving as a potential basis for adaptations [Nar+21]. Second, trainees are able to move and look around freely in a VE that is completely customisable.

The present work addresses the question how to design a VR training application for a class of search and navigation tasks that dynamically adapts to users by adjusting visual aspects of the VE for visual guidance.

Although literature examined a wide range of approaches for visual guidance, there is currently no combined approach for an adaptive VR training application that dynamically adjusts visual aspects (lighting, colour) in VE based on a mix of measuring user behaviour (head position, head orientation) and training performance (training time, error rate).

The class of search and navigation tasks is described in Section 2.1. Requirements for designing a respective theoretical concept are derived in Section 2.2. We review literature for relevant approaches which fulfil at least parts of the derived requirements in Section 3. We combined relevant previous approaches in an overarching theoretically sound concept, presented in Section 4. A respective PoC implementation is shown in Section 5. In Section 6 we critically reflect on the results and discuss limitations, followed by a conclusion and outlook on potential future work in Section 7.

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.

https://www.doi.org/10.24132/CSRN.3401.23

2 REQUIREMENT ANALYSIS

In this Section, we describe the class of search and navigation tasks in Section 2.1. Requirements for designing a respective theoretical concept for a VR training application that dynamically adapts to users by adjusting visual aspects for visual guidance are presented in Section 2.2.

2.1 Training Scenario

We focus on training search and navigation task for single users, as these activities play central roles in different job profiles and offer a wide range of applications. In picker-to-parts-systems for example, which are the most common form of order picking, products need to be manually retrieved from the product storage by the order picker. Such job profiles are mainly low-skilled [OEC22; KLR07]. The training scenario essentially consists of two parts. First, search and navigate to a requested item which is located in a specific region of origin. Second, transporting the requested item to a specific destination.

2.2 Requirements

Based on the scenario, we derived five key requirements for a suitable training application for search and navigation tasks described below.

- ***Virtual Reality.*** VR is of special suitability due to the already in the Introduction described reasons. Furthermore, the application should be as widely applicable as possible
- ***Training Content.*** The goal is to find a certain item in a specific region of origin and transport it to a specific destination. Potential errors arise from selecting a wrong item or storing it at the wrong destination.
- ***Environment Interaction.*** The trainee requires possibilities for moving in the training environment and interact with the items to solve the training task.
- ***Adaptive Training.*** The training application requires adaption of the training difficulty to the trainee in the context of individual needs to enable efficient learning experiences [Kel69].
- ***Visual Guidance.*** The following requirements must be met to adapt training difficulty by using visual guidance for targeted assistance.
  - ***Draw Trainee's Attention***
    Able to attract learner's attention.
  - ***Dynamically Adaptable at Runtime***
    Able to adjust visual aspects in the VE dynamically at runtime without recognisable delay.
  - ***Adjustable in Different Level of Assistance***
    Visualisation of even small differences in the LOA.

3 RELATED WORK

In Section 3.1 we describe adaptive training approaches and address the key components of such approaches in the context of VR training. In Section 3.2, we reflect on previous approaches of visual guidance in VR training.

3.1 Adaptive Virtual Reality Training

There are various definitions of adaptive learning (see, e.g. Landsberg et al. [Lan+12]). The overall goal is to "meet the needs of individualized learners" [Vog+16]. We follow the definition of Kelley [Kel69] of a "training in which the problem, the stimulus, or the task is varied as a function of how well the trainee performs".

The advantages of adaptive training are justified by learning theories from the field of psychology (e.g. cognitive load theory [Swe88], Yerkes-Dodson law [YD+08], flow theory [Csi75]).

An adaptive training is a closed-loop feedback system consisting of three key components (see Figure 1) [Kel69]. Although only a small number of adaptive VR training approaches have been analysed in terms of their effectiveness, most of them have shown positive correlations [VGD16; ZA20].

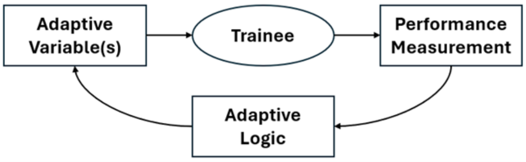


Figure 1. Illustration of an adaptive training cycle according to Kelley [Kel69].

In the following, we address the three key components of adaptive training approaches in the context of VR.

- *Adaptive Variable(s).* One or more adjustable variables that influence the user and their performance [Kel69]. There is a wide range of potential adaptive variables (see e.g. [ZA20; Lan+12]). With regard to the focus of the present work, we concentrate on attention guidance. Here, an overview of previous approaches can be found in Rothe et al. [RBH19]. While most approaches follow visual guidance, others follow haptic guidance [KR17; Cha+18], auditive guidance [Mar+19], guidance by forced rotation of the user's virtual body [Nie+16] or guidance of the physical body [Gug+16].
- *Performance Measurement.* Measurement data as the basis for determining the necessary LOA [Kel69]. Potential metrics can be task-related (e.g. performance during a training), related to physiological measurements (e.g. heart rate), kinematics (e.g. range of motion) [Lan+12; ZA20], the measurement of trainees skills, affective states,

and additional indicators such as personality traits or learning styles [SZ12].

- *Adaptive Logic.* Computation of the LOA based on the *Performance Measurement* to derive the corresponding adjustment of the *Adaptive Variable(s)* [Kel69]. The collected data can be processed with techniques such as machine learning, optimisation strategies, rule-based systems, or conditional statements [ZA20]. Multiple approaches have been identified with adaptions prior to the training, during the training or, combined in a two-step approach, with adaptions prior and during the training [PL04; Vog+16; Lan+12]. Continuous assistance during a training leads to over-dependence, which results in lower training performance without assistance [TGH15].

3.2 Visual Guidance in Virtual Reality

Visual guidance approaches often add visual indicators to the user's field of view like specific pointers towards relevant areas of the VE. These have been investigated in form of arrows [Gru+18b] [Lin+17], attention tunnels [Bio+07; SK08] or halos and wedges [Gru+18a]. Other indicators visualize the location of relevant areas in relation to the user's current position and orientation [Bor+18]. According to Bork et al. [Bor+18], these approaches can have the problem of adding visual clutter and occupying huge amounts of screen space, which is a limited commodity in current HMDs.

Other approaches of visual guidance suggest changing visual aspects of the VE to guide the user's attention. Here, previous literature examined the use of lighting [RH18], the blurring of irrelevant sections of the VE [ST13] [HKS16], slight visual changes to the VE based on saliency maps [MFS10] [Vea+11] or modulations of luminance at the edge of the user's vision [Bai+09] [Gro+17]. According to El-Nasr et al. [El-+09] visual guidance in form of lighting can reduce the time for a user to search and find enemies in a respective game [El-+09]. Boggus and Crawfis [BC10] show that visual guidance in form of lighting can reduce the time to navigate within the VE.

The previous literature provides a wide range of approaches for visual guidance in VR, even in the context of adjusting visual aspects. Although a number of approaches provide potential answers to parts of our derived requirements, there is currently no combined approach fulfilling them all. In the next section we therefore present an appropriate approach.

4 CONCEPT

In this Section we present a concept for a VR training application for a class of search and navigation tasks that dynamically adapts to users by adjusting visual aspects (lighting, colour) in VE for visual guidance. We base the adjustments on a mix of measuring user behaviour (head position, head orientation) and training performance (training time, error rate).

We build the concept on the derived requirements from Section 2.2 which we meet with a mix of previous approaches from literature from Section 3. We therefore consider the concept as theoretically sound.

To meet the requirements of ***Virtual Reality*** our concept is designed for HMDs with head motion tracking in six degrees of freedom, a common capability [Dör+19].

To meet the requirements of ***Adaptive Training***, our concept consist of three key components following Kelley [Kel69].

In Section 4.1 we describe the concept meeting the requirements for a class of training of search and navigation tasks. Regarding the three key components, the adaptive variables are described in Section 4.2, the performance measurements are described in Section 4.3 and the adaptive logic is described in Section 4.4. The concept covers all five requirements which we will explain in detail.

4.1 Training Search and Navigation

To meet the requirement of ***Training Content***, our concept suggests the following content for training search an navigation tasks.

- **Item Requests**
  Generation and displaying of item requests.
- **Specific Destination(s)**
  Specific destination(s) for selected items.
- **Region of Origin**
  Different regions of origin with various items.
- **Display Results**
  Discerning correct and wrong delivered items.

To meet the requirement of ***Environment Interaction***, our concept suggests the following interaction and movement possibilities for the training of search an navigation tasks.

- **Locomotion**
  Locomotion by transferring physical movements from the real world to the VE and by teleportation.
- **Object Manipulation**
  Possibility to interact with designated items.

4.2 Adjustable Visual Aspects

We suggest and assume the use of common game engines (e.g. Unity [22b], Unreal Engine [22c]) for the design of VEs. To meet the requirements of ***Visual Guidance***, our concept suggests the adjustment of the following two visual aspects as adaptive variables.

- **Object Colour**
- **Lighting Intensity**

The requirement to ***Draw Trainee's Attention*** is met, as colour has been shown to attract direct attention [WH17], and lighting has been shown to draw attention to the area around the light source [Vin+09]. The primary colours red, green and blue are potential candidates as colours stand out more clearly that are closer to being a primary colour [BK17].

The intensity of the attention guiding effect depends on the degree to which the region of origin is visually distinguishable from the non-region of origin [WH17] [DH89].

The requirement of ***Dynamically Adaptable at Runtime*** is met by using the functionalities of common game engines enabling to adjust the material properties and thus the object colour, and lighting at runtime.

The requirement of ***Adjustable in Different Level of Assistance*** is met, by using common game engines where object colour and lighting intensity are stored in form of floating point values and therefore finely adjust- and visualisable.

4.3 Behaviour and Performance

Our concept suggests to distinguish between measuring the behaviour (*Training Behaviour*) and the performance of the trainee (*Training Performance*) with two metrics each.

Training Behaviour is measured by the two metrics:

- **Distance to the Task Goal.** This metric measures the length of a straight line drawn from the trainee's head position to the region of origin. It indicates how close the trainee is to physically reaching the region of origin.
- **Angular Distance to the Task Goal.** This metric measures the angle between the trainee's current head orientation and the necessary orientation to face the region of origin. It indicates how close the trainee is to visually finding the region of origin.

Training Performance is measured by the two metrics:

- **Training Time** This metric measures the time it takes to complete a training session. We suggest to start measuring when a training session begins until all objectives of the training task have been achieved. A shorter task execution time indicates a higher level of competence.
- **Number of Errors** This metric measures the number of errors made during a training session. A lower amount of errors indicates a higher level of competence.

4.4 Adaptive Logic

The adaptive logic contains two steps. First, *Determining the LOA* based on the collected data from *Behaviour and Performance* which will be described in Section 4.4.1. Second, *Adjusting the Visual Aspects* according to the determined LOA which will be described in Section 4.4.2.

4.4.1 Determining the Level of Assistance

Two values each for *Behaviour and Performance* (as described in Section 4.3) determine the LOA.

First, the *Training Behaviour* values define the **Needed LOA**, which in turn quantifies the current need for assistance during a *Training Round*. This LOA is adjusted at runtime.

Second, the **Maximum LOA** is based on *Training Performance*. The upper bound of assistance, hence Maximum LOA, is intended to prevent the adaptive logic from providing more assistance than justified by the trainee's performance. This limit is adjusted after each *Training Round*.

Both determined LOAs are in the range from 0 to 1, with the respective range of no assistance to full assistance. As the circumstances of training scenarios can differ, it is essential to define the two boundary values for every scenario. Whether values are considered high or low must therefore be defined on an individual basis.

The concept follows Landsberg et al. [Lan+12] with a two-step approach, containing adaptions prior and during the training. Hence, the training consist of at least two rounds. In the first training round, the Maximum LOA is initially set to 1.

The following steps are subsequently executed to determine the LOA:

1. **Calculating the Needed LOA**
   We assume that the further away the trainee is from reaching the task goal, measured by the *Training Behaviour* values, the greater is the need for assistance. The *Needed LOA* is calculated by averaging the normalised values of *Distance to the Task Goal* and *Angular Distance to the Task Goal*.

2. **Adjusting the Maximum LOA**

   The *Maximum LOA* is adapted to the trainee's changing abilities. The adjustments are based on the relative change in the *Training Performance*.

   As performance increases, the *Maximum LOA* is reduced to avoid boredom. If performance decreases, the *Maximum LOA* is increased to avoid anxiety. Adjustments are made after each training round and correspond to the relative change in training performance.

   The relative change is determined by subtracting the prior training performance from the current training performance.

4.4.2 Adjusting the Visual Aspects

After deriving the LOA calculated by the adaptive logic, the visual aspects are respectively adjusted. We subsequently describe both adjustable visual aspects in Section 4.2.

Lighting Intensity. Two conditions must be met when using light intensity as an adjustable visual aspect. First, the VE must be illuminated by different, separately adjustable light sources, with at least one light source per potential region of origin. Secondly, the potential regions of origin must be illuminated as uniformly as possible at the beginning to ensure changes in light intensity are perceivable to guide trainee's attention.

Adjusting light intensity according to the LOA in form of visually perceptible differences involves reducing the light intensity of the light sources that illuminate the non-regions of origin while increasing the light intensity of the light source that illuminates the region of origin.

Object Colour. To use object colour as an adjustable visual aspect, each possible region of origin must contain largely similar items with similar surface colours. This is necessary to avoid large visual differences between the region of origin and the non-region of origin in the starting point.

To guide trainees' attention, the surface colour of one or more selected items in the region of origin is highlighted according to the LOA, while the surface colour of the items outside the region of origin remain the same.

5 IMPLEMENTATION

This Section describes a PoC implementation of the previously presented concept in form of a VR application that trains search and navigation tasks by training order picking of parcels in a warehouse. The PoC covers all elements of the concept, which we will explain in detail.

The VR training application is presented in Section 5.1. Instead of repeating the already in Section 4 described three key components of an adaptive approach, we go into relevant details of the application of the concept. The boundary values of measuring behaviour and performance are defined in Section 5.2.

In Section 5.3 we present exemplary visual guidance in form of adjusting lighting intensity and colour for three different LOA each.

In Section 5.4 we present a system overview regarding the used hard- and software.

5.1 Training Order Picking of Parcels

Down below, the training task, VE and environment interaction are described.

5.1.1 Training Task

We choose order picking in a warehouse as the training task. Order picking refers to the process of "retrieving products from storage in response to a specific customer request" **(Item request)** and transporting it to a specific destination **(Specific Destination)**.

Four regions of the VE are defined as possible regions of origin, one for each group of three shelves **(Region of Origin)**. A display shows weather the correct parcel was delivered **(Display Results)**. See Figure 3 for respective illustrations. As the position of the destination is defined and known, changing visual aspects are intended to assist in finding the parcels in the regions of origin.

Therefore, the PoC contains all four elements for content of training search and navigation tasks from our concept in Section 4.1.

5.1.2 Environment Interaction

There are two possibilities for locomotion. First, participants can move in the real world, with their movements being replicated in the VE. Second, participants can teleport within the VE. By pressing a button on the controller, the possible movement is displayed. When the button is released, the trainee is teleported there **(Locomotion)**.

Interactions with parcels are possible by pressing the grip button on the controller as soon as a virtual hand crosses a parcel. After pressing the grip button, the parcel is fixed in the position of the learner's hand until the button is released again. Each interaction is defined for the left and right controller, enabling the trainee to perform each interaction with both hands. The self-representation is limited to the virtual representation of the hands **(Object Manipulation)**. See Figure 2 for respective illustrations.

The PoC contains both elements of interaction and movement possibilities for training search and navigation tasks from our concept in Section 4.1.

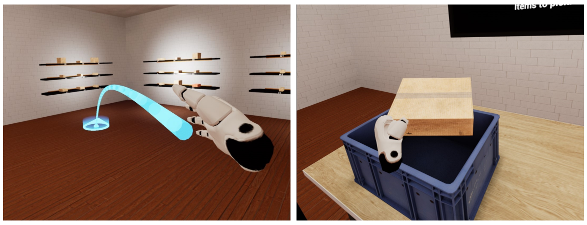


Figure 2. Illustration of a potential teleportation (left) and the interaction with a parcel by transporting it to the destination (right) [Lei21]

5.1.3 Virtual Environment

We implemented a simple designed warehouse for the training environment. It consists of a square room, with four groups of three shelves each attached to the walls of the rear half of the room as item storage. Each shelf contains three spaces for storing items. Each storage location is randomly filled with a cardboard box of different sizes, which serves as the stored items. To enable textual representations of the product locations for customer enquiries, each storage location is labelled alphanumerically with a letter followed by a number. The letter clearly identifies the shelf on which the product is located. The number indicates where the product is located on this shelf. On the other side of the room, a table with a sufficiently large crate has been set up, which is intended as the destination for the requested products. A large monitor on the wall behind the table enables the visual display of customer requests and other relevant information. See Figure 3 for respective illustrations.

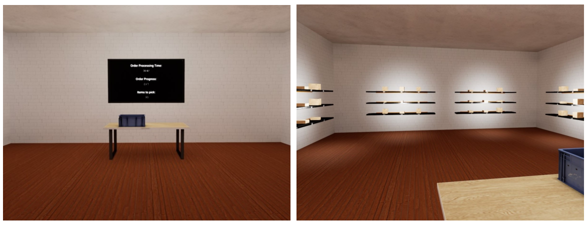


Figure 3. Illustration of the front side (left) and back side of the warehouse (right) [Lei21]

5.2 Setting the Boundary Values

By measuring *Training Behaviour* and *Training Performance*, the PoC contains all suggested measurements from our concept in Section 4.3.

As described in Section 4.4, both values from *Behaviour and Performance* are mapped between 0 and 1 and need to be defined individually for a training scenario. For the PoC we defined the following, based on performance observations during the development.

Regarding *Training Behaviour*, for the **Distance to the Task Goal** a small distance is defined as 0.5 meters, while a large distance is defined as 10 meters (e.g. wall on the other side of the room).

For the **Angular Distance to the Task Goal.** a small angular distance is defined as 0°, since at this angle, the trainee's head is oriented right at the region of origin. The definition of a large angular distance is closely related to the used HMD, respectively the field of view.

In Section 5.4, we go into more detail about the used hard- and software. However, as the used HMD (Oculus Rift S) has been measured with 94.25° diagonally [MMR20] any object further than 47.125° from the centre, won't be visible for the trainee anymore. We therefore defined 40° as a large angular distance.

Regarding *Training Performance*, for the **Training Time** a sufficient performance is defined as ≤ 5 seconds and an insufficient is defined as ≥ 15 seconds per requested parcel. For the **Number of Errors** a sufficient performance is defined as ≤ 1 error and an insufficient is defined as ≥ 1 error per requested parcel.

5.3 Adjusting Visual Aspects

In this Section we present the adjustable visual aspects *Lighting Intensity* as well as *Object Colour*. The PoC contains all relevant elements for both visual aspects from our concept in Section 4.2 as well as the adaptive logic by respective adjustments as described in Section 4.4.

5.3.1 Adjusting Colour

In this section we present exemplary visual guidance in form of adjusting colour for three different LOA. See Figure 4 for respective illustrations.

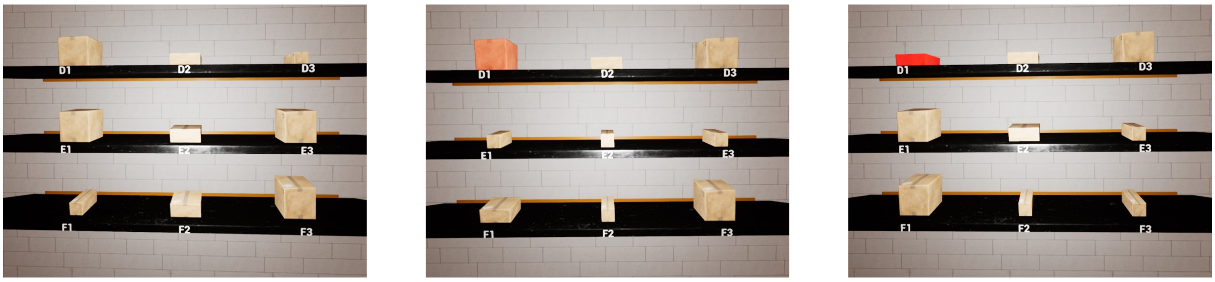
For the LOA of 0, the surface colour for each object remains at its base colour. It is provided by a texture image to make it look like a cardboard box. Since all products use the same texture image, there is no visual difference between the region of origin and non-regions of origin.

With an LOA of 1, the surface colour of the selected object is set to the target colour. Thus, the region of origin is clearly distinguishable.

For LOAs between 0 and 1, the surface colour of the region of origin is interpolated accordingly. To prevent the target colour from differing too much from the base colour even at a small interpolation, we use red as the target colour. This primary colour comes closest to the light brown basic colour of the base colour. In addition, red is often used as a warning colour, which can attract a lot of attention if required.

5.3.2 Adjusting Lighting Intensity

In this section we present exemplary visual guidance in form of adjusting lighting intensity for three different LOA. See Figure 5 for respective illustrations.



(a) Level of Assistance: 0 **(b) Level of Assistance: 0.5** **(c) Level of Assistance: 1**

Figure 4. Examples of assistance by adaptation of colour [Lei21]

In accordance with the requirements for the use of light intensity described in Section 4.4.2, each potential region of origin (shelf group) has an adaptive light source. The attenuation radius (radius in which the light source acts on the VE) is limited to ensure that, in case of a certain LOA, the adaptive light sources only illuminate the assigned shelf group, while the other adaptive light sources diminish. Basic lighting is provided for the other areas of the VE that are not illuminated by adaptive light sources. Light intensity is measured in candela [NPL24].

The intensity for basic lighting is set to 2.5 candela. For the LOA of 0, the lighting intensity is set to basic lighting for all potential regions of origin. With a LOA of 1, lighting intensity is set to 5 candela for the specific region of origin and 0 candela for all other non-regions of origin. For LOAs between 0 and 1, the surface colour of the region of origin and non-regions of origin are interpolated accordingly.

5.4 System Overview

We used the Oculus Rift S an HMD offering 6 degrees of freedom and inside-out tracking from five cameras mounted on the outside of the HMD. It has a resolution of 1280x1440 pixels per eye and speakers mounted in the headband above the user's ears and needs to be connected to a desktop PC. The HMD is controlled by two (right and left) *Oculus Touch controllers* of the second generation as input devices [22a]. For the implementation we used *Unreal Engine 4.27*, a suite of creation tools for the purposes of game development and architectural visualisations [22c] and the programming language *C++*.

6 RESULTS AND DISCUSSION

In this Section, we present and discuss the results.

6.1 Results

The major contribution of this work is the presented theoretical concept. To create the concept for the selected class of training applications, we first derived five core requirements. We then conducted a literature review to examine previous approaches to be able to fulfil at least partial requirements.

Although literature examined a wide range of approaches for visual guidance, there is currently no combined approach for an adaptive VR training application that dynamically adjusts visual aspects (lighting, colour) in VE based on a mix of measuring user behaviour (head position, head orientation) and training performance (training time, error rate).

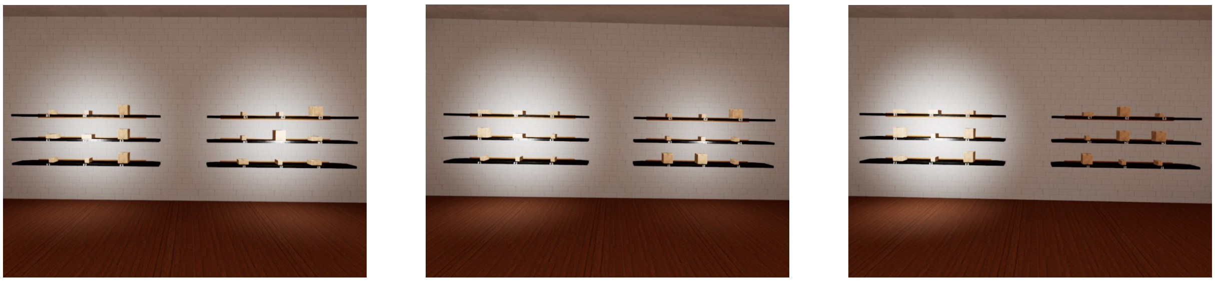
The presented concept meets all five key requirements. As it basically consists of a combination of previous approaches, we consider it to be theoretically sound.

An additional contribution can be seen in the creation of the PoC implementation.

6.2 Discussion

Looking at the presented results, we have to mention some limitations. Above all, the theoretical soundness we assume is based on the validity of the approaches we combined from literature. Although these are certainly valid when considered individually, their validity in combination is subject to an appropriate validation, e.g. by user studies. Here, the presented PoC implementation emphasises its relevance for future research.

Furthermore, the effect of the presented concept on training effectivity can be influenced by several factors. The first factor concerns the determination of the correct LOA. Based on simple measurements and algorithms, there is a risk that the adaptation to the trainee's needs are inadequate. The second factor concerns the effect of the proposed adjustable visual aspects. It is reasonable to assume that these have different effects on the guidance of visual attention. Corresponding evaluations must clarify the respective influence and be taken into account accordingly in the extent of the adjustments. The third factor concerns possible technological challenges in form of sufficient computing power, e.g. to prevent motion sickness for the user. In the context of near-real-time adjustments of visual aspects, this requires the maintenance of high frame rates, which requires fast and efficient graphical



(a) Level of Assistance: 0 **(b) Level of Assistance: 0.5** **(c) Level of Assistance: 1**

Figure 5. Examples of assistance by adaptation of lighting intensity [Lei21]

rendering. The needed performance in turn varies with the selected adjustable visual aspects.

We also see a potential limitation regarding object colour as an adjustable visual aspect, since all objects must have a comparable base colour to enable visual distinguishability according to the LOA.

7 CONCLUSION AND OUTLOOK

In this section, we present a conclusion and outlook on future work.

7.1 Conclusion

This paper investigates how to design a VR training application for a class of search and navigation tasks that dynamically adapts to users by adjusting visual aspects for visual guidance. Building on a requirement analysis and literature review, a theoretically sound concept and a respective PoC implementation is presented.

As there is no comparable approach in previous literature, the approach contains a certain independence and novelty. As the theoretical soundness however, is derived from the fact that the approach basically consists of a combination of previous approaches, further validation is necessary (e.g. by user studies).

Since, in addition to the theoretical concept, a corresponding PoC implementation is presented, important prerequisites as well as concrete suggestions for promising future research are given.

7.2 Outlook

We see two approaches for future work. First, since the presented concept is build on previous approaches from literature, we consider the concept as theoretically sound. However, we propose to validate it with quantitative data by conducting user studies.

Second, we propose to examine the integration of physiological measurement data as performance measurement. Modern HMDs (e.g. Varjo [Var24]) are often equipped with a variety of sensors on board (e.g. eye tracking). Physiological measurement data contain information about the physical, emotional and cognitive states of the user and therefore offer the potential to adapt training to the user [HL21; Mar+20].

ACKNOWLEDGMENTS

We thank Daniel Leidreiter for his support in this work, the MNM Team, especially Vitalian Danciu, Fabian Dreer, Sophia Grundner-Culemann, Daniel Diefenthaler and David Linder who contributed valuable discussions and feedback.

REFERENCES

[22a] *Oculus-Headsets*. Meta Platforms, 2022.

[22b] *Unity - Manual*. Unity Technologies, 2022.

[22c] *Unreal Engine - Manual*. Epic Games, 2022.

[Bai+09] R. Bailey et al. “Subtle Gaze Direction”. In: *ACM Transactions on Graphics* 28.4 (2009).

[BC10] M. Boggus and R. Crawfis. “Distance Field Illumination: A Rendering Method to Aid in Navigation of Virtual Environments”. In: *Advances in Visual Computing*. Ed. by G. e. a. Bebis. Lecture notes in computer science. Berlin, Heidelberg: Springer, 2010, pp. 501–510.

[Bio+07] F. Biocca et al. “Attention Issues in Spatial Information Systems: Directing Mobile Users’ Visual Attention Using Augmented Reality”. In: *Journal of Management Information Systems* 23.4 (2007), pp. 163–184.

[BK17] A. Butz and A. Krüger. *Mensch-Maschine-Interaktion*. 2., erweiterte Auflage. De Gruyter Studium. Berlin and Boston: De Gruyter Oldenbourg, 2017.

[Bor+18] F. Bork et al. “Towards Efficient Visual Guidance in Limited Field-of-View Head-Mounted Displays”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.11 (2018), pp. 2983–2992.

[Cha+08] G. Chanel et al. “Boredom, Engagement and Anxiety as Indicators for Adaptation to Difficulty in Games”. In: *MindTrek ’08: Proceedings of the 12th International Conference on Entertainment and Media in the Ubiquitous Era*. Ed. by A. e. a. Lugmayr. New York, NY, USA: Association for Computing Machinery, 2008, pp. 13–17.

[Cha+18] H.-Y. Chang et al. “FacePush: Introducing Normal Force on Face with Head-Mounted Displays”. In: *UIST ’18: Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 927–935.

[Csi75] M. Csikszentmihalyi. *Beyond boredom and anxiety: Experiencing flow in work and play.* Jossey-bass, 1975.

[DH89] J. Duncan and G. W. Humphreys. “Visual search and stimulus similarity”. In: *Psychological Review* 96.3 (1989), pp. 433–458.

[Dör+19] R. Dörner et al. *Virtual und Augmented Reality (VR / AR)*. 2., erweiterte und aktualisierte Aufage. Berlin: Springer Vieweg, 2019.

[El-+09] M. S. El-Nasr et al. “Dynamic Intelligent Lighting for Directing Visual Attention in Interactive 3-D Scenes”. In: *IEEE Transactions on Computational Intelligence and AI in Games* 1.2 (2009), pp. 145–153.

[Get84] M. Gettinger. “Individual differences in time needed for learning: A review of literature”. In: *Educational Psychologist* 19.1 (1984), pp. 15–29.

[GM20] S. Grogorick and M. Magnor. “Subtle Visual Attention Guidance in VR”. In: *Real VR – Immersive Digital Reality*. Ed. by M. Magnor and A. Sorkine-Hornung. Lecture Notes in Computer Science. Cham: Springer, 2020, pp. 272–284.

[Gro+17] S. Grogorick et al. “Subtle gaze guidance for immersive environments”. In: *SAP ’17: Proceedings of the ACM Symposium on Applied Perception*. Ed. by S. N. Spencer. New York, NY, USA: Association for Computing Machinery, 2017.

[Gru+18a] U. Gruenefeld et al. “Beyond Halo and Wedge: Visualizing out-of-View Objects on Head-Mounted Virtual and Augmented Reality Devices”. In: *MobileHCI ’18: Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services with Mobile Devices and Services*. New York, NY, USA: Association for Computing Machinery, 2018.

[Gru+18b] U. Gruenefeld et al. “FlyingARrow: Pointing Towards Out-of-View Objects on Augmented Reality DevicesReality”. In: *PerDis ’18: Proceedings of the 7th ACM International Symposium on Pervasive Displays*. Ed. by A. e. a. Schmidt. New York, NY, USA: Association for Computing Machinery, 2018.

[Gug+16] J. Gugenheimer et al. “SwiVRChair: A Motorized Swivel Chair to Nudge Users’ Orientation for 360 Degree Storytelling in Virtual Reality”. In: *CHI ’16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2016, pp. 1996–2000.

[HKS16] H. Hata, H. Koike, and Y. Sato. “Visual Guidance with Unnoticed Blur Effect”. In: *AVI ’16: Proceedings of the International Working Conference on Advanced Visual Interfaces*. Ed. by P. e. a. Buono. New York, NY, USA: Association for Computing Machinery, 2016, pp. 28–35.

[HL21] A. Halbig and M. E. Latoschik. “A Systematic Review of Physiological Measurements, Factors, Methods, and Applications in Virtual Reality”. In: *Frontiers in Virtual Reality* 2 (2021).

[Kel69] C. R. Kelley. “What is Adaptive Training?” In: *Human Factors* 11.6 (1969), pp. 547–556.

[KLR07] R. de Koster, T. Le-Duc, and K. J. Roodbergen. “Design and control of warehouse order picking: A literature review”. In: *European Journal of Operational Research* 182.2 (2007), pp. 481–501.

[KR17] O. B. Kaul and M. Rohs. “HapticHead: A Spherical Vibrotactile Grid around the Head for 3D Guidance in Virtual and Augmented Reality”. In: *CHI ’17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 3729–3740.

[Lan+12] C. R. Landsberg et al. “Review of Adaptive Training System Techniques”. In: *Military Psychology* 24.2 (2012), pp. 96–113.

[Lei21] D. Leidreiter. “Personalised Guidance for Virtual Reality Training by means of self-adaptive Environment Variables”. Bachelor thesis. Ludwig-Maximilians-Universität München, 2021.

[Lin+17] Y.-C. Lin et al. “Tell Me Where to Look: Investigating Ways for Assisting Focus in 360° Video”. In: *CHI ’17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 2535–2545.

[Mar+19] A. Marquardt et al. “Non-Visual Cues for View Management in Narrow Field of View Augmented Reality Displays”. In: *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. Institute of Electrical and Electronics Engineers, 2019, pp. 190–201.

[Mar+20] J. Marín-Morales et al. “Emotion Recognition in Immersive Virtual Reality: From Statistics to Affective Computing”. In: *Sensors (Basel, Switzerland)* 20.18 (2020).

[MFS10] E. Mendez, S. Feiner, and D. Schmalstieg. “Focus and Context in Mixed Reality by Modulating First Order Salient Features”. In: *Smart Graphics*. Lecture Notes in Computer Science.

Berlin, Heidelberg: Springer, 2010, pp. 232–243.

[MMR20] S. Mareck, V. Macedo, and C. Runde. “Head-Mounted Displays: Messung des Sichtfelds (Field of View) | Measurement and Comparison of VR headsets’ Field of View”. In: (2020).

[Nar+21] D. Narciso et al. “A systematic review on the use of immersive virtual reality to train professionals”. In: *Multimedia Tools and Applications* 80.9 (2021), pp. 13195–13214.

[Nie+16] L. T. Nielsen et al. “Missing the Point: An Exploration of How to Guide Users’ Attention during Cinematic Virtual Reality”. In: *VRST ’16: Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. New York, NY, USA: Association for Computing Machinery, 2016, pp. 229–232.

[NPL24] NPLWebsite. *candela (cd)*. 15.04.2024.

[OEC22] OECD. *OECD Social, Employment and Migration Working Papers*. Vol. 282. OECD, 2022.

[Pek+10] R. Pekrun et al. “Boredom in achievement settings: Exploring control–value antecedents and performance outcomes of a neglected emotion”. In: *Journal of Educational Psychology* 102.3 (2010), pp. 531–549.

[PL04] O.-c. Park and J. Lee. “Adaptive instructional systems”. In: *Handbook of Research on Educational Communications and Technology*. Ed. by D. Jonassen. Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers, 2004, pp. 651–684.

[RBH19] S. Rothe, D. Buschek, and H. Hußmann. “Guidance in Cinematic Virtual Reality-Taxonomy, Research Status and Challenges”. In: *Multimodal Technologies and Interaction* 3.1 (2019).

[RH18] S. Rothe and H. Hußmann. “Guiding the Viewer in Cinematic Virtual Reality by Diegetic Cues”. In: *Augmented Reality, Virtual Reality, and Computer Graphics*. Lecture Notes in Computer Science. Cham: Springer, 2018, pp. 101–117.

[SK08] B. Schwerdtfeger and G. Klinker. “Supporting order picking with Augmented Reality”. In: *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. Ed. by M. A. e. a. Livingston. Institute of Electrical and Electronics Engineers, 2008, pp. 91–94.

[ST13] W. S. Smith and Y. Tadmor. “Nonblurred regions show priority for gaze direction over spatial blur”. In: *Quarterly Journal of Experimental Psychology* 66.5 (2013), pp. 927–945.

[Swe88] J. Sweller. “Cognitive load during problem solving: Effects on learning”. In: *Cognitive science* 12.2 (1988), pp. 257–285.

[SZ12] V. J. Shute and D. Zapata-Rivera. “Adaptive Educational Systems”. In: *Adaptive Technologies for Training and Education*. Ed. by P. J. Durlach and A. M. Lesgold. Cambridge: Cambridge University Press, 2012, pp. 7–27.

[TGH15] J. G. Tullis, R. L. Goldstone, and A. J. Hanson. “Scheduling Scaffolding: The Extent and Arrangement of Assistance During Training Impacts Test Performance”. In: *Journal of Motor Behavior* 47.5 (2015), pp. 442–452.

[Van13] Vanessa Paz Dennen. “Cognitive Apprenticeship in Educational Practice: Research on Scaffolding, Modeling, Mentoring, and Coaching as Instructional Strategies”. In: *Handbook of Research on Educational Communications and Technology*. Routledge, 2013, pp. 804–819.

[Var24] Varjo.com. *The world’s most advanced virtual and mixed reality devices*. 22.03.2024.

[Vea+11] E. E. Veas et al. “Directing Attention and Influencing Memory with Visual Saliency Modulation”. In: *CHI ’11: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2011, pp. 1471–1480.

[VGD16] N. Vaughan, B. Gabrys, and V. N. Dubey. “An overview of self-adaptive technologies within virtual reality training”. In: *Computer Science Review* 22 (2016), pp. 65–87.

[Vin+09] B. T. Vincent et al. “Do we look at lights? Using mixture modelling to distinguish between low- and high-level factors in natural image viewing”. In: *VISUAL COGNITION* 17.6-7 (2009), pp. 856–879.

[Vog+16] J. J. Vogel-Walcutt et al. “Improving the efficiency and effectiveness of adaptive training: using developmental models as a framework and foundation for human-centred instructional design”. In: *Theoretical Issues in Ergonomics Science* 17.2 (2016), pp. 127–148.

[WH17] J. M. Wolfe and T. S. Horowitz. “Five factors that guide attention in visual search”. In: *Nature Human Behaviour* 1.3 (2017).

[YD+08] R. M. Yerkes, J. D. Dodson, et al. “The relation of strength of stimulus to rapidity of habit-formation”. In: (1908).

[YLC18] J. C. Yang, M. Y. D. Lin, and S. Y. Chen. “Effects of anxiety levels on learning performance and gaming performance in digital game-based learning”. In: *Journal of Computer Assisted Learning* 34.3 (2018), pp. 324–334.

[ZA20] M. Zahabi and A. M. Abdul Razak. “Adaptive virtual reality-based training: a systematic literature review and framework”. In: *Virtual Reality* 24.4 (2020), pp. 725–752.

AI-based Density Recognition

Simone Müller
Leibniz Supercomputing Centre (LRZ)
simone.mueller@lrz.de

Daniel Kolb
Leibniz Supercomputing Centre (LRZ)
daniel.kolb@lrz.de

Matthias Müller
German Aerospace Center (DLR)
matthias.mueller@dlr.de

Dieter Kranzlmüller
Ludwig-Maximilians-Universität (LMU)
kranzlmueller@ifi.lmu.de

Abstract

Learning-based analysis of images is commonly used in the fields of mobility and robotics for safe environmental motion and interaction. This requires not only object recognition but also the assignment of certain properties to them. With the help of this information, causally related actions can be adapted to different circumstances. Such logical interactions can be optimized by recognizing object-assigned properties. Density as a physical property offers the possibility to recognize how heavy an object is, which material it is made of, which forces are at work, and consequently which influence it has on its environment. Our approach introduces an AI-based concept for assigning physical properties to objects through the use of associated images. Based on synthesized data, we derive specific patterns from 2D images using a neural network to extract further information such as volume, material, or density. Accordingly, we discuss the possibilities of property-based feature extraction to improve causally related logics.

Keywords

AI, Density Recognition, Computer Vision

1 INTRODUCTION

Modern machines and robots use various sensors to capture and navigate their surroundings. Particularly in road traffic, situations may appear inconspicuous at first sight but require constant attention and quick reactions. This can involve evaluating the potential risks in autonomous driving scenarios when a car not only recognizes objects but can also estimate the potential damage in the event of a collision and adapt its driving behavior accordingly. Additional information could increase a system's scope of action and decision-making as well as the automatic assessment of real-life scenes.

Whether it is a ball that rolls onto the road, a car that suddenly brakes, or an item that falls off a moving vehicle. Such reactions are often based on causal relationships that are logical for us humans but not for machines. Despite their logic, machines lack the necessary background knowledge and specific skills, such as the assessment of physical properties, to gain a causal understanding.

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.

Material recognition and the association of related properties can be helpful in causal decision-making [26]. For example, an industrial robot can apply the optimum force for gripping an inelastic object if it knows the approximate material, mass, roughness, and size of this object. All this information relates to the physical density and material property.

Machine learning offers solutions for material recognition [26]. Databases such as Flickr [14] are able to recognize different materials, as shown in Tab. 1.

Fabric Foliage Glass Leather Metal
Plastic Water Paper Wood Stone

Table 1: **State-of-the-Art Material Database [14].** Illustration of ten example material categories of Flickr database. The lighting conditions, compositions, colors, textures, surface shapes, material subtypes, and object associations were considered by an image diversity of 100 pictures, 50 close-ups, and 50 normal views in each category [1].

These patterns are recognized by visual features and stored in the model through a learning process, using sophisticated AI algorithms for the recognition of objects in 2D images, such as YOLOv3 [21], Faster Region-Based Convolutional Neural Networks (R-CNN) [22], and Multi-Scale Convolutional Neural Network (MSCNN) [3]. However, the previously trained 2D object recognition is limited to specific object classes and visual interference effects of images [17]. To collect ambient information and make accurate decisions with high confidence, the AI usually needs to be trained extensively with vast sets of data. Additionally, this material information has usually no connection with the associated physical properties.

Based on the challenges of accurate processing and linking causal information, we present an approach that enables the assignment of physical properties in objects based on a 2D image by using machine learning and pattern recognition. The object is extracted and scaled into triangles to estimate the volume. We derive the associated materials from a database and ultimately calculate the density as a physical quality.

AI-based recognition of density and volume provides a solid foundation for the extraction of additional information from the environment. As an example, object-related forces can be calculated based on equation-specific coefficients, constants, and acquired sensor data including density. This expands the information content in a visual scene. Especially in road traffic, this can be an additional aid to improve the perception of autonomous vehicles.

This paper describes a proof of concept for the implementation of AI-based density recognition. Our work comprises the following contributions:

- Neural-specific object and texture detection based on object classification
- Concept of AI-based density recognition
- Analysis of recognized object density and material composition

Our evaluation reveals the feasibility and transferability of AI-based density recognition. For our empirical examination, we use synthetically generated data from the Unreal Engine.

The paper is organized according to a fixed structure consisting of related work, concept, methodology of AI-based object and texture detection, evaluation, conclusion, as well as future work.

2 MODERN RECOGNITION

This section presents recognition models and existing approaches for physical property recognition. The basic idea involves material recognition, which gives rise to entire databases such as Flickr [14] which assigns materials based on visual appearance.

Liu et al. [14] describe that the visual appearance of a surface depends on illumination conditions, geometric structure of surfaces at different spatial scales, and reflectance properties. Thereby, the reflectance properties of the surface are often characterized by features with a bidirectional reflectance distribution function (BRDF) [18]. In this context, material recognition can employ the recognition of colors and textures, micro-textures, outline shapes, or reflectance-based features [14] by SIFT algorithms. This algorithm can recognize contours based on corners and edges [13].

Standard k-means algorithms Eq. 1 are used to cluster instances of each feature [14] in order to assign the image-specific materials M to the respective words.

$$J = \sum_{\mathrm{i}=1}^{k} \sum_{x_\mathrm{j} \in S_\mathrm{i}} ||x_\mathrm{j} - \mu_\mathrm{i}||^2 \tag{1}$$

S_i describes the cluster in Eq. 1, which is determined from data points x_j and centroids μ_i on the basis variance minimization [6] and squared Euclidean distance $||x_\mathrm{j} - \mu_\mathrm{i}||^2$. The random mean value k is determined in the visual data set m_i, ..., m_k. Each data object is assigned to the cluster with the lowest variance for all $l = [1, \ldots, k]$, shown in Eq. 2.

$$S_\mathrm{i} = \{x_\mathrm{j} : ||x_\mathrm{j} - m_\mathrm{i}||^2 \le ||x_\mathrm{j} - m_\mathrm{l}||^2\} \tag{2}$$

Machine learning can be used as a static method to learn continuously and specifically from experiences. Thereby, the training-based data is divided into different classes. The classification S_i permits the mapping of input variables $f_{\mathrm{i,x}} : \mathbb{R}^{n_\mathrm{i}} \rightarrow \mathbb{R}^{n_{\mathrm{i}-1}}$ to discrete output variables O. In this view, the regression $g_{(\mathrm{x})}$ refers to a distinction between more than two categories [15]. Fig. 1 summarises the related components of neural networks.

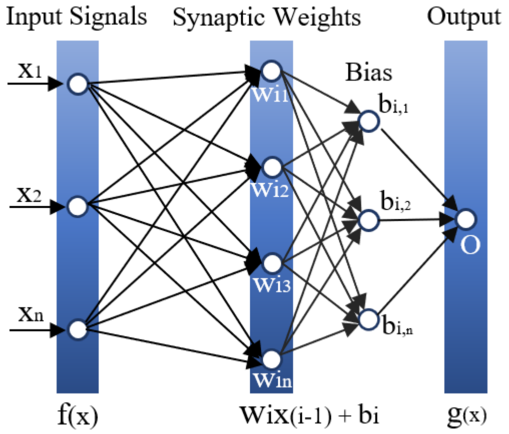


Figure 1: **Schematic Neural Network, adapted from [29].** The input signals $f(i,x)$:[x_1, .. , x_n] are weighted using the weighting factor w_i before they reach the main part of the neuron. A bias b_i is included as a threshold value which must first be exceeded to generate the output signal O.

Maschine learning techniques are broadly divided into two processes: feature extraction and classification. Feature extraction involves the identification and correlation of patterns with large datasets suitable for modeling. Thereby, a feature refers to a property derived from raw data input intending to provide a suitable representation [10]. In both cases, numerous processing nodes of neural networks are tightly interconnected and further layered in organized nodes to perform complex calculations [24].

The overall quality of learning-based feature extraction depends on the task area and associated data set. In this respect, the data must contain a high density of information. To differentiate between classes, the algorithms try to recognize existing patterns of colors, shapes, textures or pixel values within this information [9].

Girshick et al. [8] describe an approach in which high-capacity convolutional neural networks are applied to bottom-up region proposals for localization and segmentation. They introduce a paradigm for training large CNNs when labeled training data is scarce. They detail how pre-training the network with supervision on an auxiliary task with abundant data (image classification) and then fine-tuning the network on the target task where data is scarce (recognition) can improve overall efficiency. This approach is similar to R-CNN. In fast R-CNN, the CNN is first fed with the input image to generate a convolutional feature map. Subsequently, the selective search is performed and the region suggestions are warped into squares. Those region suggestions are called Regions of Interest (RoI) and refer to a subset of the original image [5]. By using RoI pooling layers, each region that has been proposed and which may have different sizes, is reshaped into fixed size so that it can be fed into a fully connected layer. On the output, a softmax layer is used to predict the class of the proposed region and the values of the bounding box [19]. This method produces results faster since it calculates the CNN features only once per image and not two thousand times as with the R-CNN method.

Sean Bell et al. [2] suggest in a direct comparison between three different CNN models (AlexNet, VGG-16, GoogleNet) that material recognition and segmentation of everyday images which are based on Materials in Context Database (MINC) is possible with a probability of 82.2 % (AlexNet) to 85.9 % (GoogleNet).

Shukla et al [25] evaluated the accuracy between CNN classifiers for material recognition and deep learning classifiers. They found that CNN classifiers have better and faster recognition accuracy since the existing probability level allows the classifier to recognize materials with higher accuracy.

Various previous works [3, 16, 21, 22] refer to AI algorithms such as YOLOv3, Faster R-CNN and MSCNN that use a class-verifying diversity of object propositions for object recognition.

Modern Architectures such as YOLO have been continuously improved to perform tasks in the areas of general and oriented recognition, instance segmentation, pose, key points and classification [12]. A DarkNet-19 model architecture (YOLOv2) was expanded to a more complex backbone model DarkNet-53 (YOLOv3) in which features on three different scales can be recognized [4]. Although the implementation of such new functions and targeted optimizations reduce latency times, they often require computationally intensive operations that demand considerable computing power.

Ren et al. [22] found a solution to an issue of R-CNN caused by selective search. Selective search is a rigid algorithm that is unable to improve or learn, which can lead to poor suggestions for candidate regions. They developed the Faster-R-CNN algorithm, replacing selective search with a separate network, the Region Proposal Network (RPN), to predict region proposals. The RPN takes an image as input and outputs a series of rectangular object proposals, each with a class prediction and a confidence value. The network can be trained throughout by backpropagation, where the gradient of the loss function is calculated taking into account the weights of the network for a single input-output example [23].

Wu et al. [28] teach a computational vision system to understand physical relationships with the help of unlabelled videos. They address specific physical scenarios and distinguish between two groups of physical properties: The first inherits the intrinsic physical properties of objects such as volume, material, and mass. The other group is the descriptive physical properties, which describe the scene and are determined by the first group. These include, but are not limited to, the speed of the objects, the distance they travel, or whether they fall into water. The presented model uses CNN to learn the object properties exclusively from unlabelled data. This approach provides serviceable results on a physical data set.

3 DENSITY RECOGNITION

This chapter describes the concept of density and material recognition in order to calculate physical properties like masses. Fig. 2 illustrates the fundamental pipeline of AI-based density recognition.

Building on identifying specific features from 2D images of previous work, we combine object detection with the assignment and calculation of physical properties. The image data is first analyzed by using a neural network. Salient objects can be identified and classified texturally. Object areas are identified as $b_1, ..., b_\mathrm{n}$. Within the bounding boxes, possible materials can be

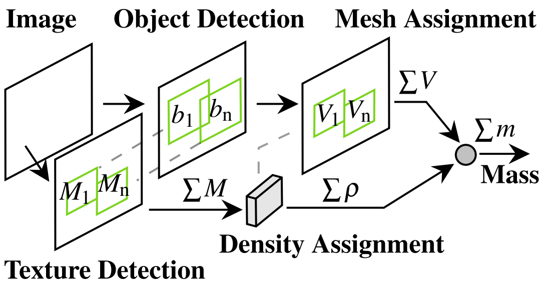


Figure 2: **Pipeline of AI-based density recognition.** The pipeline includes the detection of objects and their textures as well as the assignment of density and meshes to calculate physical quantities like object masses.

attributed based on Flickr Material Database (FMD) and Materials in Context Database (MINC). In this process, these materials M will be assigned a specific literary density ρ where we consider further pattern properties like image colors, shapes, textures and pixels. Since it is necessary to determine the volume V of the respective objects, the process calculates their corresponding meshes. The physical properties like mass M can be derived from the extensive information.

3.1 Object Detection

In order to detect objects, we use the convolutional network of YOLOv4 [12]. We extract features from the input images and calculate them into feature maps. As part of the YOLO architecture, we use backbone, neck, and head detectors, as shown in Fig. 3.

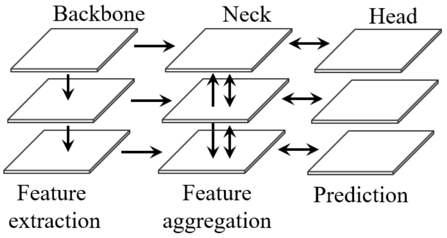


Figure 3: **YOLO Detectors used in this Work.** The backbone extracts important features from the image at different scales. The neck concatenates the semantic information from different layers of the backbone network and transmits it as input to the head. The head applies the refined features for predictive object recognition.

YOLOv4 contains a pre-trained convolutional neural network such as VGG16 or CSPDarkNet53 as a backbone which is based on SPP-Modul (Spatial Pyramid Pooling) and PAN (Path Aggregation Network). As part of the prediction, the head processes aggregated features and predicts the bounding boxes, objecthood, and classification values.

Our model is trained on a MS COCO dataset, which contains over 80 different classes and 1.5 million object instances in 200 thousand labeled images.

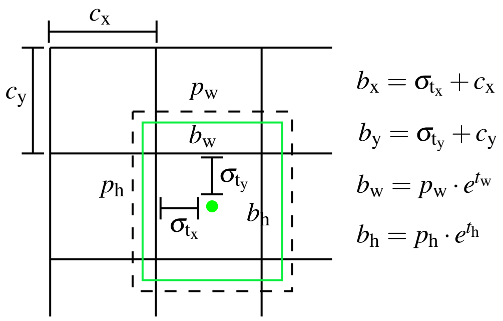

c_x
c_y
p_w
b_w
p_h
σ_{t_y}
σ_{t_x}
b_h
$b_x = \sigma_{t_x} + c_x$
$b_y = \sigma_{t_y} + c_y$
$b_w = p_w \cdot e^{t_w}$
$b_h = p_h \cdot e^{t_h}$

Figure 4: **Bounding Boxes with Dimension Priors and Location Prediction, adapted from [20].** The center coordinates of the box can be calculated with the predicted values t_x, t_y using a sigmoid function and offset by the location of grid cell c_x, c_y. The width and height of the final box are adjusted to the previous width p_w and height p_h and scaled by e^{t_w} and e^{t_h}.

Recognition first divides the image into a grid of cells like shwon in the example in Fig. 4. The number of cells depends on the size of the image. For example, With a size of 608 × 608 pixels, the cell size is usually 32 × 32 pixels. Our data set is divided into 19 × 19 cells.

Each object is assigned to exactly one cell, containing the object's center point. Objectless cells are filtered out according to their low probability of all 80 classes. The use of Non-Max-Suppression, as shown in Fig. 5, eliminates unwanted bounding boxes so that only the most probable bounding box remains for each detected object.

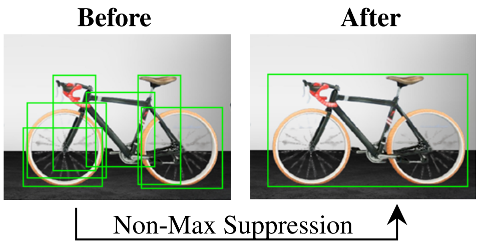


Figure 5: **Effect of Non-Max Suppression (NMS).** The post-processing technique Non-maximum suppression reduces the number of overlapping bounding boxes.

The bounding boxes localize the position of objects in order to recognize possible textures and calculate the object-specific volumes. The area is adjusted to 224 × 224 pixels for the matching designation of the model input.

3.2 Texture Detection

The object-specific bounding box detection enables the continuous application of a material detection model to the image area of the box. Inside the box, features such as color, SIFT, jet, micro-SIFT, micro-jet, curvature, edge-slice and edge-ribbon are combined and quantized into visual words by Bayesian framework [14].

Three different models are used, consisting of the MINC dataset. MINC comprises 23 different classes, each with 2500 images. We measure the confidence score for all three models across MINC dataset and Flickr Material Database (FMD) for appropriate model selection (see also Tab. 2).

Classes	VGG16	GoogleNet	AlexNet
Fabric	(78 | 78)	(69 | 78)	(45 | 64)
Foliage	(71 | 95)	(68 | 95)	(62 | 93)
Glass	(40 | 82)	(40 | 84)	(27 | 78)
Leather	(29 | 88)	(18 | 84)	(9 | 80)
Metal	(44 | 72)	(37 | 76)	(33 | 69)
Paper	(41 | 90)	(35 | 90)	(11 | 85)
Plastic	(78 | 75)	(84 | 78)	(74 | 68)
Stone	(78 | 89)	(62 | 87)	(52 | 85)
Water	(47 | 96)	(43 | 94)	(30 | 93)
Wood	(36 | 74)	(25 | 78)	(13 | 71)

Table 2: **Recognition on (FMD | MINC)**. VGG16 slightly outperforms GoogleNet. AlexNet provides the lowest accuracy out of these three contrasted models.

FMD consists of ten classes with 100 images of each. Tab. 2 shows the performance of the models for the FMD dataset. VGG16 provides optimum results with an overall accuracy of 86 % at MINC and 52 % at FMD. The model's size, however, requires significant time for both training and recognition. Alternatively, GoogleNet offers a good replacement model with an accuracy of 86 % for MINC and 47 % for FMD. The use of so-called inception modules allows a shorter calculation time. This module replaces a sparse CNN with a normal dense construction since most activations in a deep network are zero values or redundant due to correlations. As a result, not all output channels are connected to the input channels, hence the reduced computing time [27].

3.3 Mesh and Density Assignment

Through the accompanying object recognition, the next step is to assign a suitable 3D mesh. The mesh must have certain properties for the correct calculation of volume. For example, each triangle of the mesh must have corner points stored in a clockwise direction. The mesh must be complete without open areas and completely closed to prevent subsequent miscalculations. Eq. 3 describes the surface calculation of each signed triangle with $V_\mathrm{i} \in V$ and $i \in [1,n]$:

$$V_\mathrm{i}' = \frac{1}{6}(-x_{\mathrm{i},3}y_{\mathrm{i},2}z_{\mathrm{i},1} + x_{\mathrm{i},2}y_{\mathrm{i},3}z_{\mathrm{i},1} + \\ x_{\mathrm{i},2}y_{\mathrm{i},3}z_{\mathrm{i},1} + x_{\mathrm{i},3}y_{\mathrm{i},1}z_{\mathrm{i},2} - x_{\mathrm{i},1}y_{\mathrm{i},3}z_{\mathrm{i},2} - \\ -x_{\mathrm{i},2}y_{\mathrm{i},1}z_{\mathrm{i},3} + x_{\mathrm{i},1}y_{\mathrm{i},2}z_{\mathrm{i},3}) \tag{3}$$

We use i as the index for the triangles. $x_{\mathrm{i},[1,2,3]}$, $y_{\mathrm{i},[1,2,3]}$ and $z_{\mathrm{i},[1,2,3]}$ are the coordinates of the vertices of triangle i. Various shapes of objects are not considered in our analysis. Instead, each class is assigned to a specific 3D mesh. Fig. 6 exemplifies the resulting 3D mesh calculated by Eq. 3.

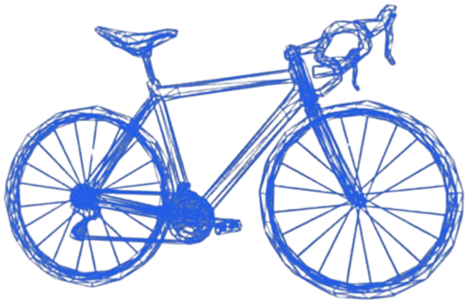


Figure 6: **Object Detection and Triangle Estimation.** Triangles: 5.022, Vertices: 4.159, UV Channels: 4 with approx. Size: 50 × 198 × 114.

After recognition, the applicable model is assigned. The object's mass is attributed based on the result of texture recognition. We simplify our analysis by assuming solid material for the respective model. A further database is created for the classification, which specifies the density of each recognizable material.

3.4 Physical Properties

Physical quantities such as forces, friction, pressures, temperatures, air resistances, inertia moments, energies or material properties n depend on density ρ. The fundamental calculation of density $(\rho = m/V)$ and the inclusion of further coefficients and constants offers possibilities of inferring different values.

By recognizing the actual object and the possible material assignment, we can deduce the volume and mass of the object. The physical property is determined by iterating over each section since many objects are divided into sub-objects. The bicycle in Fig. 6, for example, includes the individual wheels, the handlebars, the saddle, and the frame. For each section, the signed volume is now calculated for each triangle and added to the total volume. Surfaces that point outwards contribute to the total volume. Surfaces that point inwards subtract from it. This leaves only the volume on the inside. The density and volume can then be multiplied to calculate the weight of the object.

4 EXPERIMENTAL SETUP

Our approach utilizes Unreal Engine 4.27 and its high-fidelity rendering pipeline. The realistic rendering and lighting allow us to assume real test simulations as shown in Tab. 3. The neuronal training is based on realistic test images.

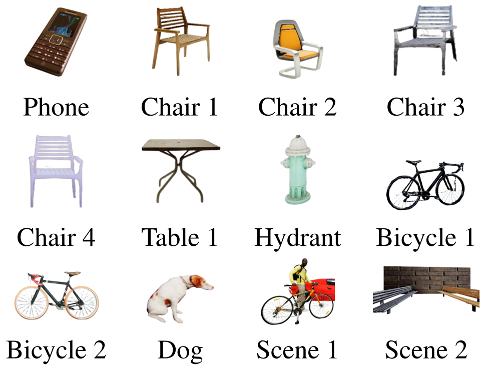


Table 3: **Sample of Items used in our Experiment.** We used Unreal Engine 4.27 for its realistic rendering capabilities.

The accuracy of the physical information is directly linked to the image quality. In our conceptual consideration, we use images with a size of 608 × 608 pixels and three assigned color channels.

Our computing hardware has an integrated Quad-Core Intel or AMD, 2.5 GHz, 8GB RAM, external GPU1 Nvidia GeForce GTX 1050 Ti, and onboard GPU0 of Intel HD Graphics 630. Cv2 is used for image processing and reading deep neural networks, and NumPy for mathematical functions. Mean subtraction calculates the average pixel intensity over all images of the used training set of all three color channels and subtracts these values from the channels of the input image. When using YOLOv4, channel swapping is also applied for the mean subtraction. Here the image is swapped in RGB order. In order to obtain one optimal bounding box for each object, non-max suppression is applied. To predict the material, we iterate over each object and use the corresponding image area as input to the texture model, similar to the object detection model.

5 EVALUATION

Our evaluation employs density recognition of several diverse objects.

We used a Convolutional Neural Network to recognize the texture of the object and MINC for texture recognition. The model trained and used in our approach shows the highest accuracy when compared to other CNN architectures.

In order to evaluate the functionality of our approach, we explored and tested select scenarios (see Tab. 3). We summarize our results in Tab. 4.

However, the FMD dataset only achieves an accuracy of 52 % across all classes. One of the reasons is the insufficient data set, which consists of only ten categories in the neural network and 23 in the MINC data set. Increasing the data density would significantly increase the hit rate. With more extensive training, our approach can also be transferred to other environments. This requires the RGB image for analysis and the database with the necessary 3D networks and recognition models for evaluation.

	Material Type	**Density** $[kg/dm^3]$ **(Literary | Measured)**
Phone	Plastic	(4.0 | 1.2) ± 70 %
Chair 1	Wood	(0.7 | 0.7) ± 4 %
Chair 2	Plastic	(4.8 | 1.2) ± 75 %
Chair 3	Metal	(7.9 | 8.0) ± 1 %
Chair 4	Metal	(0.9 | 1.2) ± 30 %
Table 1	Metal	(7.9 | 8.0) ± 1 %
Hydrant	Metal	(7.9 | 8.2) ± 14 %
Bicycle 1	Metal	(2.9 | 8.0) ± 180 %
Bicycle 2	Metal	(7.9 | 8.0) ± 2 %
Bench 1	Metal	(7.9 | 8.0) ± 1 %
Bench 2	Wood	(2.1 | 0.7) ± 66 %
Dog	Other	(1.1 | 1.0) ± 5.6 %
Person	Plastic	(1.1 | 1.0) ± 9 %
Backback	Fabric	(1.4 | 1.6) ± 16 %
Car	Plastic	(5.4 | 1.2) ± 77 %

Table 4: **Detected Materials and Density as well as Percentage Error [%] from the actual Physical Values of the Objects.** The measurements for Phone, Chair 2, Bicycle 1, Bench 2, and Car deviated significantly from the actual values. These deviations stemmed from incorrect detection of the material.

The actual size of objects within a scene has not been considered in previous work. This would be useful for volume calculation and different scaling of objects. Even within the categories, no distinctions are made between different types of objects. Each object is only assigned a 3D mesh, which is considered the average for that class. This means that object shapes are not taken into account. Furthermore, for successful mapping and analysis of physical properties, the use of error-free, detailed, and complete 3D models is essential. However, depending on the orientation and movement of the objects, the calculated volume may be inaccurate. Serial images could help alleviate this error.

6 LIMITATIONS

While our implementation shows promising initial results, it solely serves to illustrate the feasibility of our proposed concept. Although our implementation used neural networks that were trained with real-world images, we relied on synthetic datasets to assess the performance of the implementation. Consequently, the generalizability and applicability of our findings may be limited.

Additionally, our evaluation of the detected physical properties examined the average density of each object.

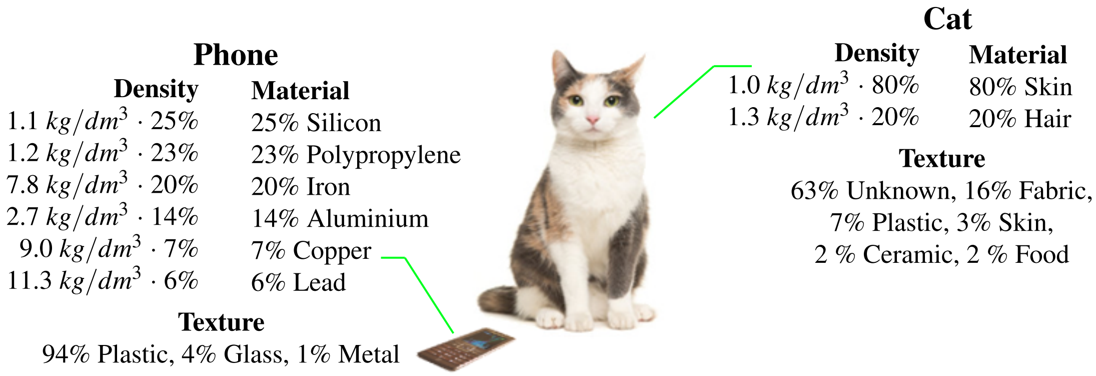


Figure 7: **Material and Density Composition of Recognized Objects.** The inner ingredients are frequently more complex and diverse than the exterior texture suggests. The illustration describes the composition of two example objects: A smartphone and cat. The density of occluded components can be estimated from the average composition of each object.

Therefore, our work might not guarantee that our approach is readily applicable to other properties, such as plasticity or thermal conductivity, as well as to more complex object compositions.

7 CONCLUSION AND FUTURE WORK

In this paper, we presented a concept for the object-based recognition and assignment of physical properties as density or material based on a 2D image. Our work is motivated by the challenges of distinguishing objects and their properties from each other. The distinction of mass or density enables new interaction possibilities in which the causal relationships of an environment can be linked to the properties of a given object. Our method recognizes specific patterns from 2D images by neural networks in which we estimate the volume by the number of object-recognized triangles. The density is ultimately calculated from the object-specific assignment of a material recognition model and the associated volume.

Despite the promising results of our approach, further work and improvements are needed. An essential aspect relates to the data sets that are used. The accuracy of object-based density recognition goes hand in hand with the quality of the trained AI model. Our current field of application is limited to synthetic test data. Future iterations and evaluations using real-world data sets could help deliver further insights into AI-based density recognition. To achieve serviceable recognition and acceptable results, the appropriate data set needs to be selected for the application area. Therefore, it is necessary to extend the data sets and the training model. The COCO dataset covers numerous categories but neglects existing subcategories. By selecting such data sets, the transferability of the model could be increased.

Our evaluation shows only a limited number of materials assigned to the objects. In reality, the number and composition of materials may be different and more diverse (see Fig. 7). In particular, the interior composition of a recognized object may differ from the recognized surface texture. Consequently, drawing on and combining a wide variety of databases could lead to more precise and serviceable results. These assignments can be linked using an ontological approach. Utilizing ontologies, information and their relationship to each other can be stored in a machine-readable form or made comprehensible. Additional graph databases can visualize the data nodes and their relationship and make them interpretable. In principle, different ontologies can be merged within one ontology. In this context, possible databases on physical properties such as material, geometry and objects could be linked ontologically and applied to the principle of AI-based density recognition.

Supporting these classifications with suitable image segmentation, such as with self-organized maps [17], could further increase the number of distinguishable materials. The partial change of the segmentable areas could be cut out or reduced to densely recognizable areas, which would also reduce quality restrictions and latencies.

Estimating the object size within a scene proves to be a difficult task. This process assumes the same size for all everyday objects in a scene. In the future, it will be necessary to measure the object size and distance of acquired 3D models for meter-level distinctions. The model transfer to a spatial data set would be suitable for this purpose.

The use of image series or video material can also be helpful to support a spatial data set [7]. In this context, the distance and perspective of objects within a scene can be used to determine the speed and possible acceleration of an object. Javadi et al. [11] describe a video-based vehicle speed system for measuring speed based on a measured route. By determining the speed and acceleration of an object, statements can be made about the forces released in a collision. Video analysis can also be useful for other areas of physical property recognition. The depiction of an object in several individual images with different perspectives allows properties derived previously to be checked and re-evaluated. This includes, for example, the volume or size of the object.

8 ACKNOWLEDGEMENTS

We thank Philip Raschdorf for his support during concept development and data collection. We also thank Thomas Odaker and Elisabeth Mayer, who supported this work with helpful discussions and feedback.

9 REFERENCES

[1] Flickr material database (fmd). MIT (2024-05-06), `https://people.csail.mit.edu/celiu/CVPR2010/FMD/index.html`

[2] Bell, S., Upchurch, P., Snavely, N., Bala, K.: Material recognition in the wild with the materials in context database. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3479–3487 (2015). https://doi.org/10.1109/CVPR.2015.7298970

[3] Cai, Z., Fan, Q., Feris, R., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: Computer Vision - ECCV. pp. 354–370 (2016). https://doi.org/10.1007/978-3-319-46493-0_22

[4] Daglioglu, M.A.: Object-detection mit you only look once (yolo) : Einführung in die objekterkennung mit yolo sowie die weiterentwicklung in den versionen v2-v4 (2021)

[5] Erdem, K.: Understanding region of interest (roi pooling) (2020), `https://erdem.pl/2020/02/understanding-region-of-interest-ro-i-pooling`, (visited on 2024-02-20)

[6] Ester, M., Sander, J.: Knowledge discovery in databases. Springer (2000). https://doi.org/10.1007/978-3-642-58331-5

[7] Evain, A., Khemmar, R., Orzalesi, M., Ahmedali, S.: Impact of calibration matrices on 3d monocular object detection: Filtering, dataset combination and integration of synthetic data. International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG) (2024)

[8] Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)

[9] Ippolito, P.P.: Feature extraction techniques (2019), `https://towardsdatascience.com/feature-extraction-techniques-d619b56e31be`, (visited on 2024-02-20)

[10] Janiesch, C., Zschech, P., Heinrich, K.: Machine learning and deep learning. Electronic Markets **31**(3), 685–695 (2021). https://doi.org/10.1007/s12525-021-00475-2

[11] Javadi, S., Dahl, M., Pettersson, M.I.: Vehicle speed measurement model for video-based systems. Computers & electrical engineering **76**, 238–248 (2019). https://doi.org/10.1016/j.compeleceng.2019.04.001

[12] Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics yolov8 (2023), `https://github.com/ultralytics/ultralytics`, (visited on 2024-02-20)

[13] Krig, S.: Computer vision metrics: Survey, taxonomy and analysis of computer vision, visual neuroscience, and deep learning. Springer (2016)

[14] Liu, C., Sharan, L., Adelson, E.H., Rosenholtz, R.: Exploring features in a bayesian framework for material recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010). https://doi.org/10.1109/CVPR.2010.5540207

[15] Mehlig, B.: Machine learning with neural networks: an introduction for scientists and engineers. Cambridge University Press (2021)

[16] Müller, J., Fregin, A., Dietmayer, K.: Disparity sliding window: Object proposals from disparity images. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5777–5784 (2018). https://doi.org/10.1109/IROS.2018.8593390

[17] Müller, S., Kranzlmüller, D.: Self-organising maps for efficient data reduction and visual optimisation of stereoscopic based disparity maps. In: International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (2022). https://doi.org/10.24132/CSRN.2021.3101.3

[18] Nicodemus, F.E.: Directional reflectance and emissivity of an opaque surface. In: Applied Optics (1965). https://doi.org//10.1364/AO.4.000767

[19] Nielsen, M.A.: Neural networks and deep learning. Determination press, San Francisco, CA, USA (2015)

[20] Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017). https://doi.org/10.1109/CVPR.2017.690

[21] Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. Tech. rep. (2018). https://doi.org/10.48550/arXiv.1804.02767

[22] Ren, S., He, K., Girshick, R., Jian, S.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: IEEE transactions on pattern analysis and machine intelligence. pp. 1137–1149 (2017). https://doi.org/10.1109/TPAMI.2016.2577031

[23] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. nature **323**(6088), 533–536 (1986)

[24] Shalev-Shwartz, S., Ben-David, S.: Understanding machine learning: From theory to algorithms. Cambridge university press (2014)

[25] Shukla, A., Kalnoor, G., Kumar, A., Yuvaraj, N., Manikandan, R., Ramkumar, M.: Improved recognition rate of different material category using convolutional neural networks. Materials Today: Proceedings **81**, 947–950 (2023)

[26] Sun, Y., Gu, Z.: Using computer vision to recognize construction material: A trustworthy dataset perspective. In: Elsevier (2022). https://doi.org/10.1016/j.resconrec.2022.106362

[27] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015). https://doi.org/10.1109/CVPR.2015.7298594

[28] Wu, J., Lim, J.J., Zhang, H., Tenenbaum, J.B., Freeman, W.T.: Physics 101: Learning physical object properties from unlabeled videos. In: Proceedings of the British Machine Vision Conference (2016)

[29] Yadav, N., Yadav, A., Kumar, M.: An introduction to neural network methods for differential equations. Springer (2015). https://doi.org/10.1007/978-94-017-9816-7

Cocoa beans moisture content prediction using Machine Learning Model based on the color image features

Joel E AKO[1,2]
Joel.Ako-Ekissi@insa-rennes.fr

Camille E. N'ZI[1]
camille.nzi@inphb.ci

Kidiyo KPALMA[2]
Kidiyo.Kpalma@insa-rennes.fr

(1)Institut National Polytechnique Félix Houphouët Boigny (INP-HB), Unité Mixte de Recherche et d'innovation (UMRI) en Science des Technologies d'Ingenieur (STI), BP 1093 Yamoussoukro, Côte d'Ivoire ;
(2)Univ Rennes, CNRS, Institut National des Sciences Appliquées (INSA), IETR (Institut d'Electronique et des Technologies du numéRique) - UMR 6164, F-35000 Rennes, France.

ABSTRACT

The moisture content of cocoa beans is an essential factor in their quality. Modeling it during drying is still problematic due to the wide variation in drying conditions and the wide variation in cocoa bean varieties. This article aims to investigate the possibility of modeling the moisture content of cocoa beans as a function of RGB images features of unshelled cocoa beans. The approach is to extract features, analyze them and then use the most relevant ones to study Machine Learning models. Features are extracted by calculating mean, standard deviation, energy, entropy, kurtosis and skewness of the components of the rgb (RGB normalized), HSV, L*a*b*, YCbCr color spaces without the brightness components. These features are extracted from 4 types of samples, namely 10, 30, 50 and 70 bean samples per image. Features analysis using the F-test and RReliefF methods shows that the features based on the energy and entropy of the components rg, yb, Cr, Cb, a*, b* and h* are fairly relevant for predicting the water content of cocoa beans. However, they are highly correlated. The selected predictors allow the analysis of linear models, such as Ridge Regression (RR), PLS Regression (PLSR) and non-linear models, such as polynomial, Support Vector Regression (SVR) with rbf kernel, and Decision Trees Regression (DTR). Except RR and PLSR, the other models were preceded by a principal component analysis (PCA) to handle the collinearity problem. The non-linear models give good predictions for the training dataset, with coefficients of determination R^2 ranging from 0.94 to 0.96 and RMSE from 3.85 to 4.81. However, there is a significant difference between these results and the predictions of the new datasets. RR and PLSR are stable models, but their predictions are less than non-linear ones. It is therefore possible to predict the moisture content of cocoa beans from the features of RGB images.

Keywords

cocoa beans, Moisture content, color features, F-test, RReliefF, Regression, Machine Learning

1 INTRODUCTION

The moisture content of a product is the amount of water present in this product. It is important for the microbiological and nutritional properties of food products, as well as for regulatory and economic aspects. As a result, determining moisture content is one of the most frequent analyses carried out in the food industry. If products are to be stored for long periods, they need to be dried to a certain water content. In the case of cocoa, after harvesting the ripe cocoa pods, fresh cacao beans are fermented and dried immediately after fermentation to safe moisture content from around 60% to 7-8% (ISO 2451/2014 standard) [DJE09] to facilitate storage, transport and guarantee the quality of the beans. Too high a moisture content can encourage the development of mold and alter the quality of the final product, while too low a moisture content can make the beans brittle and difficult to process. It can cause damage that contributes to the depreciation of bean quality [HUM10]. Predicting moisture content during drying is therefore an optimum solution for ensuring quality drying. The moisture content prediction during drying requires a non-destructive solution. However, the implementation of such a solution is very complex due to the instability of drying conditions and the diversity of cocoa beans. Despite this, researchers propose a few solutions. These include modeling drying kinetics [DJE09], [HII11], [IGO15], [KAR18], [CAS23], predicting moisture content by Near-Infrared spectral

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.

https://www.doi.org/10.24132/CSRN.3401.25

[HAS18] and designing artificial dryers [KAV21]. Despite this research, modeling bean moisture content during drying is still problematic, because the models developed depend on drying conditions. This work involves exploring image features to design a model that predict the moisture content of cocoa beans during the drying process. The aim is to identify image features that are not affected by drying conditions and to design a model for this prediction. Our approach consists of two main steps. The first step involves selecting the discriminant predictors and the optimal quantity of beans by using two variable selection methods: the F-test and the Regression Relief Features selection (RReliefF) algorithm. The second step is devoted to studying linear and non-linear Machine Learning models, such as Ridge Regression (RR), PLS Regression (PLSR), polynomial, Support Vector Regression (SVR) with RBF kernel and Decision Trees Regression (DTR), the selected relevant features.

The remainder of this work is organized as follows: Section 2 presents the related solutions. Section 3 and 4 describe the proposed approach and experiments and the datasets, respectively. Section 5 presents the results and discussion and section 6 the main conclusions.

2 RELATED WORKS

The related works concern predicting or modeling the moisture content during drying. The models already developed are based on drying kinetics and Near-Infrared spectral.

Drying kinetics-based modeling

Drying kinetic modeling solutions are based on mathematical and artificial neural network models. Hii C. L. et al. have used Fick's theoretical model to study the drying kinetics of cocoa beans. They obtain coefficient of determination R^2 of training data ranging from 0.9845 to 0.9976. But they mention that the drying process is highly unsteady due to the fluctuating ambient conditions [HII09]. A. Djedjro et al. evaluate a suitable drying mathematical model for describing the drying curves. Among the mathematical models studied, the logarithmic model satisfactorily described the drying behavior of cocoa beans with a coefficient of determination 0.976 and RMSE 0.0128 [DJE09]. Nogbou A. et al. described the behavior of cocoa beans in predicting their moisture content during intermittent microwave drying at different power levels (450 W, 600 W, 700 W). They proposed a recurrent artificial neural network model using drying time, microwave power and moisture content as inputs. They obtained a coefficient of determination ranging between 0.9967 and 0.9993. [IGO15]. Daouda K. et al. proposed a mathematical model of the evolution of cocoa beans moisture content as a function of time using an artificial neural network during the sun drying. They found the multilayer perceptron with two neurons on the input layer to be the most suitable. The coefficient of determination of the linear regression between observed and predicted water content values was 0.99 [KAR18]. Eduardo Castillo et al. fitted a diffusion approximation model using nonlinear regression to the moisture ratio of the CCN51 cocoa bean with the drying time for the constant drying temperatures of $40, 50, 60$, and $70°C$. The coefficient of determination for all cases was 0.9999 with RMSE 0.0044 [CAS23].

Near-infrared spectrum-based modeling

Hashimoto et al built PLS regression models from near-infrared diffuse reflectance spectrum for the prediction of several cocoa bean quality parameters including water content. The coefficient of determination of moisture content prediction is 0.67. [HAS18].

The majority of papers found focuses on Drying kinetics. The advantage of these solutions is that they give a good prediction of moisture content for the training data. The disadvantage is that they depend on drying conditions, i.e. temperature and drying time. However, under natural drying conditions, time and temperature are highly unsteady due to the fluctuating ambient conditions [HII09]. The other solution, which is independent of theses conditions, uses near-infrared spectra, which has a low prediction rate. The proposed approach uses image features to propose model independent of drying conditions.

3 PROPOSED SOLUTION

The proposed solution is based on prediction of cocoa beans moisture content using color features. It involves acquiring images of batches of cocoa beans during drying at regular time intervals to designing the best Machine Learning model for predicting moisture content. The differents stages of the proposed solution are shown in the block diagram in figure 1.

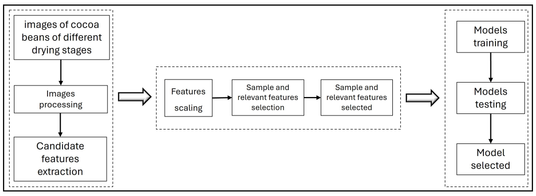

images of cocoa beans of different drying stages
Images processing
Candidate features extraction
Features scaling
Sample and relevant features selection
Sample and relevant features selected
Models training
Models testing
Model selected

Figure 1: Proposed solution steps

3.1 Image processing

The image processing involves extracting cacao beans from the blue acquisition background. Then, the RGB images are segmented using color thresholding in L*a*b* space. A region of interest (ROI) are created firstly with Matlab Color Thresholder app [MATSG24], then apply morphological opening and closing with the optimal structuring element(disk) to perfect the edges of the extracted beans (figure 2).

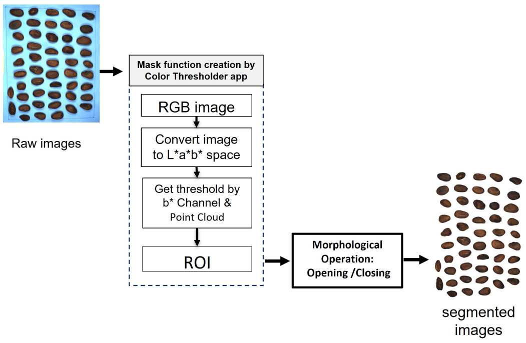

Raw images
Mask function creation by Color Thresholder app
RGB image
Convert image to L*a*b* space
Get threshold by b* Channel & Point Cloud
ROI
Morphological Operation: Opening /Closing
segmented images

Figure 2: Image segmentation

3.2 Extraction of color-based features

Features are extracted from RGB images, using statistical Moments such as mean, standard deviation, energy, entropy, kurtosis and skewness. These methods were applied to the color components of the rgb [RAS06], $HSV, YCrCb$ and $L^*a^*b^*$ spaces, and the chromatic components in spherical coordinates θ and ϕ [RAS06], without the luminance components. We also have the components rg (red-green) and yb (yellow-blue) derived from rgb space [WAN14] and C^* and h^* derived from $L^*a^*b^*$ space. $HSV, YCbCr$ and $L^*a^*b^*$ space components are derived from RGB images, using the corresponding MATLAB functions. Expressions for the other components (from equation (1) to equation (9) and the statistical Moments (from equation (10) to equation (15)) are following, where R, G, B are the components of RGB space, N is the number of bean pixels in the image, A_i is the gray level of pixel i and h_A is the normalised histogram of gray level A of the image I.

$$\mathrm{r} = \frac{R}{R+G+B} \tag{1}$$

$$\mathrm{g} = \frac{G}{R+G+B} \tag{2}$$

$$\mathrm{b} = \frac{B}{R+G+B} \tag{3}$$

$$\mathrm{rg} = r - g \tag{4}$$

$$\mathrm{yb} = \frac{r}{2} + \frac{g}{2} - b \tag{5}$$

$$C^* = \sqrt{a^{*2} + b^{*2}} \tag{6}$$

$$h^* = \arctan\left(\frac{a^*}{b^*}\right) \tag{7}$$

$$\theta = \arctan\left(\frac{G}{R}\right) \tag{8}$$

$$\phi = \arcsin\left(\frac{\sqrt{R^2+G^2}}{\sqrt{R^2+G^2+B^2}}\right) \tag{9}$$

$$\text{Mean} = \frac{1}{N}\sum_{i=1}^{N} A_i \tag{10}$$

$$\text{SD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(A_i - Mean)^2} \tag{11}$$

$$\text{Energy} = \sum_{A=0}^{255} h_A^2 \tag{12}$$

$$\text{Entropy} = -\sum_{A=0}^{255} h_A ln(h_A) \tag{13}$$

$$\text{Kurtosis} = \frac{1}{SD^4}\sum_{A=0}^{255}(A - Mean)^4 h_A \tag{14}$$

$$\text{Skewness} = \frac{1}{SD^3}\sum_{A=0}^{255}(A - Mean)^3 h_A \tag{15}$$

3.3 Variable selection methods

Variable selection is a step that precedes model design. It allows to analyze the potential explanatory and selection variables most relevant to a model design. It also provides the necessary information on each of the explanatory variables for better use and interpretation in a model. To select discriminating predictors, we use two different methods. The F-test method to assess the significance of candidate variables and Regression Relief Features selection (RReliefF) algorithm to assess their relevance.

F-test

The F-test is a statistical test that compares the variances of two samples, or the ratio of variances between several samples. It is often used to test equality of means in an analysis of variance, or to test the goodness of fit of a regression model. The F-test is based on the F-statistic. It is the ratio of the variance explained by the model to the residual variance. The observed p_{value} are used to interpret the F-test. Higher the F-statistic, the smaller the p_{value}, thus better the model fits data. We use it to assess the goodness of fit of each candidate variable in a linear regression model with moisture content. This involves examining the importance of each candidate color feature, then ranking them using the p_{value} of the F-test statistics. The score for each candidate variable is determined by the following relationship [MATFT24], [OME14]:

$$\text{S} = -\log(p_{value}) \tag{16}$$

Regression Relief Features selection (RReliefF) algorithm

The RReliefF algorithm is an extension of the Relief algorithm, which is a variable selection method based on assigning weights to variables. It detects relevant variables by considering interactions between variables and noise in the data. It also penalizes the predictors that give different values to neighbors with the same response values, and rewards predictors that give different values to neighbors with different response values. However, it uses intermediate weights to compute the final predictor weights. Then, it calculates the predictor weights W_j after fully updating all the intermediate weights [ROB97].

$$\mathrm{Wj} = \frac{W_{(dy \wedge dj)}}{W_{dy}} - \frac{W_{dj} - W_{(dy \wedge dj)}}{m - W_{dy}} \tag{17}$$

W_{dy} and W_{dj} are the weights of having different values for the response y and predictor x_j, respectively. $W_{(dy \wedge dj)}$ is the weight of having different response y and different values for the response y values and different values for the predictor x_j. m is the number of iterations.

The selected variables will be used to analyse the prediction of water content using Machine Learning models.

3.4 Machine Learning regression models

Different types of regression models are explored, namely: Ordinary Least Squares, Kernel Support Vector, Decision Trees, Ridge and Partial Least Squares models.

Ordinary Least Squares regression (OLSR)

We analyse the Multiple linear Regression (MLR) and Polynomial Regression (PR) Models. The sample regression model has the form [MONT21]:

$$y_i = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i \tag{18}$$

Where x_{ij} and y_i are the k predictors and the response respectively of ith observation. The parameters β_j, $j = 0, 1, ..., k$ are the regression coefficients. OLS is the most popular estimation method; its purpose is to find the unbiased coefficients $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^\top$ which minimize the residual sum of squares:

$$RSS(\beta) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 \tag{19}$$

Support Vector Regression (SVR) with the radial basis function (RBF) kernel

SVR is based on the Support Vector Machine algorithm. It also based on the computation of a linear regression function in a multiple variables feature space where the input data can be used via a non-linear regression function. Unlike OLSR that aim to minimize the error between the predicted and actual values, SVR aims to fit as many instances as possible within a margin while limiting violations of the margin and controlling the margin width, in other words find a hyperplane that best fits as many data points as possible while minimizing the margin violations. The margin is defined as the region between the hyperplane and the support vectors. A nonlinear function has the form [CAS20]:

$$f(x_i) = \omega^T \Phi(x_i) + b \tag{20}$$

Given training vectors $x_i \in R^p$, $i = 1, .., N$, and a vector $y \in R^N$ SVR solves the following primal problem:

$$min_{(\omega,b,\zeta,\zeta^*)} \left(\frac{1}{2} \omega^T \omega + C \sum_{i=1}^{N} (\zeta_i + \zeta_i^*) \right) \tag{21}$$

subject to $y_i - \omega^T \Phi(x_i) - b \leq \varepsilon + \zeta_i$,
$\omega^T \Phi(x_i) + b - y_i \leq \varepsilon + \zeta_i^*$; $\zeta_i, \zeta_i^* \geq 0, i = 1, .., N$

Where ω is the weight vector, b is the bias, and $\Phi(x_i)$ is the high dimensional feature space. $C < 0$ is a prespecified constant that is responsible for regularization and represents the weight of the loss function. The first term of the objective function $\omega^T \omega$ is the regularized term and the second term $C \sum_{i=1}^{N} (\zeta_i + \zeta_i^*)$ is called the empirical term and measures the $\varepsilon-$insensitive loss function. ζ_i and ζ_i^* are the slack variables to guard against outliers, they represent the distance between the potential support vector and the potential outliers.

Kernel methods achieve flexibility by fitting simple models in a local region to the target point x. Localization is achieved via a weighting kernel K, and individual observations receive weights $K(x_i, x)$, so the Radial basis function is written as:

$$f(x) = \sum_{i \in SV} (\alpha_i - \alpha_i^*) K(x_i, x) \tag{22}$$

With $K(x_i, x) = \exp(-\gamma \,||\, x_i - x \,||^2)$

$\alpha_i - \alpha_i^*$ are coefficients of the support vector in the decision function, γ is the kernel coefficient and x is the center of feature.

Decision Trees Regression (DTR)

DTR uses a tree-like structure to model the relationship between the set of predictors and the response. The tree is composed of nodes that represent the possible values of the predictors or the response. Its purpose is to find the best split at each node. The quality of a candidate split θ of node m is then computed using an impurity function or loss function $H(Q_m)$, the choice depends on the task being solved:

$$G(Q_m, \theta) = \frac{n_m^{left}}{n_m} H(Q_m^{left}(\theta)) + \frac{n_m^{right}}{n_m} H(Q_m^{right}(\theta)) \tag{23}$$

Select the parameters that minimizes the impurity:

$$\theta^* = argmin_\theta G(Q_m, \theta) \tag{24}$$

Q_m is the data at node m partitioned into Q_m^{left} and Q_m^{right} with n_m sample. $H(Q_m)$ is the Mean Squared Error (MSE) such as ($\bar{y}_m$ is the mean of predicted value):

$$H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y - \bar{y}_m)^2 \tag{25}$$

Ridge Regression (RR)

RR shrinks the Least Squares unbiased coefficients by imposing a penalty on their size. It produces biased estimators of regression coefficients, that have a small variance and more stable than the Least Squares unbiased coefficients, which called ridge estimators. The coefficients minimize a penalized residuals sum of square. For a given value of λ, a non-negative parameter, RR solves the problem [ROD22]:

$$min_{(\beta_0 \beta)} \left(\frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^\top \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right) \tag{26}$$

where N and p are the observation and predictor numbers, respectively.

Partial Least Squares Regression (PLSR)

PLS regression constructs a set of latent variables. These latent variables are linear combinations of the original predictors, created in such a way that they explain the maximum covariance between the predictors and the response variable. PLSR model with h latent variables can be expressed as follows [CHO05]:

$$X = TP^t + E \tag{27}$$

$$y = Tb + f \tag{28}$$

In Equation (27, 28) $X(n \times p), T(n \times h), P(p \times h), y(n \times 1)$ and $b(h \times 1)$ are respectively used for predictors, X scores, X loadings, response, and regression coefficients of T. The $k-th$ element of column vector b explains the relation between y and t_k, the $k-th$ column vector of T. Meanwhile, $E(n \times p)$ and $f(n \times 1)$ stand for random errors of X and y, respectively.

4 EXPERIMENTS AND DATA SETS

4.1 Experiments

4.1.1 Sample preparation

The experiments were conducted on well-fermented commercial cocoa beans from the same harvest south of Ivory Coast. We extracted $15kg$ from $25kg$ fermented cocoa beans on the last day of fermentation (day 7). Once in the laboratory, the beans are distributed in batches of 10, 30, 50 and 70 in polystyrene bags, then stored in a cold room at $-10°C$ throughout the handling process. Before using, the cocoa beans are defrosted at room temperature in the laboratory.

4.1.2 Experimental process

The experimental process is composed of three steps: drying, image acquisition, and weight determination, as shown in the experimental cycle (figure 3).

- *Cocoa drying*: the cocoa bean batches were dried in a domestic microwave oven (SHARP R-75 MT). Intermittence drying is used with a pulsing ratio (Equation 29) of 4 to limit local overheating and a drying power of $270W$. This means 2 minutes of microwave start-up and 4 minutes of shutdown. Drying was stopped when the moisture content reached around 7%. Data are acquired at regular time intervals during drying, this resulted in 30 to 45 observations per batch

- *Weight determination*: the weight of the beans batch is determined by a digital precision scale with 0.01g precision.

- *Image acquisition*: A setup for capturing customized images has been developed. It is composed of: *a color coupled charge device (CCD) camera* (SONY XCG-5005CR, Japan), which is specifically standardized for machine-vision applications based on Gigabit Ethernet technology, *a lens zoom 16 mm* (Fujifilm corporation, model HF16HA-1B, Japan). *The image acquisition card* (Mil Matrox) is used for transferring information from camera to computer (Core-i7 CPU: 2.5 GHz; RAM: 4 GB). And two 8.5 watts white LED, which cover the visible wavelength, to ensure correct and consistent lighting throughout the acquisition process. The set is placed in a closed box to control the lighting. The images have been acquired in tiff format, 2448 x 2048 definition, 96 ppi (horizontal) and 96 ppi (vertical) and unit 8.

figure 4 shows some cocoa beans images at different times with their moisture content.

$$PR = \frac{CyclePoweronTime + CyclePoweroffTime}{CyclePoweronTime} \tag{29}$$

4.1.3 Moisture content computation

At the end of the drying process, the dry mass of each batch is determined by drying the dried beans in an oven at $103°C$ for 16 hours to determine the dry weight. The weights determined during drying are used to calculate the water content using the following formula (ISO 2451/2014 standard):

$$MC_t \ (wet\ base) = \frac{m_t - m_s}{m_t} \tag{30}$$

With MC the moisture content, m_t the weight at time t and m_s the dry weight.

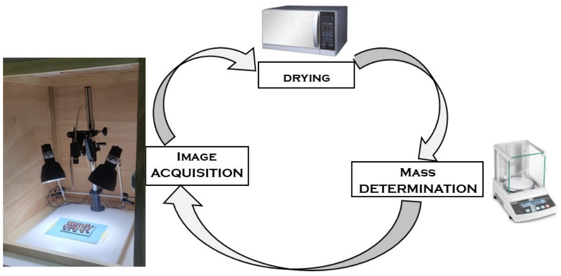


Figure 3: experimental cycle.

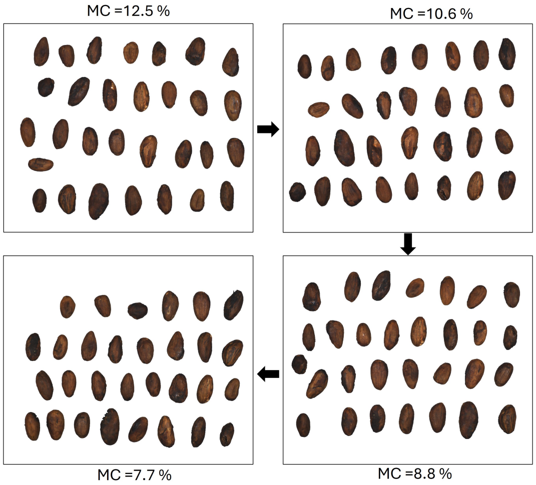


Figure 4: Cocoa bean images as a function of MC

4.2 Dataset description

The data come from 4 different samples, depending on the number of beans per batch or per image. E10, E30, E50 and E70 samples, with 10, 30, 50 and 70 beans per batch respectively. Each sample consists of 5 batches. For the variable selection, the dataset of a sample is made up of all the observations of the 5 batches of this sample. For the model analysis, the dataset consists of 80% of the dataset for training and 20% for test. The dataset consists of color features from segmented images, as predictors and moisture content as response. Application of extraction methods to the color components yielded 90 candidate variables. The database for each sample is then standardized using the z-score method. The Z-score standardization involves transforming each feature in the dataset such that it has a mean 0 with a unit standard deviation (Equation 31).

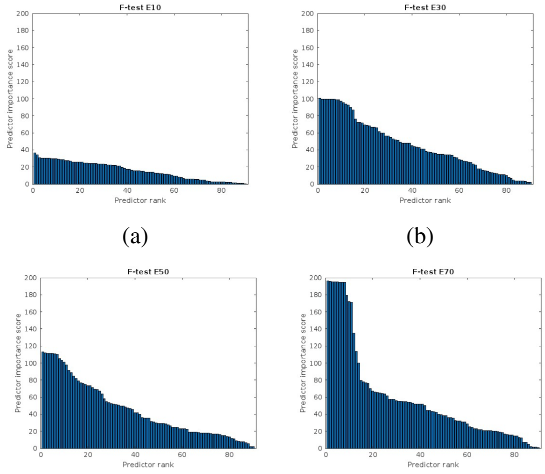
$$\mathrm{z} = \frac{(x-\mu)}{\sigma} \tag{31}$$

Where μ, σ and x are the mean, the standard deviation and the feature value of the original dataset.

5 RESULTS AND DISCUSSION

5.1 Features extraction and selection

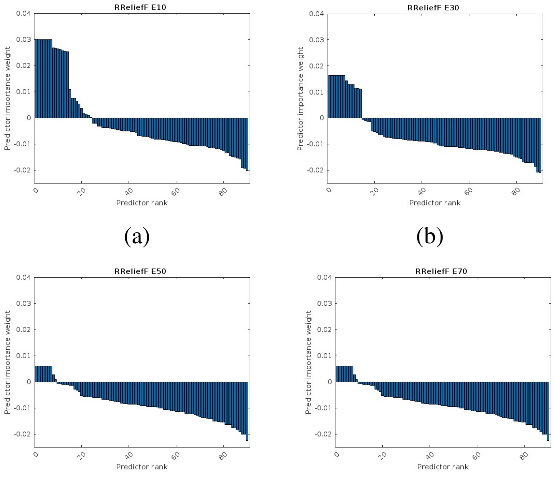
Figure 5a to figure 6d show the importance scores of each candidate variable for each sample, which (a), (b), (c) and (d) correspond to samples E10, E30, E50 and E70, respectively. The variables are ranked in importance order.



(a) (b)

(c) (d)

Figure 5: F-test predictor importance score



(a) (b)

(c) (d)

Figure 6: RReliefF predictor importance score

Figure 5a to Figure 5d display the significant score for the F-test method. The predictors importance score maximums are 36.36, 100.36, 113.10 and 196.30 from respectively samples E10, E30, E50 and E70. The significance of each variable increases as the number of beans per image increases, and the sample E70 gives the best scores. Sample E70 allows to distinguish the most significant predictors. The most significant features are those that stand out from the others in the E70 sample. Their scores range from 99.73 to 196.30. There are

14 of them, with scores ranging from 99.73 to 196.30. Even if samples $E30$ and $E50$ don't help to distinguish them, they are still ranked in the same order, except in sample $E10$.

Figure 6a to Figure 6d show the RReliefF predictor importance weight. The weights reflect how each feature discriminates between instances of different classes or categories in the dataset. Unlike F-test, the predictor importance score don't increase as the number of beans per image increases. Although the weights are different, the 4 samples display the same list of the most relevant features. The most relevant features are those that have positive weights.

Using both methods and based on several experiments, a feature is said to be relevant when its F-test score is greater than 55 and its RReliefF weight is positive. Thus, the relevant features for predicting moisture content are energy and entropy of rg, yb, Cb, Cr, a^*, b^* and h^* components. This result is confirmed by the four samples. Of these features, energy-based variables are better than entropy-based features. On can also observe that the energy-based features have approximately the same scores in the F-test and RReliefF.

As a reminder, the database used for feature selection is made up of a set of observations from several different batches of beans. Thus, the redundancy of features can be explained by poor correlation with the moisture content or by the instability of features. In the case of instability, the feature may correlate well with the moisture content for a given batch. However, the values of this feature vary from batch to batch. The poor results for the features selected for samples E10 and E30 can be explained by the wide range of cocoa bean colors. These samples don't contain enough beans to take into account the maximum colors of cocoa bean.

The almost identical scores of energy-based features on the one hand and entropy-based features on the other may be due to the multi-collinearity between features. To confirm this hypothesis, we calculated the Variance Inflation Factors (VIF) of each selected feature by equation (32). VIF values range from 57.54 to $1.1x10^{+6}$. These values, being well above 10, show that the selected features are highly correlated [MONT21].

$$VIF_j = \frac{1}{1-R_j^2} \tag{32}$$

where R_j^2 is the coefficient of multiple determination obtained from regressing predictor x_j on other predictors. As the selected variables are highly correlated, if they are directly included in the models, this can create instability and over-fitting model due to the inflation of regression coefficients. Thus, for the model analysis in the next paragraph, PCA is used to select decorrelated variables.

5.2 Machine Learning models

The data is randomly divided into training and testing sets 20 times during the 20 training sessions of the models. The models are analyzed on two sets of predictors. The first set is made up of all selected predictors, and the second set is made up of energy-based predictors. The scikit-learn library in Python is used for this analysis. Coefficient of determination (R^2) and Root Mean Square Error $(RMSE)$ are the metrics used to evaluate models, by calculating the mean and SD of the five datasets. The hyper-parameters of each model are found with the Grid Search Cross-Validation (GridSearchCV). The hyper-parameters of SVR/RBF are regularization parameter C =10, $\gamma = 0.055$ and $\varepsilon = 0.01$. Ridge trace using all predictors gives ridge parameter $\alpha = 0.001$. For all selected predictors, there are 8 principal components for MLR, SVR, DTR, and PLSR models, and 35 principal components for the polynomial model with interaction. For the energy-only predictors, 5 principal components for MLR, SVR, DTR, and PLSR models, and 30 principal components for the polynomial model with interaction.

Table 1 and table 2 show the standard deviation (SD) and the mean of evaluation metrics for the models. SD is used to assess the stability of models. Concerning model training with all selected predictors (Table 1), more than 94% of proportion of variance in the MC is predicted from the predictors, for polynomial, SVR and DTR models. These models have mean errors of less than 5% of MC. On the other hand, for Ridge and PLSR models, less than 90% of variability in the training MC explained by the models and have mean errors of more than 5% of MC. The standard deviations of coefficient of determination for all models are less than 1%, indicating their stability during training. When testing the models with all selected predictors (Table 1), less than 90% of variability in the training MC explained by the models and less than 6% of mean errors of MC. There is a significant difference between the training and test of Polynomial, SVR, and DTR models, but this is not the case for Ridge and PLSR models. We also note that the SDs of testing are larger, more than 2% for the coefficient determination. The same observation for models trained with the energy features. However, the predictions with all selected predictors are better than energy predictors. All the models studied, for both sets of predictors, have results far better than Hashimoto et al, who obtained a coefficient of determination of 0.67 by predicting water content with near-infrared diffuse reflectance spectra [HII09]. These results are still inferior to those obtained using drying time as a predictor [IGO15], [KAR18].

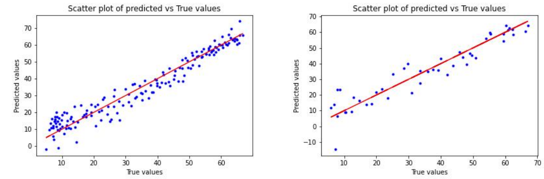
Figure 7a to Figure 11b displays the moisture content predicted value by the model versus the true value. The training and test data fit well on the straight line for the

		$R^2_{training}$		$R^2_{testing}$		$RMSE_{training}$		$RMSE_{testing}$	
Models	hyperparameters	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*
PCA + Polynomial	$degree = 2$	0.94	0.004	0.83	0.11	4.81	0.2	7.59	2.20
PCA + SVR	rbf	0.95	0.003	0.90	0.02	4.54	0.16	6.02	0.66
PCA + DTR	$depth = 7, min-samp.-spl. = 15$	0.96	0.004	0.86	0.07	3.85	0.22	7.00	1.83
Ridge R.	$alpha = 0.001$	0.89	0.01	0.86	0.08	6.50	0.43	7.01	1.66
PLSR	$component = 8$	0.86	0.01	0.83	0.09	7.24	0.41	7.70	1.90

Table 1: results of training and testing models for all selected predictors

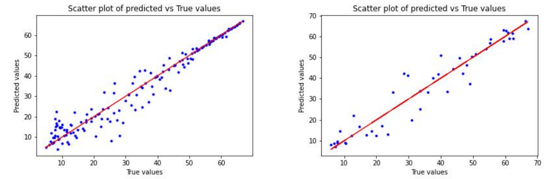
		$R^2_{training}$		$R^2_{testing}$		$RMSE_{training}$		$RMSE_{testing}$	
Models	hyperparameters	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*
PCA + Polynomial	$degree = 2$	0.93	0.005	0.82	0.10	5.28	0.20	7.73	2.15
PCA + SVR	rbf	0.88	0.01	0.87	0.05	6.72	0.33	6.80	1.38
PCA + DTR	$depth = 5$, $min-samp.-spl. = 10$	0.94	0.004	0.83	0.07	4.59	0.20	7.59	1.62
Ridge R.	$alpha = 0.001$	0.86	0.01	0.84	0.07	7.20	0.55	7.36	1.53
PLSR	$component = 5$	0.85	0.01	0.84	0.07	7.52	0.45	7.54	1.60

Table 2: results of training and testing models for energy predictors



(a) (b)

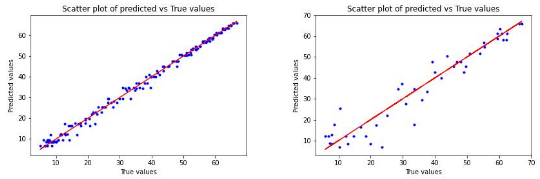
Figure 7: Scatter plot of predicted vs True values for PCA + Polynomial model for all selected predictors; (a) training (b) testing



(a) (b)

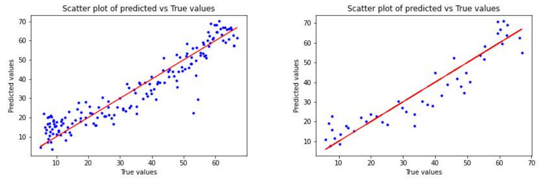
Figure 8: Scatter plot of predicted vs True values for PCA + SVR model for all selected predictors; (a) training (b) testing

non-linear models, Polynomial, SVR, and DTR (Figure 7a to Figure 9b. The linear models, Ridge and PLSR reveal outliers and significant deviations in the 30 to 50 MC range (Figure 10a to Figure 11b). This shows that the evolution of MC during drying is not linear. The tree-like structure is more effective in fitting all the training data as compared to the other two non-linear structures. It is important to note that the model performs well for MC values above 45%, while the fit is slightly less for values below 45%. Both the training and the test data show good results for the given model. The ideal moisture content for declaring cocoa to be dry is 7-8%. Thus, the most important MC range is below



(a) (b)

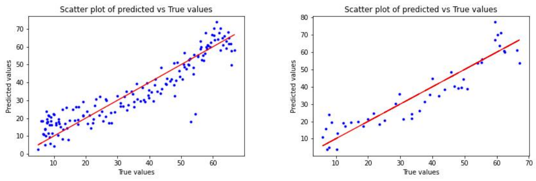
Figure 9: Scatter plot of predicted vs True values for PCA + DTR model for all selected predictors; (a) training (b) testing



(a) (b)

Figure 10: Scatter plot of predicted vs True values for RIDGE model for all selected predictors; (a) training (b) testing

10%. Support Vector Regression predicts 95% of training moisture content, 90% of new moisture content, and well the moisture content below 10%. The standard deviation shows that this model is more stable than others. Therefore, For this particular study, it has shown the most promising results in predicting the moisture content of cocoa beans during the drying process. The model uses predictors based on the energy and entropy of rg, yb, Cb, Cr, a*, b*, and h* components. However, it's important to note that the mean error still high in predicting new data.



(a) (b)

Figure 11: Scatter plot of predicted vs True values for PLSR model for all selected predictors; (a) training (b) testing

6 CONCLUSION

This article aims to analyze the color features of unshelled cocoa beans during the drying process. Additionally, it studies various linear and non-linear Machine Learning models to predict moisture content based on color features. The article analyzes the mean, standard deviation, entropy, energy, kurtosis, and skewness of the components of the RGB, YCbCr, HSV, and Lab color spaces, without luminance components, using both the F-test and RReliefF methods. The analysis is performed on samples of 10, 30, 50, and 70 beans per batch. The color components that are most relevant for predicting moisture content during drying are derived from the energy and entropy of YCbCr, and Lab color spaces. The relevance of these components becomes more important as the number of beans in a batch increases. Additionally, the selected features are highly interdependent. Non-linear models provide more accurate moisture content predictions during drying than linear models, precisely Support Vector Regression with radial basis function performs better. To conclude, moisture content can be predicted during drying with color image features. This article opens the way for the study of cocoa beans moisture content prediction using image data. It provides relevant information on the evolution of water content during drying as a function of colour characteristics and also in the different colour spaces.

7 ACKNOWLEDGMENTS

This work is financed in part by Ministère des Affaires Etrangères - France from Service de Coopération et d'Action Culturelle (SCAC) of Embassy of France in Ivory Coast.

8 REFERENCES

[CAS23] E. Castillo-Orozco, O. Garavitto, O. Saavedra, et D. Mantilla, The Drying Kinetics and CFD Multidomain Model of Cocoa Bean Variety CCN51 , Foods, vol. 12, no 5, p. 1082, mars 2023, doi: 10.3390/foods12051082.

[CAS20] M. Castelli, F. M. Clemente, A. Popovic, S. Silva, et L. Vanneschi, A Machine Learning Approach to Predict Air Quality in California , Complexity, vol. 2020, p. e8049504, August 2020, doi: 10.1155/2020/8049504.

[CHO05] I.-G. Chong et C.-H. Jun, Performance of some variable selection methods when multicollinearity is present , Chemom. Intell. Lab. Syst., vol. 78, no 1-2, p. 103-112, juill. 2005, doi: 10.1016/j.chemolab.2004.12.011.

[DJE09] A. Djedjro, E. Assidjo, K. Patrice, et B. Yao. Mathematical Modelling of Sun Drying Kinetics of Thin Layer Cocoa (Theobroma Cacao) Beans, J. Appl. Sci. Res., vol. 5, p. 1110-1116, sept. 2009.

[HAS18] J. C. Hashimoto et al. Quality Control of Commercial Cocoa Beans (Theobroma cacao L.) by Near-infrared Spectroscopy , Food Anal. Methods, vol. 11, no 5, p. 1510-1517, mai 2018, doi: 10.1007/s12161-017-1137-2.

[HII09] C. L. Hii, C. L. Law, M. Cloke, et S. Suzannah. Thin layer drying kinetics of cocoa and dried product quality , Biosyst. Eng., vol. 102, no 2, Art. no 2, fÃ©vr. 2009, doi: 10.1016/j.biosystemseng.2008.10.007.

[HII11] C. L. Hii, C. L. Law, et S. Suzannah. Drying kinetics of the individual layer of cocoa beans during heat pump drying, J. Food Eng., vol. 108, no 2, p. 276-282, janv. 2012, doi: 10.1016/j.jfoodeng.2011.08.017.

[HUM10] E. M. Humston, J. D. Knowles, A. McShea, et R. E. Synovec. Quantitative assessment of moisture damage for cacao bean quality using two-dimensional gas chromatography combined with time-of-flight mass spectrometry and chemometrics, J. Chromatogr. A, vol. 1217, no 12, Art. no 12, mars 2010, doi: 10.1016/j.chroma.2010.01.069.

[IGO15] N. A. L. Igor, A. D. Clement, B. Kouakou, et A. N. Emmanuel, Modélisation de la cinétique de séchage des feves de cacao par des modeles semi-empiriques et par un réseau de neurones artificiels récurrent: cas du séchage microonde par intermittence, p. 16, 2015.

[KAR18] D. Karidioula, D. C. Akmel, N. E. Assidjo, et A. Trokourey, Modélisation du séchage solaire de feves de cacao par le Réseau de Neurones Artificiel , Int. J. Biol. Chem. Sci., vol. 12, no 1, p. 195, juin 2018, doi: 10.4314/ijbcs.v12i1.15.

[KAV21] M. Kaveh, R. Chayjan, I. Golpour, S. Poncet, F. Seirafi, et B. Khezri, Evaluation of exergy performance and onion drying properties in a multistage semi-industrial continuous dryer: Artificial Neural Networks (ANNs) and ANFIS models , Food Bioprod. Process., vol. 127, p. 58-76, February. 2021, doi: 10.1016/j.fbp.2021.02.010.

[MATFT24] Univariate feature ranking for regression using F-tests - MATLAB fsrftest - MathWorks France . Accessed: February 25, 2024. [Online]. Available: https://fr.mathworks.com/help/stats/fsrftest.html

[MATSG24] Segment Image and Create Mask Using Color Thresholder - MATLAB and Simulink - MathWorks France. Accessed: May 07, 2024. [Online]. Available: https://fr.mathworks.com/help/images/image-segmentation-using-the-color-thesholder-app.html

[MONT21] D. C. Montgomery, E. A. Peck, et G. G. Vining, Introduction to Linear Regression Analysis. John Wiley and Sons, 2021.

[OME14] N. Omer Fadl Elssied, O. Ibrahim, et A. Hamza Osman, A Novel Feature Selection Based on One-Way ANOVA F-Test for E-Mail Spam Classification , Res.J. Appl. Sci. Eng. Technol., vol. 7, no 3, p. 625-638, janv. 2014, doi: 10.19026/rjaset.7.299.

[RAS06] Color Image Processing Methods and Applications by Rastislav Lukac, Kostantinos N. Plataniotis

[ROB97] M. Robnik-Sikonja et I. Kononenko, An adaptation of Relief for attribute estimation in regression , presented in International Conference on Machine Learning, juill. 1997.

[ROD22] A. Rodríguez Sánchez, R. Salmerón Gómez, et C. García, The coefficient of determination in the ridge regression , Commun. Stat. - Simul. Comput., vol. 51, no 1, p. 201-219, janv. 2022, doi: 10.1080/03610918.2019.1649421.

[WAN14] X.-Y. Wang, B.-B. Zhang, et H.-Y. Yang, Content-based image retrieval by integrating color and texture features , Multimed. Tools Appl., vol. 68, no 3, p. 545-569, February. 2014, doi: 10.1007/s11042-012-1055-7.

Deep learning-based classification of breast tumors using selected subregions of lesions in sonograms

Christian Schmidt
Westfälische Hochschule
University of Applied Sciences
Neidenburger Strasse 43
45897 Gelsenkirchen
Germany
christian.schmidt@w-hs.de

Heinrich Martin Overhoff
Westfälische Hochschule
University of Applied Sciences
Neidenburger Strasse 43
45897 Gelsenkirchen
Germany
heinrich-martin.overhoff@w-hs.de

ABSTRACT

Breast cancer, a prevalent disease among women, demands early detection for better clinical outcomes. While mammography is widely used for breast cancer screening, its limitation in e.g., dense breast tissue necessitates additional diagnostic tools. Ultrasound breast imaging provides valuable tumor information (features) which are used for standardized reporting, aiding in the screening process and precise biopsy targeting. Previous studies have demonstrated that the classification of regions of interest (ROIs), including only the lesion, outperforms whole image classification. Therefore, our objective is to identify essential lesion features within such ROIs, which are sufficient for accurate tumor classification, enhancing the robustness of diagnostic image acquisition. For our experiments, we employ convolutional neural networks (CNNs) to first segment suspicious lesions' ROIs. In a second step, we generate different ROI subregions: top/bottom half, horizontal subslices and ROIs with cropped-out center areas. Subsequently these ROI subregions are classified into benign vs. malignant lesions with a second CNN. Our results indicate that outermost ROI subslices perform better than inner ones, likely due to increased contour visibility. Removing the inner 66% of the ROI did not significantly impact classification outcomes ($p = 0.35$). Classifying half ROIs did not negatively impact accuracy compared to whole ROIs, with bottom ROI performing slightly better than top ROI, despite significantly lower image contrast in that region. Therefore, even visually less favorable images can be reliably analyzed when the lesion's contour is depicted. In conclusion, our study underscores the importance of understanding tumor features in ultrasound imaging, supporting enhanced diagnostic approaches to improve breast cancer detection and management.

Keywords
breast tumor, classification, CNN, ultrasound, tumor subregions

1 INTRODUCTION

According to the WHO, breast cancer stands as the most prevalent cancer among women worldwide [Who24], highlighting the need for effective screening and diagnostic methods. As early detection is crucial for achieving favorable patient outcomes, methods like mammography are widely used for breast cancer screening. The integration of artificial intelligence assistance for mammography screening has shown promising results [Lan23], demonstrating its potential to enhance detection rates. However, the complexity of breast tissue composition and the challenges posed by dense breast tissue in particular underscore the necessity for additional diagnostic tools and modalities. In comparison to alternative imaging modalities, ultrasound breast imaging is a non-ionizing, cost-effective, highly mobile, real-time imaging modality, making it widely accessible in most healthcare settings around the world. Ultrasound imaging excels in differentiating between different types of suspicious masses (e.g., cystic and solid lesions), and provides information about lesion shape, size, internal appearance, and other characteristics, which are essential for uniform reporting [Men13]. By enabling precise targeting of suspicious masses, ultrasound also aids in guiding biopsy procedures.

Our objective is to enhance the diagnostic accuracy of breast tumor ultrasound imaging by improving the understanding of lesion characteristics. This will facilitate the development of better-targeted, more effective diagnostic approaches, leading to improved cancer detection and patient care. In previous work [Sch23] convolutional neural network (CNN) based classification of suspicious masses into benign vs. malignant lesions was performed in whole breast sonograms vs. regions of interest (ROIs). In that work, the ROI is defined

https://www.doi.org/10.24132/CSRN.3401.26

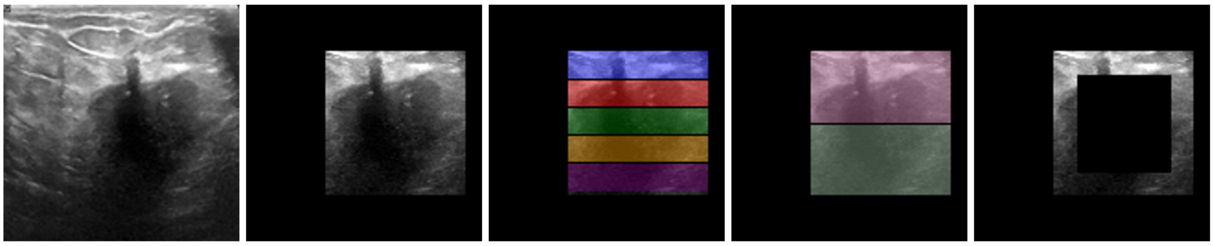


Figure 1: Overview of ROI subregions. From left to right: unprocessed ultrasound image (US), region of interest crop around the lesion mass ($\mathrm{ROI}_{\mathrm{whole}}$), subslices of ROI ($\mathrm{ROI}_{1-5}$), $\mathrm{ROI}_{\mathrm{top}}$ and $\mathrm{ROI}_{\mathrm{bottom}}$, ROI with cropped-out center area ($\mathrm{ROI}_{\mathrm{crop}\ x\%}$, here: $\mathrm{ROI}_{\mathrm{crop66}}$)

as a rectangular subregion around the suspicious lesion. Classification accuracy was considerably higher for ROI vs. whole sonogram (0.89 vs. 0.83). We hypothesize, that subregions of the lesion ROI yield similar classification accuracies to full ROIs. If so, a coarse subregion is sufficient for classification and the necessity of precise tumor segmentation and precise ROI positioning respectively diminishes. Our approach is to create different subregions of lesion ROIs (top/bottom half, horizontal subslices and ROIs with cropped-out center areas) and use a CNN to classify these subregions into benign and malignant lesions. By performing a comprehensive analysis of the classification results, we then identify which subregions contribute most to accurate lesion classification.

2 RELATED WORK

In the realm of breast tumor classification, researchers have explored various methodologies, each offering distinct advantages and insights into the diagnostic process. One prevalent approach relies heavily on deep learning techniques, particularly convolutional neural networks (CNNs), to analyze ultrasound tumor images. These CNN-based methods extract convolutional features through transfer learning, leveraging pre-trained architectures such as VGG16, YOLOv3, or GoogLeNet. Notable studies [Ald19, Chi19, Han17, Kal21] have demonstrated the efficacy of this approach in accurately classifying breast tumors based solely on learned features. However, deep learning methods come with certain trade-offs. While they offer automatic feature learning and high flexibility, requiring minimal manual intervention, they often demand large amounts of labeled data for training and significant computational resources. Additionally, deep learning models can be complex and challenging to interpret, potentially limiting their applicability in clinical settings where user scepticism towards AI is still widely present [Tam22, Che22] and interpretability of the employed algorithms is crucial for their trustworthiness.

In contrast, another avenue of investigation involves preselecting informative features from ultrasound images and subsequently employing classical machine learning algorithms for classification. Methods such as support vector machines and decision trees have been utilized in this context [Zha21, Cha14, Muh22]. These preselected features encompass a range of characteristics, including texture-based attributes derived from image data (e.g., gray-level co-occurrence matrix, contrast, homogeneity, energy) envelope and spectral-based properties extracted from raw radiofrequency data (e.g., K-distribution, Nakagami distribution). By focusing on specific features relevant to tumor characterization, these approaches offer a complementary perspective to the purely deep learning-based methodologies. Nevertheless, preselected feature classification methods also have their limitations. While they may offer greater interpretability and computational efficiency, they often rely on manual feature engineering, which can be labor-intensive and may not capture all relevant information in the data. Additionally, these methods may struggle to capture complex patterns and relationships in the data, potentially limiting their performance compared to deep learning approaches, especially in scenarios with large and heterogeneous datasets.

Recent investigations on a hybrid approach, such as [Dao20, Abh23, Saj23], suggest that integrating handcrafted features with deep learning-based convolutional features can lead to further improvements in classification accuracy. This capitalizes on the strengths of both methodologies, potentially enhancing the robustness and reliability of breast tumor classification systems.

Moreover, the discussion between whole image classification and subregion-based analysis has garnered some attention within the research community. Further stud-

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.

https://www.doi.org/10.24132/CSRN.3401.26

ies [Cha14, Sch23] have underscored the superiority of the analysis of selected tumor subregions over features derived from entire images. This finding aligns with the notion that localized analysis can provide more precise insights into tumor characteristics and facilitate more accurate classification outcomes.

3 MATERIALS AND METHODS

3.1 Dataset

In this study we employed the BUSI [Ald20] (breast ultrasound images) dataset for our analysis. This dataset comprises of 780 breast sonograms (437 benign masses, 210 malignant masses, 133 normal) of women aged between 25-75 years. For each sonogram in the dataset, ground truth lesion segmentations are available, enabling precise localization of the lesions. To ensure comparability with other state-of-the-art classification research, we focused exclusively on benign and malignant masses, excluding normal images, with no masses, from our consideration. Before training our models, all images were uniformly resized to dimensions of 128×128 pixels, and the grayscale values were normalized to span the range [0, 1]. This preprocessing step standardizes the image dimensions and intensity values, facilitating consistent model training and evaluation across the dataset. The root mean square (RMS) contrast, which is given by

$$\text{RMS Contrast} = \sqrt{\frac{1}{N}\sum_{x,y}(I(x,y) - \bar{I})^2}, \qquad (1)$$

where N is the total number of pixels in the image, $I(x,y)$ is the intensity value of a pixel at position (x,y), and $\bar{I}$ denotes the mean intensity value of all image pixels, serves as an indicator of image quality. Notably, the top half of the ROIs ($\mathrm{ROI}_{\mathrm{top}}$) exhibit an RMS contrast of 0.22 ± 0.12, while the bottom half ($\mathrm{ROI}_{\mathrm{bottom}}$) show an RMS contrast of 0.14 ± 0.07, with a statistically significant difference ($p < 0.0001$) based on the unpaired t-test.

3.2 ROI subregions

Because ultrasound B-mode imaging analyzes reflected waves, which disperse while passing through different tissues, image contrast decreases in deeper layers of tissue, as shown above. These B-mode images are also prone to artifacts, especially along the direction of the wavefront. This geometric condition is reflected in the implementation and choice of the image versions. We created the following four versions of ROI subregions (Fig. 1) to evaluate which lesion areas are most important for classification:

- $\mathrm{ROI}_{\mathrm{whole}}$: For the entire ROI ($\mathrm{ROI}_{\mathrm{whole}}$), a contour segmentation model [Sch23] is applied on the original unprocessed BUSI ultrasound dataset. A tight rectangular area is cropped around the resulting segmentation to obtain the desired ROI. This image version ensures that the entire lesion area, along with some surrounding tissue, is included for comprehensive analysis, capturing the full extent of tumor features.
- ROI_{1-5}: To evaluate the effects of artifacts and better understand the distribution of pertinent information among $\mathrm{ROI}_{\mathrm{whole}}$ images, we divide it into five, evenly spaced, horizontal subslices ROI_{1-5} (indexing applies top down). This segmentation strategy allows us to explore variations in lesion characteristics across different depths within the tissue, providing insights into the spatial distribution of features relevant to classification.
- $\mathrm{ROI}_{\mathrm{top}}$ and $\mathrm{ROI}_{\mathrm{bottom}}$: To evaluate the effects of potentially worse image quality (low contrast) in the lower lesion area, we horizontally divide $\mathrm{ROI}_{\mathrm{whole}}$ into two halves $\mathrm{ROI}_{\mathrm{top}}$ and $\mathrm{ROI}_{\mathrm{bottom}}$. By separately analyzing the upper and lower halves of the lesion, we can assess whether image quality variations (contrast) impact classification performance differently across different regions of the ROI.
- $\mathrm{ROI}_{\mathrm{crop}\,x\%}$: To gain better insights into the importance of the outer, contour regions vs. the inside of the lesions, we crop out a rectangle with dimensions of $x\%$ of the original $\mathrm{ROI}_{\mathrm{whole}}$ dimensions, to create $\mathrm{ROI}_{\mathrm{crop}\,x\%}$. This approach allows us to systematically evaluate the significance of the lesion's outer boundary and surrounding tissue in classification, providing valuable information on the spatial localization of discriminative features.

3.3 Classification experiments

To obtain the desired lesion ROI, we applied a *segmentation* model detailed in [Sch23]. It consists of seven convolutional layers (number of filters: 32, 64, 128, 256, 128, 64, 32; kernel size 3×3), which each perform feature extraction by applying learnable filters to the input images, followed by a max-pooling and a dropout layer to enhance model generalization. This segmentation process delineates the lesion region of interest from the surrounding tissue, enabling focused analysis and classification of suspicious masses. The network architecture (Table 1) we used for our *classification* experiments is also a CNN with sequential layers. The CNN architecture incorporates fully connected dense layers, which are widely employed in classification tasks. These dense layers process the flattened output from the convolutional layers to generate meaningful class predictions. The final output layer consists of two neurons for binary classification, providing predicted probabilities for each class. These probabilities serve as classifier parameters for our receiver operating

Layer (Type)	Output Shape	Channels
Conv	(64, 64, 4)	4
Conv_1	(32, 32, 8)	8
Conv_2	(16, 16, 16)	16
Conv_3	(8, 8, 32)	32
Flatten	(None, 2048)	-
Conv_4	(None, 32)	32
Dense	(None, 2)	2

Table 1: Summary of the network structure for our classification models.

characteristic (ROC) curve evaluation in the results section, enabling comprehensive performance assessment. During training, the classification model was initialized randomly and trained from scratch for 100 epochs using the binary cross-entropy loss function, the Adam optimizer and a learning rate of $\eta = 1 \cdot 10^{-4}$. The dataset was split into training, test, and validation data at a $60 : 30 : 10$ ratio for all experiments. Additionally, data augmentation in the form of horizontal flip and scaling was applied before training to improve model generalization. To account for potential variability in the dataset and improve the reliability of performance estimates, we employed a 10-fold cross-validation for each classification experiment. This technique involves partitioning the dataset into ten subsets, training the model on nine subsets, and evaluating its performance on the remaining subset. By repeating this process with different subsets for evaluation, we obtain more stable and representative performance estimates for our classification models. The area under the curve (AUC) metric was employed to compare classification accuracy across all models, providing a quantitative measure of model performance and efficacy in distinguishing between benign and malignant lesions.

We trained three distinct models for our evaluation: The first model was exclusively trained on $\text{ROI}_{\text{whole}}$ data (whole net, WN). The WN is used to evaluate the subslice experiment (ROI_{1-5}) as well as the crop experiment ($\text{ROI}_{\text{crop x\%}}$). In addition to the WN, we trained two additional models: the "top net" (TN) and the "bottom net" (BN). These models were specifically trained on ROI_{top} and $\text{ROI}_{\text{bottom}}$, respectively. The TN and BN models were designed to explore the effects of contrast variations across different regions of the lesion and to investigate whether robust classification models can be effectively trained using only half of the ROI information. By training separate models on the upper and lower halves of the lesion ROI, we aimed to discern any disparities in classification performance and ascertain the significance of ROI composition in model training and evaluation. In the following results section, the significance threshold was set at .05.

4 RESULTS

We found that our baseline model ($\text{AUC}(\text{ROI}_{\text{whole}}) = 0.922 \pm 0.032$) performed in line with recent, comparable studies [Ghe22, Byr21], which also used the BUSI dataset. These studies applied vastly more complex vision transformers as well as transfer learning-based, large convolutional models to address this task and to achieve similar classification accuracies to our baseline. This suggests that while more complex models may offer marginal improvements, our approach maintains competitiveness within the current state-of-the-art. Results of the subslice experiment (Fig. 2) showed that subslices ROI_1 and ROI_5 performed better than inner ones (Fig. 3). We observed that the model classified ROI_5 significantly more accurately than ROI_3 ($\text{AUC}(\text{ROI}_5) = 0.824 \pm 0.071$ vs. $\text{AUC}(\text{ROI}_3) = 0.760 \pm 0.052, p = 0.036$), while ROI_1 vs. ROI_3 did not reach statistical significance ($p = 0.074$).

We found that cropping out the inner 66% of the ROI (Fig. 4) did not lead to a significant deterioration in AUC ($\text{AUC}(\text{ROI}_{\text{whole}}) = 0.922 \pm 0.032$ vs. $\text{AUC}(\text{ROI}_{\text{crop66}}) = 0.897 \pm 0.058$, $p = 0.35$). AUC began to significantly increase with $\text{ROI}_{\text{whole}}$ vs. $\text{ROI}_{\text{crop75}}$ ($p = 0.016$).

Additionally, $\text{BN\_ROI}_{\text{whole}}$ performed significantly better than $\text{BN\_ROI}_{\text{bottom}}$ ($p = 0.045$), despite the upper half not being known during the training process (Fig. 5). The BN demonstrated superior generalization to $\text{ROI}_{\text{whole}}$ compared to the TN ($\text{AUC}(\text{BN\_ROI}_{\text{whole}}) = 0.922 \pm 0.032$ vs. $\text{AUC}(\text{TN\_ROI}_{\text{whole}}) = 0.876 \pm 0.051$, $p = 0.029$).

5 DISCUSSION

We hypothesized that ROI subsections would achieve classification results comparable to using the entire ROI. Our hypothesis was supported by the results, as ROI_{top} and $\text{ROI}_{\text{bottom}}$ indeed demonstrated classification performance similar to that of $\text{ROI}_{\text{whole}}$. However, for smaller subslices, this assumption was proven false. We observed a significant deterioration in classification accuracy in every subslice ROI_{1-5} compared to ROI_{top}, $\text{ROI}_{\text{bottom}}$ and $\text{ROI}_{\text{whole}}$. The observed decline in performance is likely attributed to an excessive reduction in contextual information. In addition, we observed that outer subslices tend to perform better, likely due to the preservation of more visible contour information. We further conducted verification tests by removing rectangular areas inside and outside the lesion, confirming that the inner lesion area is not critical for classification with our CNN model.

Additionally, we investigated whether analyzing only the upper or lower half of the lesion ROI would yield

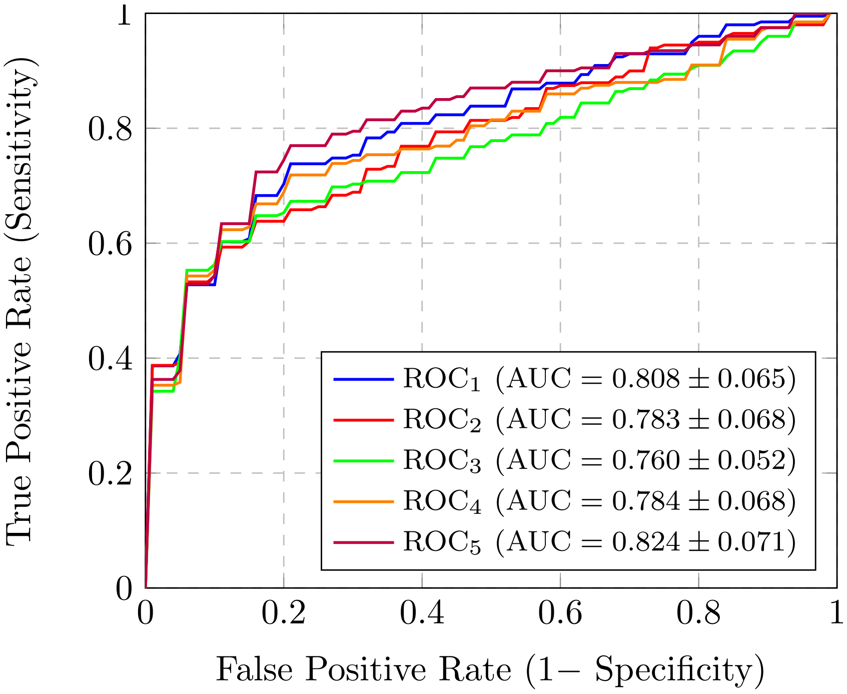


Figure 2: Results of subslice classification experiment.

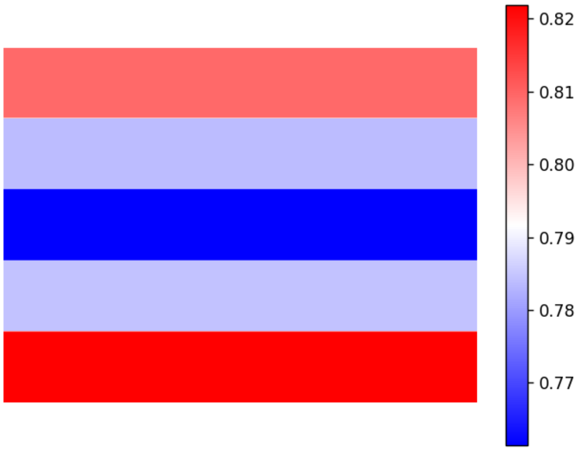


Figure 3: Color-coded results (AUC) of subslice classification experiment, mapped onto the ROI$_{\text{whole}}$ image version.

significantly different classification accuracies. Results showed, that even visually less favorable images can be reliably analyzed if the upper or lower edge of the lesion is depicted. Surprisingly, our experimental findings demonstrated that the variation in image contrast, particularly the substantially lower contrast observed in ROI$_{\text{bottom}}$, did not adversely impact the classification accuracy of our proposed method. Therefore, successful classification is possible even for images or image subregions of low quality, highlighting the robustness of our approach across varying imaging conditions.

To further strengthen the validity and generalizability of our findings, future research should include experiments conducted on other datasets beyond the BUSI dataset. While our results demonstrate promising classification performance using ROI subsections, it is essential to validate these findings on diverse datasets with varying characteristics such as imaging protocols, patient demographics, and lesion types. This broader exploration will help assess the external validity of our approach and its applicability across different clinical settings.

Additionally, conducting further experiments to compare our results with "heatmaps" generated by classification networks could provide valuable insights into the discriminative features utilized by our CNN model. Heatmaps visualize regions of interest within the images that contribute most to the classification decision, offering a deeper understanding of the underlying mechanisms driving our model's performance. By comparing the performance of our ROI subsections with the spatial distribution of discriminative features identified by heatmaps, we can gain further insights into the robustness and interpretability of our classification approach. These experiments would contribute to the ongoing efforts to enhance the transparency and interpretability of deep learning models in medical image analysis.

6 REFERENCES

[Abh23] Abhisheka B, Biswas S, Purkayastha B, Das S. (2023). Integrating Deep and Handcrafted Features for Enhanced Decision-Making Assistance in BreastCancer Diagnosis on Ultrasound Images. 10.21203/rs.3.rs-3276190/v1.

[Ald19] Al-Dhabyani W, Gomaa M, Khaled H, Fahmy A. Deep Learning Approaches for Data Augmentation and Classification of Breast Masses using Ultrasound Images. International Journal of Advanced Computer Science and Applications 10 (2019).

[Ald20] Al-Dhabyani W, Gomaa M, Khaled H, Fahmy A. Dataset of breast ultrasound images. Data in Brief. 2020;28:104863.

[Byr21] Byra M. Breast mass classification with transfer learning based on scaling of deep representations. Biomedical Signal Processing and Control. 2021;69:102828.

[Cha14] Chaudhury B et al., Using features from tumor subregions of breast DCE-MRI for estrogen receptor status prediction, 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), San Diego, CA, USA, 2014, pp. 2624-2629, doi: 10.1109/SMC.2014.6974323.

[Che22] Chen M, Zhang B et al., Acceptance of clinical artificial intelligence among physicians and medical students: A systematic review with cross-sectional survey, Frontiers in Medicine 2022 Volume 9, doi: 10.3389/fmed.2022.990604

[Chi19] Chiao JY, Chen KY, Liao K, Hsieh I, Zhang G, Huang TC. Detection and classification the breast tumors using mask R-CNN on sonograms. Medicine 98 (2019), e15200.

[Dao20] Daoud, M.I.; Abdel-Rahman, S.; Bdair, T.M.; Al-Najar, M.S.; Al-Hawari, F.H.; Alazrai, R. Breast Tumor Classification in Ultrasound Images Using Combined Deep and Handcrafted Features. Sensors 2020, 20, 6838.

[Ghe22] Gheflati B, Rivaz H. Vision Transformer for Classification of Breast Ultrasound Images. 2022.

[Han17] Han S, Kang HK, Jeong JY, Park MH, Kim W, Bang WC et al. A deep learning framework for supporting the classification of breast lesions in ultrasound images. Physics in Medicine and Biology 62 (2017).

[Kal21] Kalafi E, Jodeiri A, Setarehdan K, Ng W, Rahmat K, Mohd Taib NA et al. Classification of Breast Cancer Lesions in Ultrasound Images by Using Attention Layer and Loss Ensemble in Deep Convolutional Neural Networks. Diagnostics 11 (2021), p. 1859.

[Lan23] LÃ¥ng K, Josefsson V, Larsson AM, Larsson S. Artificial intelligence-supported screen reading versus standard double reading in the Mammography Screening with Artificial Intelligence trial (MASAI): a clinical safety analysis of a randomised, controlled, non-inferiority, singleblinded, screening accuracy study. The Lancet Oncology. 2023;24(8):936-44.

[Men13] Mendelson E, BÃ¶hm-VÃ©lez M. ACR BI-RADS Ultrasound. Reston, VA, 2013.

[Muh22] Muhtadi S. Breast Tumor Classification Using Intratumoral Quantitative Ultrasound Descriptors. Comput Math Methods Med. 2022 Mar 7;2022:1633858. doi: 10.1155/2022/1633858. PMID: 35295204; PMCID: PMC8920646.

[Saj23] Sajid U et al., Breast cancer classification using deep learned features boosted with handcrafted features. Biomedical Signal Processing and Control 2023 Volume 86, Part C, doi: 10.1016/j.bspc.2023.105353 Breast cancer classification using deep learned features boosted with handcrafted features

[Sch23] Schmidt C, Overhoff HM. Applicability of BI-RADS Criteria for Deep Learning-based Classification of Suspicious Masses in Sonograms. Bildverarbeitung für die Medizin 2023. Ed. by Deserno TM, Handels H, Maier A. Wiesbaden: Springer Fachmedien Wiesbaden, 2023:108-13.

[Tam22] Tamori H, Yamashina H, Mukai M, Morii Y, Suzuki T, Ogasawara K. Acceptance of the Use of Artificial Intelligence in Medicine Among Japan's Doctors and the Public: A Questionnaire Survey. JMIR Hum Factors. 2022 Mar 16;9(1):e24680. doi: 10.2196/24680.

[Who24] Fact sheet: Breast cancer, World Health Organization. https://www.who.int/newsroom/fact-sheets/detail/breast-cancer/. Accessed: 2023-07-31.

[Zha21] Zhang B, Song L, Yin J. Texture Analysis of DCE-MRI Intratumoral Subregions to Identify Benign and Malignant Breast Tumors. Front Oncol. 2021 Jul 8;11:688182. doi: 10.3389/fonc.2021.688182.

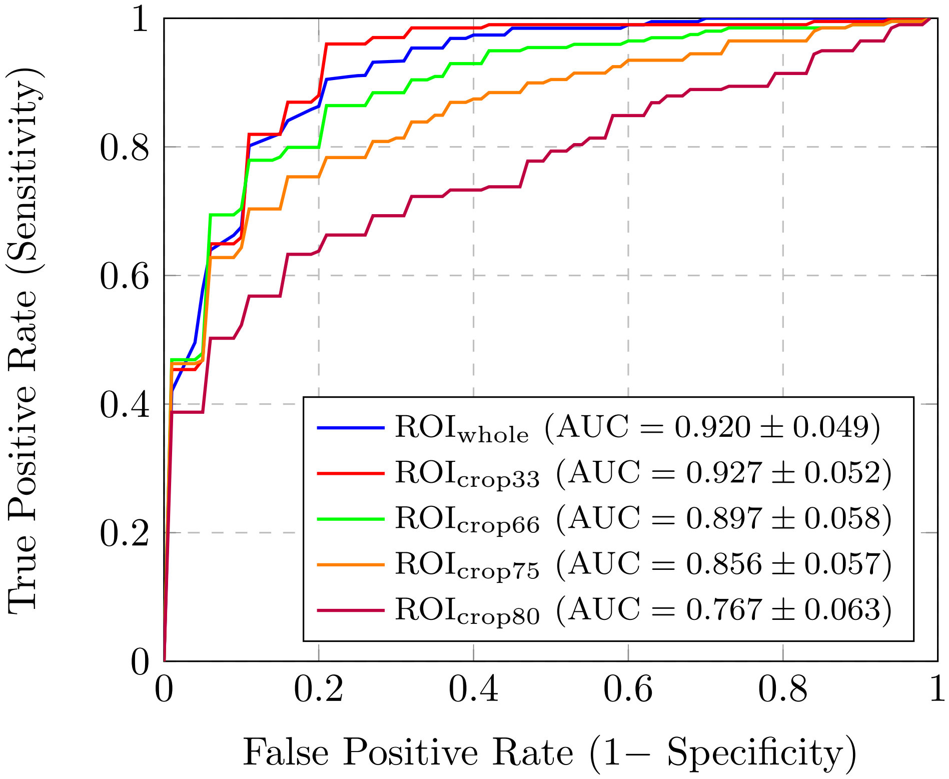


Figure 4: Results of crop classification experiment.

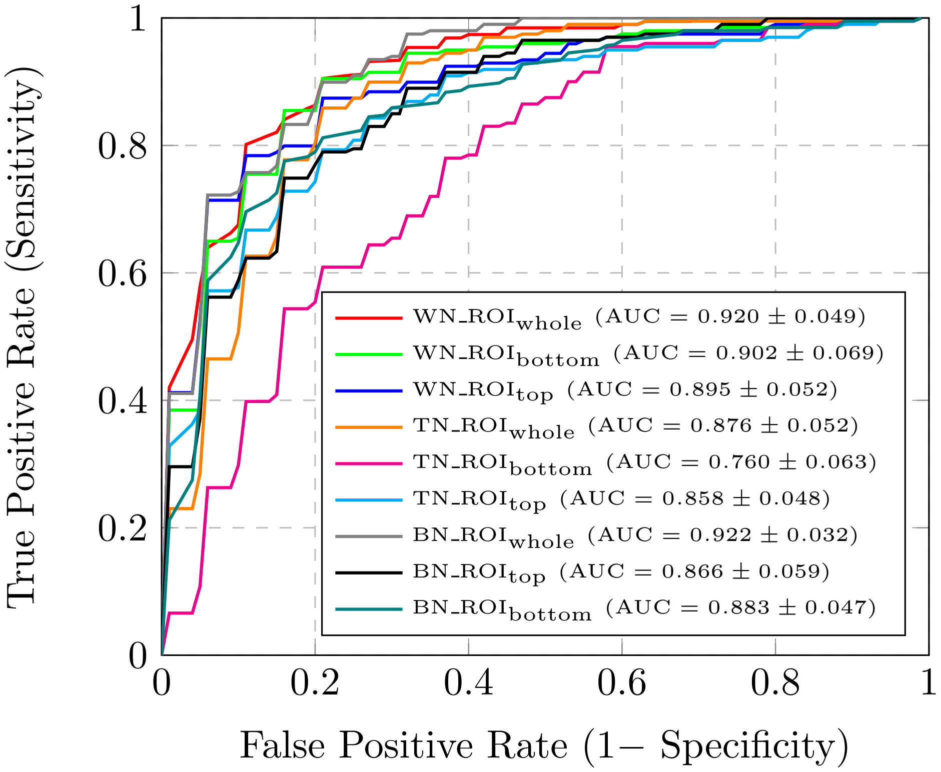


Figure 5: Results of top/bottom halves experiment.

LVCluster: Bounded Clustering using Laguerre Voronoi Diagram

Abu Quwsar Ohi
Department of Computer Science
University of Calgary
Calgary, AB, Canada
mdabuquwsar.ohi@ucalgary.ca

Marina L. Gavrilova
Department of Computer Science
University of Calgary
Calgary, AB, Canada
mgavrilo@ucalgary.ca

ABSTRACT

Clustering, a fundamental technique in unsupervised learning, identifies similar groups within a dataset. However, clustering algorithms encounter limitations when requiring a predetermined number of clusters/centroids/labels. This paper proposes a novel approach of clustering by integrating concepts from Voronoi diagrams in Laguerre geometry, namely, **L**aguerre **V**oronoi **Cluster**ing (LVCluster). Laguerre geometry introduces circles by adding radius weight metric to centroids, enabling dynamic exclusion from clustering criteria. Consequently, this approach offers flexibility by necessitating only one hyperparameter, an upper-bound value for the number of circles. LVCluster can be optimized using gradient descent and can be jointly optimized with deep neural network architectures. The experimental results indicated that LVCluster outperforms clustering algorithms when trained individually and jointly with deep neural networks on increased cluster centroids.

Keywords

Laguerre Geometry, Voronoi Diagram, Clustering, KMeans, Gradient Descending

1. INTRODUCTION

Clustering, an essential technique in data mining and machine learning, involves grouping data points into meaningful clusters based on their similarities. It is a fundamental tool for exploratory data analysis, pattern recognition, and knowledge discovery in various domains. Over the years, clustering has witnessed extensive research and application across diverse fields, including image analysis, document clustering, bioinformatics, customer segmentation, and anomaly detection, etc.

Clustering algorithms require hyperparameters based on which they extract the similarity of data. Due to the advancement of deep learning [Yan15a] methods, clustering algorithms are being incorporated with deep learning architectures [Maz20a]. The integration of clustering algorithms with deep learning architectures can be performed through either joint or independent training. Joint training indicates training deep learning (DL) models by adding DL objective loss and clustering loss during training. Independent training involves training DL separately with loss function and applying unsupervised clustering algorithms in the learned DL representation.

The most common clustering algorithm used in deep learning is KMeans [Stu82a], due to its simple clustering objective. KMeans clustering objective can be optimized using backpropagation, which gives an advantage of implementation with deep learning architectures. However, the required hyperparameter of the clustering algorithm often limits the usability of clustering algorithms. KMeans algorithm requires the number of centroids/classes as hyperparameters. Pre-defining the number of classes can be critical in unsupervised learning as the number of classes is sometimes impossible to measure in large datasets. On the other hand, density-based clustering algorithms [Mar96a] do not require the number of classes pre-defined. However, it requires a distance value to connect two components into the same group. Selecting the proper density value can be critical based on the representation and deviation of data distribution.

Deep learning architectures have also been applied to clustering data [Jun16a]. These methods aim to project data into a latent space while ensuring the data points in the space are clusterable. Deep clustering techniques have gained popularity due to their ability to automatically extract features from data. However, the objectives of clustering algorithms and deep clustering algorithms are different. Deep clustering algorithms produce projections of data that are clusterable and implement clustering algorithms inside the neural network. In contrast, classic clustering algorithms directly cluster data in the latent space. In both cases, one important problem that has always been an open question is the pre-defined number of clusters to generate. This work introduces a clustering algorithm that gives flexibility in pre-defining the number of clusters using computational

https://www.doi.org/10.24132/CSRN.3401.27

geometry data structure with the property of modern deep learning architecture.

The objective of this paper is to propose a clustering algorithm that does not require a fixed number of centroids or classes during training. Therefore, the clustering algorithm would offer flexibility in varying data distributions. Additionally, the algorithm has to operate without requiring labeled data, enabling unsupervised learning scenarios. Moreover, the algorithm should be optimizable using gradient descent to be jointly trained with deep neural network architectures end-to-end. With all these advantages, the clustering algorithm will be very similar to a deep learning layer, such as softmax, yet unsupervised and loosely dependent on the number of classes and labels.

Considering the above-mentioned constraints, this paper proposes a Laguerre Voronoi clustering algorithm, namely, LVCluster. The proposed algorithm requires one hyperparameter: an upper bound of the centroids/labels of the given input data. LVCluster algorithm dynamically groups clusters and excludes extra cluster regions based on the input data distribution. Therefore, LVCluster does not require knowledge of the exact number of classes in the given data. As a result, it is well suited for unsupervised learning on large datasets using deep neural networks (DNN) with an unknown number of classes.

The overall contributions of the paper are:

- The paper proposes a Laguerre Voronoi Diagram based clustering strategy for unsupervised clustering.
- The approach leverages the properties of Laguerre geometry to introduce adaptability to the identification of necessary classes during the training process.
- The proposed clustering algorithm is trained using backpropagation and is jointly trainable with deep learning methods.

The performance of the proposed clustering algorithm is evaluated in three spatial datasets: Congress voting dataset [Con87a], Iris dataset [Raf88a], and Breast cancer dataset [Wol95a]. Moreover, the performance of the proposed clustering algorithm is evaluated when it is jointly trained with deep neural networks to cluster speakers in an open-set scenario.

2. LITERATURE REVIEW

Clustering algorithms have been effectively studied and applied across various domains of machine learning. Clustering algorithms have been one of the fundamental approaches in unsupervised learning algorithms. Early clustering algorithms such as KMeans [Stu82a] is the most widely used partitioning-based algorithms due to its simplicity and efficiency. KMeans algorithm has different optimization techniques, among which Lloyd's [Stu82a] optimization is widely utilized. Lloyd's algorithm is also known as Voronoi iteration as it tries to partition a set of data into Voronoi cells. The center of each Voronoi cell is referred to as the site, which is learned in Lloyd's algorithm through iteration. Any data point in a Voronoi cell has a minimum distance to the Voronoi site. Lloyd's algorithm is specifically formulated to work on Euclidean spaces. In Euclidean space, Lloyd's algorithm cannot penalize unnecessary Voronoi sites on demand. As a result, the number of Voronoi cells must be correctly pre-defined. Apart from Lloyd's iterative algorithm, other approaches use gradients to learn optimal sites [Leo94a].

The popularity of deep learning algorithms has contributed to the increased importance of the KMeans algorithm, primarily due to its adaptability to be trained with gradient descending approach. As a result, various unsupervised deep learning algorithms utilized KMeans clustering's loss function jointly in the training process [Maz20a]. However, the only limitation that the KMeans algorithm faces is the pre-defined number of centroids or Voronoi cells. The pre-definition limits its usability as it is challenging to know the number of classes beforehand in self-supervised learning.

Density-based clustering algorithms, such as DBSCAN [Mar96a] and OPTICS [Mih99a], focus on identifying regions with high-density data space. However, these algorithms are not optimizable using a gradient descending approach. Density-based algorithms are prone to density parameters. As a result, it is challenging to integrate density-based algorithms with deep learning as the density distribution produced by the deep learning models can change over time.

Deep learning architectures can excellently produce clusterable data by learning the inherent representation of the given data. Further, they can generate clusterable embedding vectors based on appropriate loss functions. Deep embedded clustering (DEC) [Jun16a] is one of the fundamental approaches that produces clusterable embedding from image data using KMeans clustering loss. Although it performs excellently on image representation, it is still pruned to the fixed number of clusters/classes to produce. If the number of classes is approximated, DEC would fail to produce projection form data.

The idea of identifying the number of centroids in the KMeans algorithm is not new. XMeans [Pel20a] is an early improvement of the KMeans algorithm that finds appropriate cluster regions by repeated subdivision. Unsupervised KMeans [Sin20a] introduced a version of the KMeans algorithm that does not require the number of centroids as a hyperparameter. Even though the improved KMeans algorithms optimally approximate the centroids dynamically, they lack the ability to be trained using backpropagation. Hence,

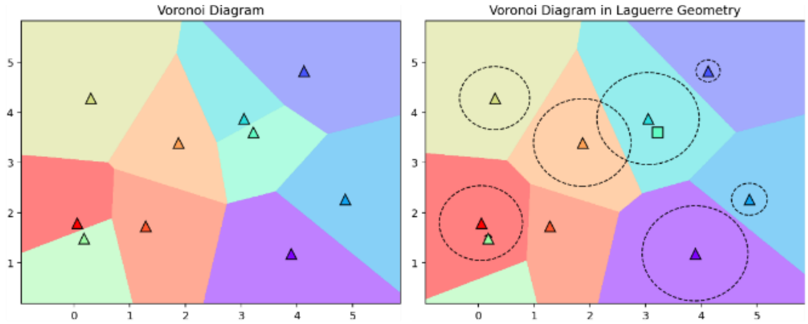


Figure 1 Difference of Voronoi diagram in Euclidean and Laguerre geometry.

recently enhanced KMeans algorithms cannot be implemented in a deep learning setup.

This paper introduces a novel clustering algorithm that inherits principles from principles of Laguerre geometry to dynamically determine the number of centroids. Instead of fixing the number of clusters, the algorithm requires an upper bound value of the number of clusters. Moreover, the clustering algorithm is trainable using backpropagation. As a result, it can be jointly trained with deep learning architectures simultaneously. The proposed clustering algorithm alleviates the need to fix the number of clusters, making the clustering algorithm truly unsupervised.

3. METHODOLOGY

The clustering algorithm inherits the basic properties of KMeans while incorporating the properties of Laguerre geometry. The algorithm can cluster spatial data. The algorithm requires an upper bound of the number of clusters. The clustering algorithm has two stages: a) initialization and b) optimization. Similar to classic KMeans, the proposed clustering method is sensitive to initial region selection. After the initialization, the circles are adjusted using backpropagation. First, the concept of the Voronoi Diagram in Laguerre geometry is discussed. Thereafter, the initialization and optimization steps are discussed in the following section.

Voronoi Diagram in Laguerre Geometry

In Laguerre geometry cluster circles are formed instead of cluster centroids. Circles in Laguerre Voronoi Diagram are set of points $C = \{c_1, c_2, \dots, c_k\}$, $c_i \in \mathbb{R}^d$ with a corresponding set of radius $R = \{r_1, r_2, \dots, r_k\}$, $r_i \in \mathbb{R}$. The coverage of circle c_i is the region that is of the minimum distance to the circle c_i. In Laguerre geometry, the coverage of a circle c_i can be controlled by increasing the value of the corresponding radius r_i and vise versa. The coverage of a circle is defined as a Voronoi polygon (convex polygon), which can be mathematically interpreted as,

$$V(c_i) = \cap_j \{P \in \mathbb{R}^d | D_L(c_i, P) \leq D_L(c_j, P)\} \qquad (1)$$

Here, $P \in \mathbb{R}^d$ is a set of points in the plane. $D_L(\cdot,\cdot)$ is a distance function in terms of Laguerre geometry derived as follows,

$$D_L(c_i, P) = d(c_i, P) - r_i^2 \qquad (2)$$

Intuitively, the equation measures the tangent line from point P to the circle $(c_i; r_i)$. $d(.,.)$ could be any distance metric function. Whenever a Voronoi diagram is drawn using the above formula of Laguerre geometry, the formed partition of the whole frame is called the Laguerre Voronoi Diagram (LVD). A gamut of applications in material sciences [Ale04a], biometrics [Che06a], and networks [Kef09a] benefitted from advanced algorithms of the Voronoi diagram and Laguerre Voronoi diagrams. One of the interesting properties of the Voronoi diagram is a Voronoi polygon can have zero area if it can be formed using one or more Voronoi polygons. Due to this property, LVD is different from Voronoi diagrams in Euclidean space, formed by KMeans and any other clustering algorithms. A circle can exist in LVD without covering any region, resulting in having no cluster assignment. Figure 1 depicts an example of the classic Voronoi Diagram and Voronoi Diagram in Laguerre Geometry. The triangles indicate centroids/sites, and the color background indicates the region/Voronoi cell of the centroids. Dashed circles demonstrate the radius of circles in Laguerre geometry. Squares indicate that the circle was not assigned any region/Voronoi cell, which is only possible in the Voronoi diagram in Laguerre geometry.

Laguerre Voronoi Clustering

The proposed algorithm leverages the property of an empty polygon in LVD to its advantage by setting an upper bound on the number of clusters. Therefore, the proposed algorithm can find the required number of cluster regions during training and shrink the remaining centroids. Figure 1 shows an example of the property. In Voronoi diagram based clustering (Figure 1-left), each centroid must have its Voronoi polygon on the space. On the contrary, LVD can dynamically shrink the area of Voronoi polygons to zero (Figure 1-

right). With proper optimization, LVCluster can learn the optimal polygon regions and disregard the excess centroids. This dynamic adaptation to the number of centroids is the novelty of the proposed LVCluster algorithm. In the following two sections, the initialization stage and optimization stage of the proposed algorithm are discussed.

3.1.1 Initialization

The initialization step defines circle centers and radii based on the input data. Initialization is an important step in clustering, as inaccurate initialization can cause the optimization to get trapped in local minima [Dav07a]. The initialization process of the proposed clustering algorithm follows the KMeans++ [Dav07a] initialization process, used in KMeans clustering. The circle centers c_i are first initialized using the KMeans++ initializer. Afterward, the radius r_i is derived from the minimum distance for each center to any other center. Using KMeans++ ensures that all the centers are placed around data points while maintaining distance among the other centers. Having different radii ensures coverage overlaps, which is to be optimized using a gradient descending algorithm.

3.1.2 Optimization

After proper initialization of the circles, their optimal placement is found using backpropagation. In the proposed algorithm the distance function to generate Voronoi polygon is stated as follows,

$$D_L(c_i, P) = d(c_i, P) - \sigma(r_i)^2 \quad (3)$$

Here $\sigma(\cdot)$ is sigmoid activation function used to give a non-linearity to the optimization problem. $d(.,.)$ is cosine distance function.

The objective of the clustering algorithm is to minimize the distance between the circle and the data points belonging to the Voronoi polygon of that circle. The objective can be mathematically interpreted as follows,

$$L(X, C) = \sum_{i=1}^{n} \min_{c_j \in C} D_L(c_j, x_i) \quad (4)$$

The function is used as a loss function and gradients of the center and radius are calculated based on the given loss function. Here, $x_i \in X$ is the set of data points given to the clustering algorithm as input. The circle center and radius are updated using the gradient descending algorithm with a fixed learning rate η.

3.1.3 Deep Learning Integration

The proposed LVCluster can be combined with deep learning architectures and can be trained jointly to achieve the clustering objective. The joint loss function can be derived as $L = L_{DNN} + L_C$. Here, L_{DNN} is the loss function of DNN that could be representation loss, reconstruction loss, or elbow loss. L_C is the clustering loss function derived in Eqn (4). Based on the architectural constraints, the clustering loss function can flow the gradients to the DNN. As the LVCluster algorithm adheres the same implementation strategy of KMeans clustering algorithm (disregarding the minor additional computation of $\sigma(r_i)^2$ in Eqn (3)), the computational complexity of both of the algorithms are identical.

4. EXPERIMENTAL RESULTS

Datasets

Three spatial datasets were used for the evaluation. The datasets are described below:

- Congress voting [Con87a] is a classification dataset containing two classes. The dataset has a total of 16 features explaining different conditions of voting. Some of the features were missing from the dataset, which were removed before training.
- Iris [Raf88a] is a classification dataset containing three classes of 50 instances each. It is a spatial dataset with four features. The dataset is widely used in statistics and machine learning.
- Breast cancer [Wol95a] dataset is a classification dataset containing two classes (malignant and benign). The dataset has 30 features extracted from a digitized image of a fine needle aspirate of a breast mass. The cancer dataset holds 569 instances in total. The features were standardized by removing the mean and scaling to unit variance.

Apart from the real-world datasets, synthetic datasets were used for the visualization of clusters. The synthetic datasets were generated using the sklearn [Fab11a] library with a standard deviation of 3.5.

Implementation

Three metrics were used for evaluation. The metrics are elaborated below:

- Accuracy: Due to an unsupervised learning strategy, the clustering algorithm generated pseudo labels for each data point. Therefore, the accuracy metric indicates the maximum match by mapping the pseudo labels with the ground truth labels.
- Adjusted Rand Index: The adjusted rand index is extracted and calculated from the contingency table determining the quality of the generated pseudo label depending on the ground truth.
- Normalized Mutual Information: Normalized mutual information extracts normalization of the mutual score by comparing the clustering algorithm-generated pseudo labels with the ground truth.

The three algorithms generate a real value score [0, 1]. A higher score from the metrics would indicate the quality of the pseudo labels compared to the ground truth. However, the number of centroids defined in the

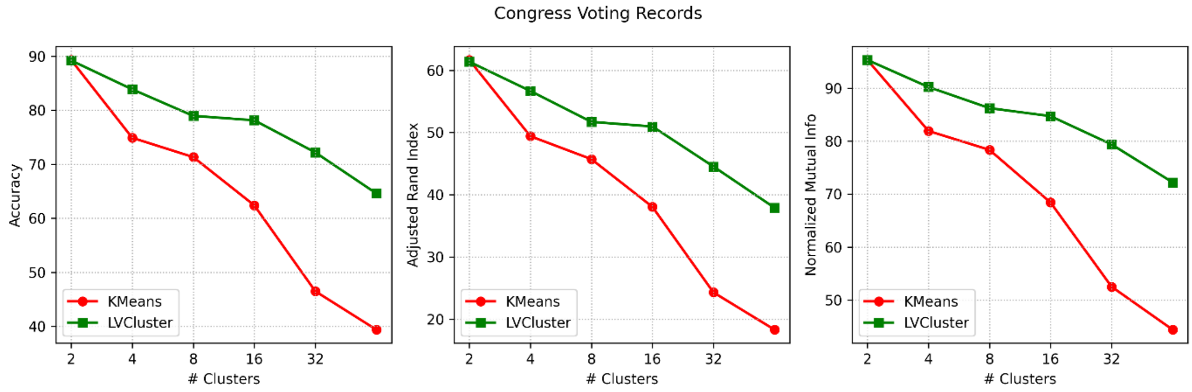


Figure 2 Performance comparison on congress voting records dataset.

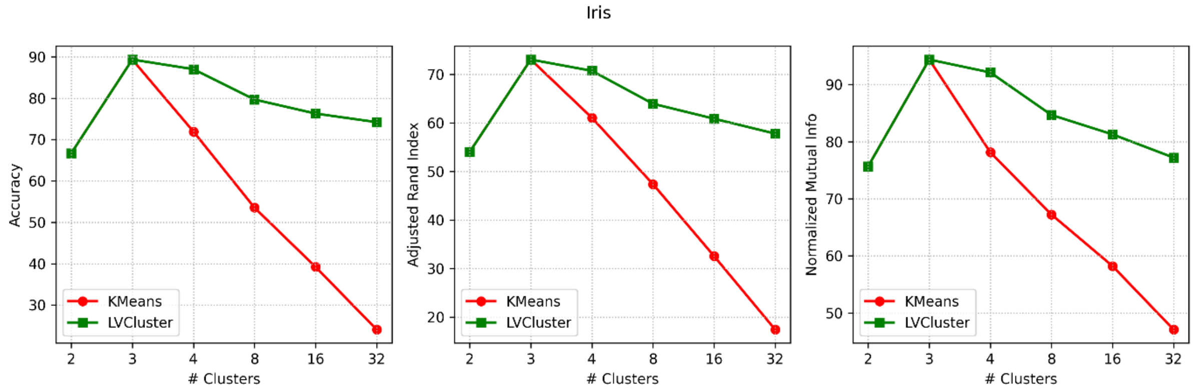


Figure 3 Performance comparison on Iris dataset.

clustering algorithm would not be the same as the number of ground truth labels. Therefore, the metrics' higher score would evaluate how well the clustering algorithm identifies the number of centroids along with the clustering problem relative to the ground truth.

PyTorch [Ada19a] was used to build the clustering algorithms that support automatic differentiation. The clustering algorithms were additionally trained for open-set speaker recognition, aiming to investigate their performance and behavior when applied to the outputs of DNNs. The training criteria were implemented from [Abu23a]. The models were trained on the VoxCeleb1 [Ars17a] dataset in an unsupervised strategy and tested on different speakers on the VoxCeleb1 [Ars17a] dataset.

Comparison Analysis

The analysis explores the influence of the initial number of clusters, which serves as a hyperparameter provided to the clustering algorithm during the initialization stage. Figure 2 reports a performance comparison of the KMeans and LVCluster algorithms on congress voting records dataset. Both clustering algorithms show optimal performance when given the accurate number of centroids. KMeans and LVCluster show similar performance when the centroid was set to 2. Increasing the number of centroids causes a reduction in performance for the KMeans algorithm. In contrast, the LVCluster shows less performance degradation with an increasing number of clusters.

Figure 3 depicts a performance comparison on the Iris dataset with three classes. Both clustering algorithms show degraded performance when the number of centroids was set to two, which is less than the actual number of classes. Initializing the clustering algorithms with three shows the best performance for both algorithms. Further increasing the number of centroids causes a reduction of performance for the KMeans algorithm. LVCluster also shows performance degradation while increasing centroids. However, compared to LVCluster degradation is slighter than KMeans performance degradation.

Figure 4 reports a performance comparison on the breast cancer dataset with two classes. Both clustering algorithms show a similar trajectory of degradation. Each class of the dataset contains smaller and independent sub-clusters. Therefore increasing the initial number of clusters in the clustering algorithm causes the clustering algorithm leads it to treat the sub-clusters as separate classes. As a result, the LVCluster struggles to effectively reduce the number of cluster

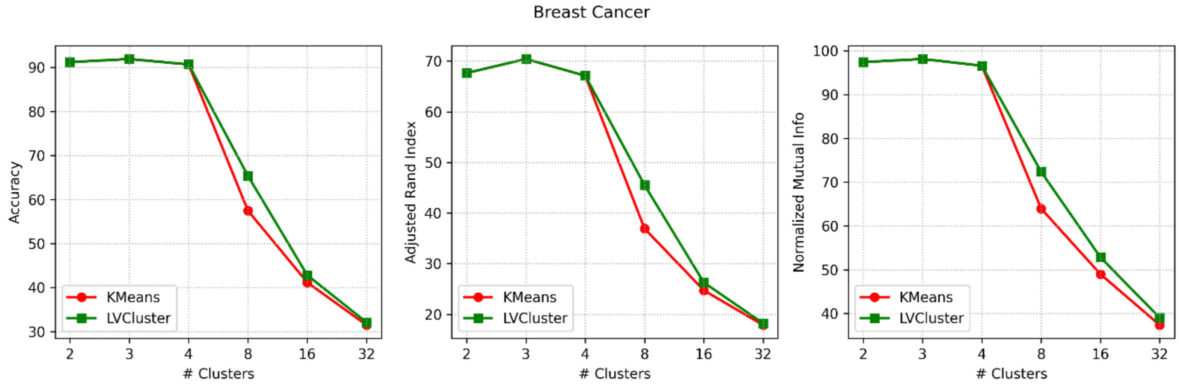


Figure 4 Clustering performance comparison on breast cancer classification dataset.

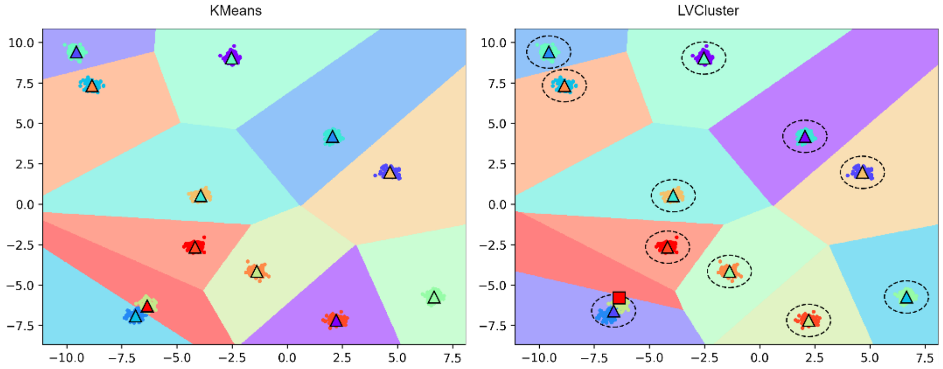
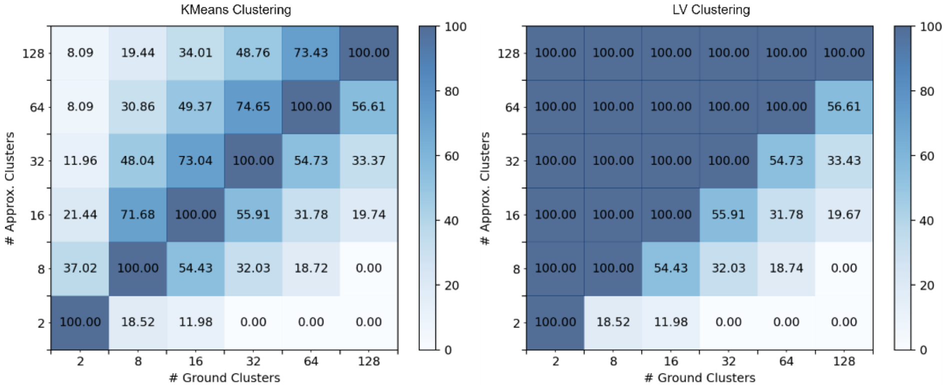


Figure 5 Voronoi diagram comparison of KMeans and LVCluster.



KMeans Clustering

# Approx. Clusters	2	8	16	32	64	128
128	8.09	19.44	34.01	48.76	73.43	100.00
64	8.09	30.86	49.37	74.65	100.00	56.61
32	11.96	48.04	73.04	100.00	54.73	33.37
16	21.44	71.68	100.00	55.91	31.78	19.74
8	37.02	100.00	54.43	32.03	18.72	0.00
2	100.00	18.52	11.98	0.00	0.00	0.00

LV Clustering

# Approx. Clusters	2	8	16	32	64	128
128	100.00	100.00	100.00	100.00	100.00	100.00
64	100.00	100.00	100.00	100.00	100.00	56.61
32	100.00	100.00	100.00	100.00	54.73	33.43
16	100.00	100.00	100.00	55.91	31.78	19.67
8	100.00	100.00	54.43	32.03	18.74	0.00
2	100.00	18.52	11.98	0.00	0.00	0.00

Figure 6 Adjusted rand index on different cluster choices.

regions relative to the original number of clusters defined in the dataset.

In the above comparisons, LVCluster shows an upper-bound performance compared to the KMeans algorithm. Moreover, LVCluster stabilizes the performance when the number of clusters were increased. As LVCluster can remove unnecessary centroids from the dataset, it can detect sub-optimal cluster representation. However, the performance of LVCluster on increasing the number of centroids depends on how well the clusters are distributed.

Figure 5 shows an example of the Voronoi diagram of the centroids. The example has 12 cluster regions where 10 of the cluster regions are well-separated. Regardless, two of the cluster regions are tightly bound together. From the perspective of the distribution of the data, there are 11 centroids as two of the regions are so close that they can be considered as one cluster region. KMeans clustering with 12 centroids would generate 12 regions. However, as the data is not well-separated, the LVCluster would generate 11 regions considering the two tightly bounded regions as one. From the perspective of data

Table 1 Comparison of clustering algorithms when trained with DNN.

Algorithm	#Clusters = 40			#Cluster = 128		
	Accuracy	Adjusted Rand Index	Normalized Mutual Information	Accuracy	Adjusted Rand Index	Normalized Mutual Information
KMeans	98.9	96.26	99.4	14.64	20.47	32.73
LVCluster	99.9	96.26	99.9	99.2	95.94	99.8

distribution, performance is a relative factor. Excessive noise and merged regions can cause LVCluster to generate fault assumptions based on the actual classification task.

Figure 6 exhibits a comparison of the adjusted rand index varied by the number of clusters on a well-separated data distribution. LVCluster shows the best result whenever the input number of clusters is greater than the number of actual clusters in the dataset. The overall comparisons highlight that LVCluster can dynamically reduce the required number of clusters based on the data distribution to gain better performance than KMeans clustering.

Table 1 shows the performance comparison of the KMeans and LVCluster algorithm when trained jointly with DNN architecture [Abu23a] to recognize speakers in unsupervised open-set conditions. In an open-set problem, a model is trained and tested on two different datasets with two different speaker sets. Therefore, the number of speakers in the testing set (40 speakers) is unknown to the model. When using the KMeans algorithm to perform clustering on the speaker embeddings, the clustering algorithm generates the same performance as LVCluster when the number of clusters is set to the same as the number of speakers in the test dataset. However, increasing the number of clusters to 128 shows that the performance of the KMeans clustering algorithm degrades drastically. In contrast, the performance of LVCluster slightly reduces. LVCluster dynamically adjusts the number of clusters and finds the best possible clustering representation from the speaker embedding space. Therefore, it can be validated that LVCluster can be useful when the number of clusters of a distribution is unknown, yet the distribution contains a clusterable representation.

5. CONCLUSION

This paper proposes LVCluster, a novel clustering algorithm that inherits the property of Laguerre geometry to dynamically determine the number of centroids. The algorithm requires an upper bound number of clusters as hyperparameters, which often reduces the necessity to identify the number of classes in an unknown dataset. LVCluster shows performance gain than KMeans whenever the number of cluster centroids is set higher than the actual number of cluster centroids. The algorithm is optimized using backpropagation, thereby opening up opportunities for its application in deep learning architectures.

6. ACKNOWLEDGMENTS

The authors acknowledge the Natural Sciences and Engineering Research Council (NSERC) Discovery Grant funding, as well as the NSERC Strategic Partnership Grant (SPG) and the University of Calgary Transdisciplinary Connector Funding for the partial funding of this project.

7. REFERENCES

[Yan15a] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. nature 521, 7553 (2015), 436–444.

[Maz20a] Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. 2020. Deep k-means: Jointly clustering with k-means and learning representations. Pattern Recognition Letters 138, (2020), 185–192.

[Stu82a] Stuart Lloyd. 1982. Least squares quantization in PCM. IEEE transactions on information theory 28, 2 (1982), 129–137.

[Mar96a] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, and others. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd, 226–231.

[Jun16a] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In International conference on machine learning, PMLR, 478–487.

[Pel20a] Pelleg, Dan, and Andrew W. Moore. 2000. X-means: Extending k-means with efficient estimation of the number of clusters. In ICML, vol. 1, pp. 727-734.

[Sin20a] Sinaga, K.P. and Yang, M.S., 2020. Unsupervised K-means clustering algorithm. IEEE access, 8, pp. 80716-80727.

[Con87a] Congressional Voting Records. 1987. UCI Machine Learning Repository.

[Raf88a] R. A. Fisher. 1988. Iris. UCI Machine Learning Repository.

[Wol95a] Wolberg and W. Street. 1995. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository.

[Leo94a] Leon Bottou and Yoshua Bengio. 1994. Convergence properties of the k-means algorithms. Advances in neural information processing systems 7, (1994).

[Mih99a] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering points to identify the clustering structure. ACM Sigmod record 28, 2 (1999), 49–60.

[Ale04a] Alexey V Anikeenko, MG Alinchenko, VP Voloshin, Nikolai N Medvedev, Marina L Gavrilova, and P Jedlovszky. 2004. Implementation of the Voronoi-Delaunay method for analysis of intermolecular voids. In Computational Science and Its Applications–ICCSA 2004: International Conference, Assisi, Italy, May 14-17, 2004, Proceedings, Part III 4, Springer, 217–226.

[Che06a] Chengfeng Wang, Marina Gavrilova, Yuan Luo, and Jon Rokne. 2006. An efficient algorithm for fingerprint matching. In 18th International Conference on Pattern Recognition (ICPR'06), IEEE, 1034–1037.

[Kef09a] Kefeng Xuan, Geng Zhao, David Taniar, Bala Srinivasan, Maytham Safar, and Marina Gavrilova. 2009. Network Voronoi diagram based range search. In 2009 International conference on advanced information networking and applications, IEEE, 741–748.

[Dav07a] David Arthur, Sergei Vassilvitskii, and others. 2007. k-means++: The advantages of careful seeding. In Soda, 1027–1035.

[Fab11a] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and others. 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research 12, (2011), 2825–2830.

[Ada19a] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and others. 2019. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems 32, (2019).

[Abu23a] Abu Quwsar Ohi and Marina L Gavrilova. 2023. A Novel Self-supervised Representation Learning Model for an Open-Set Speaker Recognition. In International Conference on Computer Information Systems and Industrial Management, Springer, 270–282.

[Ars17a] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. Voxceleb: a large-scale speaker identification dataset. arXiv preprint arXiv:1706.08612, (2017).

https://www.doi.org/10.24132/CSRN.3401.27

From Sources to Solutions: Enhancing Object Detection Models through Synthetic Data

Eduard Bartolovic
Munich University of Applied Sciences
Germany, Munich, Bavaria
eduard.bartolovic@hm.edu

Tobias Höfer
Munich University of Applied Sciences
Germany, Munich, Bavaria
tobias.hoefer@hm.edu

Clemens Hage
BSH Hausgeräte GmbH
Germany, Munich, Bavaria
clemens.hage@bshg.com

Alfred Nischwitz
Munich University of Applied Sciences
Germany, Munich, Bavaria
alfred.nischwitz@hm.edu

Abstract

Object detection, a fundamental task in computer vision, plays a crucial role in various applications such as autonomous driving, surveillance, and robotics. However, training models for this task require vast amounts of high-quality data, often involving labor-intensive manual labeling. Synthetic data, a promising alternative, remains an active area of research. This paper presents a comprehensive exploration of different object sources for the use of synthetic data in enhancing object detection models. We investigate various synthetic data generation techniques to implant objects into a scene, with a focus on enhancing training data diversity. These objects are either gathered from the training dataset itself using SegmentAnything as a new supervised self augmentation technique or imported from external sources, including a photobox with a rotating table and web scraping of online shops. Moreover, our study delves into the development of a placement logic that gradually evolves from placing objects randomly to placing objects in physically correct orientations to mimic the real world data. We investigate the use of different blending techniques. The outcome of our study demonstrates that synthetic images, when integrated with an existing real training set, substantially improve the object recognition accuracy of the model without compromising inference time. Our code can be found at `https://github.com/EduardBartolovic/synthetic-data-generation`.

Keywords

Synthetic Data Generation, Data Augmentation, Domain Randomization, Object Detection, SegmentAnything, YOLOv5

1 INTRODUCTION

In the current landscape, object detection algorithms and model architectures stand as remarkably powerful tools [1]. However, the efficacy of object detection models is profoundly influenced by the availability and diversity of training data. Traditional labeling relies heavily on manual labor, which is often time-consuming, error-prone and expensive. Occasionally, data privacy regulations further complicate the collection of substantial data volumes. Moreover, obtaining real-world data that covers a wide range of scenarios can be challenging or even impractical, especially in niche areas where there is no large publicly-accessible dataset readily available. In some cases, object detectors need to be trained for future scenarios before real training data is available, such as when introducing a new product. This has led researchers to explore alternatives, such as the creation of synthetic data. Several synthetic data generation techniques have gained popularity, including 3D Rendering [2]–[4], Generative Models [5]–[9] and 2D Image Implantation [10], [11]. While 3D rendering can produce more realistic scenes and objects, it requires significant computational resources and modeling effort compared to the more straightforward 2D image implantation. Generative models, such as GANs or diffusion models, are known to be challenging to train and can be computationally expensive, and they are not yet fully capable of generating realistic images across all categories [8]. 2D implantation with our proposed refinement techniques emerges as a practical solution for easy synthetic data generation. We propose a workflow involving the gathering and cropping of objects from different sources and their implantation into a scene. This approach, detailed further in Section 4, is notably simpler than other methodologies, making it suitable for a wide range of use cases. In our study, we enhance the 2D image im-

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.

https://www.doi.org/10.24132/CSRN.3401.28

plantation technique from [10], [11], focusing more intensely on the object sources and also applying it to a more modern object detection algorithm.

One key aspect of synthetic data generation is the origin of placed objects. Notably, prior research [10], [11] has frequently overlooked the origin of objects employed in synthetic data generation, with a tendency to favor a single object source. In contrast, this study seeks to provide comprehensive insights into this crucial aspect of the generation process. From examining the origins of objects used in synthetic data generation, we find that they typically stem from either 3D CAD models or photographic representations. We conducted an assessment of both internal and external object sources to augment the scope of our training data. To extract objects from the training data we used SegmentAnything [12] as a self augmentation method. This technique takes advantage of the existing annotated training dataset to extract objects, thereby contributing to the creation of synthetic data that closely adheres to the distribution of real-world objects. This approach is a unique form of supervised self augmentation. In addition to the self augmentation method, we also explore the incorporation of foreign objects sourced from external sources. These foreign objects are gathered via web scraping or a 360° photobox. By introducing objects from other sources, we aim to enhance the adaptability of the object detection model to new scenarios and unforeseen objects. This cross-domain synthesis holds the potential to imbue the model with a broader perceptual scope, facilitating its performance in scenarios beyond those present in the original training dataset. In summary, our contributions are as follows:

- **A Scalable Method for Synthetic 2D Data Integration:** We introduce an easy, comprehensive, and highly scalable method for seamlessly integrating synthetic 2D data into the training of object recognition models. This approach significantly reduces the need for human labeling, making it more efficient and adaptable to enlarge the training dataset without affecting inference time. This method is more accurate and less error-prone than traditional labeling. This is a solution, particularly in scenarios where there is a constraint on the availability of training images and labeling resources.

- **Overview of suitable object sources:** Additionally, we provide a comprehensive overview of suitable object sources, like reusing training dataset objects, adding objects from webscraped webshops, and a photobox with a rotating table. This information equips practitioners to choose the most suitable source based on their specific use case.

- **Integration of Synthetic data into realistic data:** We explore how to effectively mix generated data with real data, aiming to discover the ideal balance between the two.

2 RELATED WORKS

In an ideal scenario, synthetic data would seamlessly merge with real-world data, creating a high-fidelity blend with a minimal reality gap. However, achieving this perfection across the board is often hindered by technological limitations or becomes feasible only through an impractical allocation of resources. The question arises: Is it even necessary to make synthetic data perfectly realistic? According to [13], absolute realism is not always essential. A workaround called domain randomization can be employed. This technique introduces random variations to the training data, including changes in lighting, backgrounds, object placements, and more. The concept behind domain randomization is to expose the model to a diverse set of situations during training, ensuring that the synthetic domain encompasses a wide range of possibilities. This approach aims to equip a model trained on synthetic data to perform well in real-world scenarios [13]. However, a minimum level of realism remains necessary and beneficial for success. To achieve this, it is valuable to explore past research in synthetic image data generation. While this study primarily focuses on 2D image composition, insights from other methodologies can provide valuable perspectives. All methods can be categorized into four main approaches:

2D Image Composition: In this approach, 2D images are incorporated by implanting them into another image. It stands out as the simplest among the considered methods. This approach has been used by some studies [10], [11]. While [10] uses a placement logic [11] only places objects randomly. Drawing inspiration from these established workflows, our research places a special emphasis on the sources of the implantation objects. Moreover, our research undertakes the challenge of working with a highly complex dataset, characterized by significant variance within a single object category.

Full 3D Rendering: This approach aims to render complete 3D scenes through the utilization of 3D assets and the modeling of entire 3D environments. Notably, it has been employed in studies [13] and has attracted attention from major companies like NVIDIA, signifying a current trend in the field [2]–[4]. While this method is capable of generating high-quality synthetic data, it is also the most challenging in terms of design and implementation. These efforts often involve complex procedures to simulate real-world environments and object interactions, facilitating the creation of diversified training data.

3D rendered objects into a 2D Image: This approach is a combination of the previous two. It involves the integration of 3D rendered objects into a real scene. The studies [14]–[16] try to enhance the realism of synthetic data by embedding rendered 3D objects seamlessly into 2D scenes, offering a middle ground between complexity and simplicity in the generation process. One potential issue in this method, however, is ensuring the availability of detailed 3D CAD models.

Generative Models: Recent advancements in generative AI, such as GANs or diffusion models, have introduced a novel approach to synthetic data generation. Models like DALL-E 4 by OpenAI [5], [6] or Stable Diffusion XL by Stability AI [7] can create images based on descriptions or natural language. However, these models still face challenges in consistently generating realistic images across all categories [8]. It's crucial to note that synthetic data produced by generative models might unintentionally replicate or intensify existing biases [9].

All of these approaches try to handle the reality gap differently. Some papers [10], [11], [15] try to narrow this gap by generating more realistic images. This process can be called system identification [13] which is the process of tuning the parameters to match the distribution of the real world. For example, this is done by placing objects with a realistic object arrangement into a scene or using blending techniques to reduce boundary artifacts. Additionally, the concept of domain adaptation is introduced, particularly through the use of Generative Adversarial Networks (GANs) for image enhancement. This further contributes to reducing the gap between synthetic and real-world data. The majority of the referenced papers attempt to use the aforementioned domain randomization. While most studies have focused on domain-specific data, restricting broader application, they collectively show that synthetic imagery can enhance model performance. Our research draws insights from these studies. Considering the challenges in our study, it is crucial to re-evaluate methods for optimal applicability.

3 DATASET

In this study, a domain-specific dataset is used to investigate the efficacy of synthetic data in enhancing object detection models. We focused testing the methods on a closed dataset of detecting milk and milk alternatives stored in tetrapaks and bottles within refrigerators. This specialized dataset encompassed a variety of scenarios, lighting conditions, orientations, and clutter levels commonly encountered within fridge interiors. This dataset is collected with smart refrigerators equipped with a camera system. We focused on the camera looking at the fridge door. Mentioned cameras produce high resolution images with a resolution of 1920x2560 pixels. It's important to note that our dataset is geographically limited to locations within Germany, and the image capture period spans a single year. Additionally, this dataset exhibited strong imbalances. Both the distribution of objects and the variety of refrigerator models present in the dataset were notably skewed. This imbalance is a natural consequence of the dataset's real-world origin. Notably, larger refrigerators are less prevalent, yet they present a more challenging detection environment due to their pronounced viewing angles. The milk training dataset comprises 1190 images, while an additional 425 images are reserved for validation purposes. For the test dataset, 1042 images are used. This dataset, in terms of diversity and size, is more limited compared to other object classification datasets like COCO [17]. Furthermore, the images in the test dataset were captured on distinct refrigerators, reducing the potential for knowledge transfer from the training dataset.

4 METHODOLOGY TO GENERATE SYNTHETIC IMAGES

Figure 1 provides a comprehensive overview of the key steps involved in the synthetic image generation process. Additionally, it contextualizes the generation of synthetic images within the broader framework of training and evaluation, offering a holistic perspective on the integration of synthetic data into the model development pipeline.

1. **Gathering of Implantation Objects:** The first step is to collect instances which can be implanted into a scene.
2. **Gathering of Implantation Backgrounds:** Scenes where these objects can be situated are gathered.
3. **Placement Logic:** With both objects and backgrounds at hand, decisions are made regarding the placement of objects within the scenes.
4. **Blending:** This step focuses on implanting an object into a specified location on a background.

4.1 Gathering of Implantation Objects

The acquisition of implantation objects, crucial for the synthetic data generation process, involves strategic choices to ensure both diversity and relevance. In this study, three distinctive approaches were explored. In the exploration of suitable implantation objects, we deliberately avoided the use of 3D rendered objects due to their inherent complexity.

Self augmentation:
The self augmentation methodology was applied to

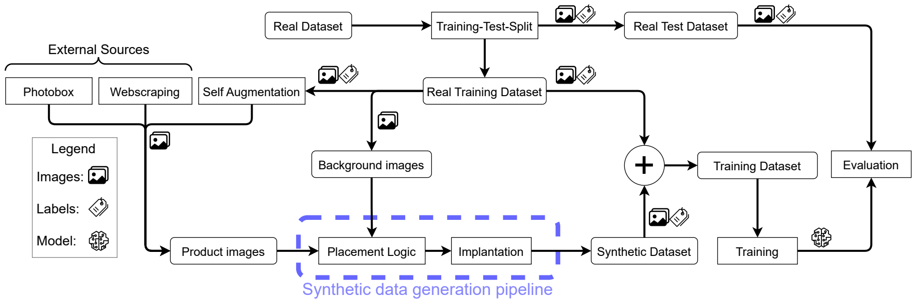


Figure 1: Illustration of the workflow as a flowchart.

extract objects directly from the pre-existing training dataset. For this approach, we used SegmentAnything [12] as a segmentation algorithm. The bounding box information from the training dataset was used to guide the segmentation process. This approach aligns with the statistical distribution of real-world objects in the dataset and introduces a unique augmentation technique. A noteworthy advantage of this technique lies in its efficiency, requiring relatively small manual intervention. SegmentAnything performs especially well on the milk dataset because its objects are relatively simple, making it a particularly effective approach in this context. An additional benefit of this technique is that it introduces significant variability into the dataset by placing objects randomly in new positions. In cases where the dataset contains a high percentage of incorrect or mislabeled objects, self augmentation may inadvertently reproduce these inaccuracies.

Web scraping:
By web scraping images from online retail platforms, it is possible to incorporate foreign objects from external sources to enhance the diversity of the dataset. Web scraping can be performed either manually or automatically using libraries like Beautiful Soup [18]. Some images may not even require cropping because they already have an alpha channel, which simplifies the incorporation of objects into our dataset. The scraping approach introduced a wider range of object variations, supplementing our dataset with new objects that might not have been adequately covered in the initial training data. An interesting aspect of this technique is its ability to easily add regularization objects. These are synthetic objects without labels placed into the scene alongside the labeled instances. This encourages the learning algorithm to focus on more than just the object boundaries when detecting objects. Furthermore, the inclusion of regularization objects can help mitigate issues related to false positive classifications. For example, in the milk dataset, juice tetrapaks are added as regularization objects. This methodology potentially expands the model's ability to recognize new objects that it has not been previously exposed to. This approach is considerably less labor-intensive compared to manual labeling. In the context of our study, we scraped webshops which sell groceries to extract product photos of fridge related objects.

360° Photobox:
In image creation, we utilize a 360° photobox featuring a rotating table. This setup captures object photographs from all perspectives, requiring some human effort. However, it provides a distinct advantage by offering multi-angle views of objects. Following a similar rationale, it is also possible to incorporate foreign objects from external sources that could potentially address gaps in our initial training data coverage. Furthermore, this approach grants us the ability to include specific objects that have historically performed poorly in our model. This is also the case for the inclusion of objects that may emerge in the future. This technique also has the ability to easily add regularization photos.

Figure 3 shows examples of all different object sources. In this study, we conducted a comparative analysis of the aforementioned data gathering techniques to offer guidance for future projects. Depending on the project's setting, one or a combination of these techniques can prove useful in enriching the dataset and improving the model's performance.

4.2 Gathering of Implantation Backgrounds

The implantation backgrounds serve as the canvas upon which objects are implanted. The selection and design of implantation backgrounds is important. The inclusion of a substantial number of backgrounds is a critical element for the domain randomization [13]. In the context of our milk dataset, the selection of implantation backgrounds consists of training images that depict empty fridges, or at the very least, fridges with some free space. The process of selecting these backgrounds can be executed manually or alternatively, by using a simple algorithm to automate the selection. The algorithm would analyze the presence or absence of labels, enabling it to identify which images depict empty spaces suitable for use as implantation backgrounds. An illustrative example image is provided in Figure

2. Furthermore, different refrigerator models are used. The use of different refrigerator models serves a dual purpose. It not only enhances background diversity, but also addresses potential imbalances in the dataset, ensuring a more representative and comprehensive training environment for our model. A crucial question to ask is how many implantation backgrounds should be used, in order to strike the right balance between diversity and the effort of collecting backgrounds.

4.3 Placement Logic

The strategic placement of the aforementioned implantation objects within scenes serves as the most important part of the system identification process, as it significantly contributes to the creation of more realistic images. In our comparison, we investigate two approaches: random placement and the proposed placement logic. While random placement lacks realism and coherence, the placement logic aims to emulate real-world spatial relationships and interactions, observed in reality. This involved implementing techniques that consider object size, occlusions, free space, foreground, and viewing angles, enhancing the verisimilitude of the generated data. For example, in the milk dataset, we put the objects where they would be in a real fridge, not just floating in the air. They go on the shelf like they do in a real fridge. It's important to note that certain aspects of this placement logic require manual labor for every background image. For example, defining a placement area, which dictates where objects are allowed to be positioned to ensure physically accurate placement. Additionally, for the reconstruction of foreground elements, the fridge holding bar needs to be accurately masked. One of these background images can be seen in the figure 2. A visual comparison between random placement and our placement logic can be seen in figure 3.

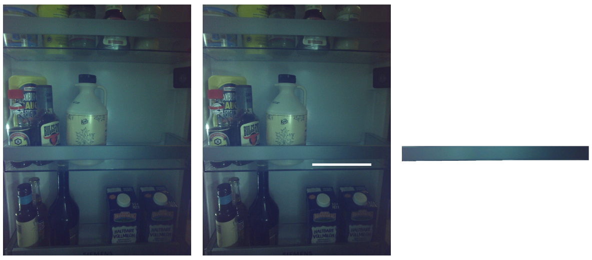

(a) (b) (c)

Figure 2: Example input background image to generate synthetic images: (a) Example background image; (b) with placement area mask: the white stripe in the middle row; (c) with foreground fridge holding bar.

4.4 Blending

The integration of synthetic objects into scenes involves a nuanced process of blending that directly impacts the visual cohesiveness and realism of the generated data. Blending can be crucial, especially because convolutional networks pay attention to edges and boundaries when recognizing objects. In this study, a list of blending techniques are evaluated. Beginning with a baseline "no blending" approach, where objects are inserted into scenes without any subsequent blending adjustments, often resulting in undesirable image artifacts. Subsequently, we tried seamless blending techniques, which prioritize the natural integration of synthetic objects into scenes. This process entails careful adjustment of object colors and lighting, creating a more visually coherent result. The application of seamless blending aims to mitigate the discernible boundaries between inserted objects and their surroundings, cultivating a more genuine representation that aligns with human visual perception [19]. Furthermore, our investigation explored the utilization of pyramid blending, a sophisticated technique that capitalizes on multi-resolution image processing to achieve a seamless fusion of objects into scenes. This approach involves creating hierarchical image pyramids that progressively refine object integration at different scales, resulting in a harmonious blend that accommodates diverse scene complexities and object scales [20], [21]. By systematically examining these diverse blending strategies, the study aims to find the most suitable blending technqiue. Figure 3 shows an image generated by the synthetic image pipeline.

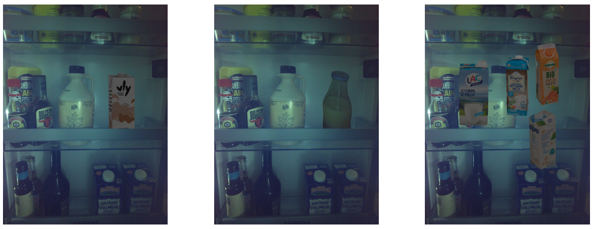

(a) (b) (c)

Figure 3: Examples of synthetic images: (a) Web-sourced tetrapak centrally placed using placement logic, seamless blending and reblending of the holding bar; (b) Self augmented milk bottle centrally placed with placement logic, seamless blending and reblending of the holding bar; (c) Photobox-sourced objects seamlessly blended centrally but lacking placement logic and holding bar reblending.

5 EXPERIMENTS AND RESULTS

For the experiments we used a YOLOv5 [22] model, a widely recognized platform well-suited for object detection tasks. To speed up the training process we used a model pretrained on the COCO dataset [17]. Employing early stopping, we ensured that the training process halts once the model's performance ceases to improve on the validation set.

To establish a robust baseline for our experimentation, the model underwent training on non-synthetic data across six distinct runs, each initiated with different seeds. To evaluate a synthetic data generation configuration, six different synthetic datasets were generated, each comprising approximately 1000 images. The repetition was necessary to account for the inherent randomness in the data generation process. This randomness includes factors like the choice of background, the selection and quantity of objects, and the positioning of these objects. Following the generation phase, these datasets were randomly combined with the real dataset. We conducted the following main experiment groups:

1. An explorative analysis of various object sources, incorporating different blending techniques, and evaluating the impact of placement logic. We tested sources like self augmentation, webscraping, photobox, and combinations of these to create a diverse object pool.

2. An investigation into the influence of the number of backgrounds used during image generation. We systematically increased the number of background images from 1 to 50.

3. An analysis of the number and ratio of synthetic images relative to the real dataset. We explored multiple configurations with artificially reduced real datasets and significantly increased synthetic images.

The outcomes of the experiments are systematically evaluated and compared based on their mean Average Precision (mAP) scores. The results are shown in figure 4 and table 1. A model trained solely on real world data exhibited an average mAP of 54.4, reflecting the challenging nature of the environment. The incorporation of synthetic data demonstrated, on average, a 2.32% increase in mAP. However, a deeper analysis is important to distinguish the specific synthetic data generation configurations that proved to be helpful and those that didn't.

5.1 Object source, Placement logic and Blending

Self augmentation: The use of self augmentation objects in synthetic datasets increased the mAP by an average of 1.0%. Incorporating a placement logic had a positive impact on mAP, while blending techniques unexpectedly seemed to lower the results. This highlights the importance of a good synthetic data generation, as incorrect methodologies can potentially degrade the model's performance. The best result is achieved by using a placement logic and simple stamping resulting in a 3.55% improvement.

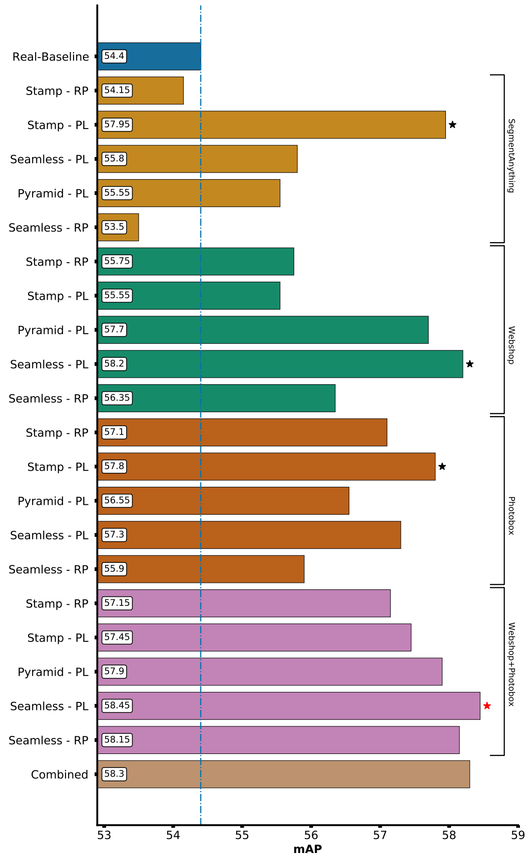


Figure 4: The bars illustrate the median mAP@50 of the experiment runs. The first bar represents the baseline performance achieved using only real data. The best overall result is achieved with a combination of Webscraping and Photobox as object sources together with seamless blending and a placement logic (marked with a red star). PL stands for placement logic. RP stands for random placement.

Webscraping: The use of webscraped object from webshop in synthetic datasets increased the mAP by an average of 2.3%. Including a placement logic enhanced the mAP. Blending techniques showed an overall improvement in results. The best result is achieved by using a placement logic and seamless blending, resulting in a 3.8% improvement.

360° Photobox: The use of the photobox objects in synthetic datasets increased the mAP by an average of 2.5%. On average, adding a placement logic improved the mAP. The impact of blending techniques on the result is uncertain and might even have a negative impact. The best result is achieved by using a placement logic and simple stamping, resulting in a 3.4% improvement.

Webscraping + 360° Photobox: Using the combined image pools of webscraped and photobox objects improved the mAP on average by 3.38%. The use of

	Self augmentation	Webscraping	360° Photobox	Webscraping + 360° Photobox
Stamp-RP	54.15	55.75	57.1	57.15
Stamp-PL	**57.95**	55.55	**57.8**	57.45
Seamless-PL	55.8	57.7	56.55	57.9
Pyramid-PL	55.55	**58.2**	57.3	**58.45**
Seamless-RP	53.5	56.35	55.9	58.15

Table 1: Experimental results as median mAP@50 scores. Baseline with solely real world training data is mAP@50 = 54.4. RP stands for random placement and PL stands for placement logic

blending techniques and placement logic improved the results. The best result is achieved by using a placement logic and seamless blending, resulting in a 4.05% improvement. This is also the best overall score.

Combined: Using the combined image pools of webscraping, photobox and self augmentation data improved the mAP by 3.9%. This dataset represents the combination of the best individual results among all object sources, where objects from self augmentation and the Photobox were implanted using stamping, and webscraped images from webshops were blended with seamless blending. The slightly lower performance compared to the combined webscraped and 360° Photobox category may be attributed to the potential negative influence of self augmentation on the overall results or just some degree of uncertainty.

Perspective Tranformation: We tried using perspective transformations on objects added to scenes, hoping to boost realism by aligning their orientation with the scene. However, despite extensive manual adjustments, this didn't lead to significant performance gains in our model, indicating that the practical benefits of these perspective adjustments might be minimal at the moment.

5.2 Number of Backgrounds

The quantity of backgrounds employed in synthetic data generation is a critical factor that contributes to enhancing the overall quality and effectiveness of the synthetic dataset. To investigate this, multiple datasets were generated with varying sizes of the available background image pool, using webscraped objects as the image pool due to being the best single-source approach. As illustrated in Figure 5, an increased number of backgrounds correlates positively with improved results. This aligns with the established concept of domain randomization found in prior research [13].

5.3 Number and Ratio of Synthetic Images

In our investigation, we explored the impact of the number and ratio of synthetic images within the training dataset. We systematically generated and incorporated synthetic images, varying both the quantity and the proportion in relation to the non-synthetic data. This allowed us to determine the optimal balance between real and synthetic data to achieve the best results and also shows how synthetic data could improve the results when real data is extremely in short supply. Our experiments involved training the model using solely 2000 synthetic images and gradually increasing the amount of real-world data from zero to 100, along with the addition of varying numbers of synthetic images (zero, 500, 1000, 1900). Subsequently, we increased the real-world data to 500, repeating the process of adding synthetic images (zero, 500, 1000, 1500). Finally, we utilized the original 1190-sized real-world dataset, incorporating different quantities of synthetic images (zero, 500, 1000, 2000, 4000). Throughout these experiments, we employed webscraped objects as the image pool, along with seamless blending and a placement logic, as this combination yielded the best results among single-source approaches. Refer to Figure 6 for a visual representation of our findings. Our experiments revealed a discernible pattern, unveiling a "sweet spot" in the quantity of synthetic data. Both insufficient and excessive infusion of synthetic data were identified as detrimental factors impacting the final results. So, while 1000 was the ideal number in our case, this number might not be the same across different projects and should be considered as an important parameter for a hyperparameter optimization. Furthermore, our analysis highlighted the significance of an increased number of real-world data, showcasing a strong positive correlation with model performance. However, we observed that the improvements with increasing amounts of only real data tended to stagnate. We observed that synthetic

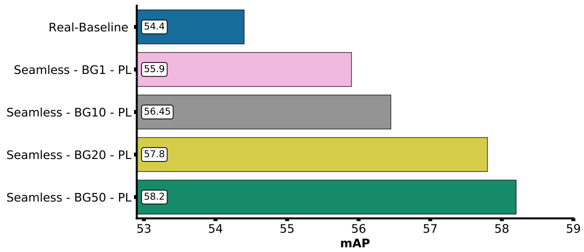


Figure 5: The bars illustrate the median mAP@50 of the experiment runs comparing different amounts of backgrounds used for the data generation. Webscraped images are used as object source. PL stands for placement logic.

data is particularly valuable in scenarios where a shortage of real data is encountered. It's important to note that our experiments demonstrated that relying solely on synthetic data proved to be insufficient.

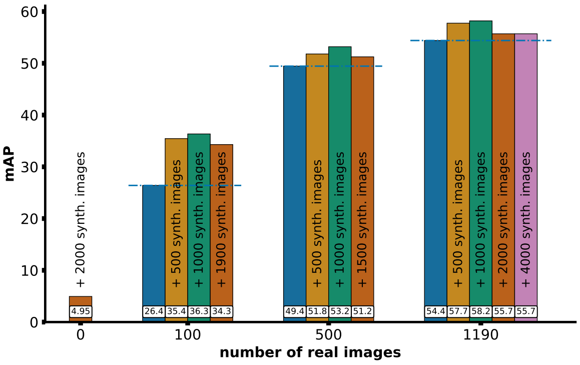


Figure 6: The bars illustrate the median mAP@50 of the experiment runs comparing different ratios between real or synthetic images. The blue bars are models trained on only real world data.

5.4 Bias in the Dataset

The synthetic data generation process notably reduced the bias in the dataset, contributing to a more representative distribution of fridge types. Consequently, the most significant mAP increase was observed in the case of larger fridge types, which were initially underrepresented in the baseline dataset. This resulted in an average mAP increase of 3.87% on larger fridges.

6 CONCLUSION

This study explores different ways to improve existing datasets through the integration of synthetic data. Our approach encompasses the incorporation of internal objects, harnessed via self augmentation, and external objects obtained through two distinct avenues: web scraping and the photobox method. We showed that all investigated object sources are useful for the synthetic data generation. Leveraging self augmentation showcased the smallest improvement that was still significant. This is a good sign because this kind of data can easily be generated. Furthermore, the incorporation of objects from the photobox and web scraping exhibited even more substantial enhancements. While web scraping proves to be a less time-consuming technique, its applicability may be limited in certain scenarios, making the photobox method a valuable alternative, especially for scenarios where manual inclusion of specific objects is essential. The integration of a placement logic proves to be a significant contributor to substantial improvements across various cases. Exploring blending techniques, while not universally applicable, showed potential in enhancing results. It's important to note that the success of blending depends on the image source. For example, self augmentation data benefits from simple stamping, while web scraped images benefit from better blending. The study demonstrated that synthetic data does not need to be ultra-realistic to deceive object detection algorithms. Synthetic data, derived from both internal and external sources, successfully addressed imbalances in the dataset, particularly in scenarios involving larger fridge types. This harmonization of the dataset led to a significant mAP boost on larger fridges, affirming the effectiveness of synthetic data in bridging gaps in real-world dataset disparities. The most notable performance improvements of 9.9 mAP@50 were observed when the training dataset was extremely limited. Moreover, this study corroborated the findings of previous works [10], [11] utilizing an updated object detection network. In conclusion, our approach not only underscores the significance of diverse object sources but also highlights the utilization of placement logic and blending techniques, collectively contributing to a better dataset.

7 FUTURE WORK

A promising direction for further exploration involves the evolution of a dynamic placement logic. Such an adaptive system would intelligently respond to different scenes, reducing the dependency on manual labor. Exploring image enhancement is another compelling direction. The integration of advanced techniques, such as CycleGANs [23], could improve the realism of generated images. CycleGANs, by learning the translation of images between domains, offer a sophisticated means to bridge the gap between synthetic and real data. While our initial experiments with a Masked-CycleGAN have shown promise, a more comprehensive evaluation is required.

Acknowledgments: We are thankful to Conrad Smith for his proofreading assistance and anonymous reviewers for their suggestion

DECLARATIONS

- **Authors' Contributions** Eduard Bartolovic led the majority of the work. Tobias Höfer, Clemens Hage, and Alfred Nischwitz contributed equally. All authors have approved the final manuscript.
- **Funding** The BSH-Group granted computational resources and data access.
- **Conflicts of interest:** Clemens Hage is employee of the BSH group. The other authors declare no conflicts of interest.
- **Availability of Data:** Due to privacy considerations, we are unable to share the data.
- **Code Availability:** The source code is available at https://github.com/EduardBartolovic/synthetic-data-generation.

REFERENCES

[1] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing*, vol. 126, p. 103 514, 2022, ISSN: 1051-2004. DOI: `https://doi.org/10.1016/j.dsp.2022.103514`.

[2] A. Prakash, S. Boochoon, M. Brophy, *et al.*, "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7249–7255. DOI: `10.1109/ICRA.2019.8794443`.

[3] NVIDIA, *Nvidia omniverse*. [Online]. Available: `https://developer.nvidia.com/blog/tag/omniverse/`.

[4] S. Iqbal, J. Tremblay, A. Campbell, *et al.*, "Toward sim-to-real directional semantic grasping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 7247–7253. DOI: `10.1109/ICRA40945.2020.9197310`.

[5] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, *Hierarchical text-conditional image generation with clip latents*, 2022. DOI: `https://doi.org/10.48550/arXiv.2204.06125`. arXiv: `2204.06125 [cs.CV]`.

[6] J. Betker, G. Goh, L. Jing, *et al.*, "Improving image generation with better captions." [Online]. Available: `https://api.semanticscholar.org/CorpusID:264403242`.

[7] D. Podell, Z. English, K. Lacey, *et al.*, *Sdxl: Improving latent diffusion models for high-resolution image synthesis*, 2023. DOI: `https://doi.org/10.48550/arXiv.2307.01952`. arXiv: `2307.01952 [cs.CV]`.

[8] A. Stöckl, "Evaluating a synthetic image dataset generated with stable diffusion," in *Proceedings of Eighth International Congress on Information and Communication Technology*, X.-S. Yang, R. S. Sherratt, N. Dey, and A. Joshi, Eds., Singapore: Springer Nature Singapore, 2023, pp. 805–818, ISBN: 978-981-99-3243-6.

[9] A. Jadon and S. Kumar, "Leveraging generative ai models for synthetic data generation in healthcare: Balancing research and privacy," in *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 2023, pp. 1–4. DOI: `10.1109/SmartNets58706.2023.10215825`.

[10] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka, "Synthesizing training data for object detection in indoor scenes," *CoRR*, vol. abs/1702.07836, 2017. DOI: `10.1109/ICCV.2017.146`. arXiv: `1702.07836`. [Online]. Available: `http://arxiv.org/abs/1702.07836`.

[11] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1310–1319. DOI: `10.1109/ICCV.2017.146`.

[12] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023, pp. 4015–4026. DOI: `DOI:10.48550/arXiv.2304.02643`.

[13] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. 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)*, 2017, pp. 23–30. DOI: `10.1109/IROS.2017.8202133`.

[14] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2686–2694. DOI: `10.1109/ICCV.2015.308`.

[15] W. Liu, J. Liu, and B. Luo, "Can synthetic data improve object detection results for remote sensing images?" *CoRR*, vol. abs/2006.05015, 2020. arXiv: `2006.05015`. [Online]. Available: `https://arxiv.org/abs/2006.05015`.

[16] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, "On pre-trained image features and synthetic images for deep learning," in *Computer Vision - ECCV 2018 Workshops: Munich, Germany, September 8-14, 2018, Proceedings, Part I*, Munich, Germany: Springer-Verlag, 2019, pp. 682–697, ISBN: 978-3-030-11008-6. DOI: `10.1007/978-3-030-11009-3_42`. [Online]. Available: `https://doi.org/10.1007/978-3-030-11009-3_42`.

[17] T. Lin, M. Maire, S. J. Belongie, *et al.*, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. DOI: `https://doi.org/10.1007/978-3-319-10602-1_48`. arXiv: `1405.0312`.

[18] L. Richardson, *Beautiful soup documentation*, 2007.

[19] P. Perez, M. Gangnet, and A. Blake, "Poisson image editing," SIGGRAPH 03, pp. 313–318, 2003. DOI: 10.1145/1201775.882269. [Online]. Available: https://doi.org/10.1145/1201775.882269.

[20] A. Nischwitz, M. Fischer, P. Haberaecker, and G. Socher, *Bildverarbeitung: Band II des Standardwerks Computergrafik und Bildverarbeitung*. Boston: Springer Vieweg, 2020, ISBN: 978-3-658-28704-7.

[21] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983. DOI: 10.1109/TCOM.1983.1095851.

[22] G. Jocher, *Ultralytics/yolov5: V7.0 - yolov5 sota realtime instance segmentations*, Zenodo, Nov. 2022. DOI: 10.5281/zenodo.7347926..

[23] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

ECNNXAI: Ensembled CNNs with eXplainable Artificial Intelligence for Colon Histopathology Image Classification

Juwaria Qadri
Department of Computer Science
Birla Institute of Technology and Science
Pilani, Dubai Campus
Dubai, United Arab Emirates
f20200043@dubai.bits-pilani.ac.in

J. Angel Arul Jothi
Department of Computer Science
Birla Institute of Technology and Science
Pilani, Dubai Campus
Dubai, United Arab Emirates
angeljothi@dubai.bits-pilani.ac.in

ABSTRACT

Colon cancer is ranked as the third most commonly diagnosed cancer and second for causing the most cancer related deaths. Histopathology is a crucial diagnostic tool for cancer since it enables the microscopic analysis of tissue samples to pinpoint abnormal cells, to identify the stage of the cancer and its kind. There is a significant need for precise detection and diagnosis from histopathology images. This research proposes a stacking ensemble model called Ensembled Convolutional Neural Networks with eXplainable Artificial Intelligence (ECNNXAI) for multiclass colon histopathology image classification. Our ensemble model consists of three pre-trained convolutional neural networks (XceptionNet, DenseNet-121 and InceptionNetV3) as base classifiers and the logistic regression as the meta classifier. Gradient-weighted Class Activation Mapping (Grad-CAM) visualization technique is used to interpret and understand the regions focused by the base classifiers to arrive at the final predictions. SHapley Additive exPlanations (SHAP) is used for understanding the predictions made by the ECNNXAI. The proposed model achieves the best overall performance with accuracy of 72.83%, precision of 77.78%, recall of 66.52% and F1 score of 71.71% on the Chaoyang dataset.

Keywords

Histopathology, colon cancer, stacking ensemble, explainable artificial intelligence, Grad-CAM, SHAP.

1 INTRODUCTION

Colon cancer is a malignant growth in the colon or rectum. Globally, colon cancer is ranked as the third most diagnosed cancer and the second leading cause of cancer-related deaths by the World Health Organization, with over 1.9 million new cases and nearly 935,000 deaths annually [WHO23]. Age, family history, genetic conditions and unhealthy lifestyle choices are a few factors that could raise the risk of colorectal cancer. Physical examination, magnetic resonance imaging, abdominal ultrasound, colonoscopy, tissue sample collection for histopathology examination are a few methods to detect colon cancer. Timely detection enhances treatment success, lowers mortality, and enables less invasive interventions, improving patient outcomes.

Accurate histopathology examination is vital in diagnosing cancer. Usually pathologists use a microscope to examine tissues and cells to make diagnoses. It helps identify tumor type, stage, and grade, guiding treatment and predicting outcomes. However, manually analyzing complex-natured histopathology images takes a lot of time, is laborious, challenging and may be error prone [Ham20]. Additionally, there is some subjectivity in the criteria used by different pathologists to identify and classify these images. This subjectivity makes it possible for different pathologists to interpret the same set of images differently. Computer assisted diagnosis can play a significant role in assisting pathologists in examining histopathology images.

Digital pathology (DP) [Sen22] is the process of digitizing histopathology slides to produce high resolution images. Due to the advent of DP, computer assisted diagnosis systems employing deep learning models are recently used to segment images, identify objects and detect diseases from histopathology images. Convolutional Neural Networks (CNNs), are deep learning models that are used in numerous computer vision (CV) applications, specifically those which process image data. They exhibit remarkable automatic feature extraction ability requiring minimal pre-processing. CNNs are a popular choice for medical image 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.

https://www.doi.org/10.24132/CSRN.3401.29

applications such as segmentation, classification and anomaly detection.

Stacking ensemble is a technique used to build a strong classification model by combining multiple individual classifiers called base models. Predictions are obtained by training various base models and then a meta-model is constructed to produce the final output. In a stacked ensemble technique, the meta model is fed with the predictions of the base models from the preceding level [Pav18]. Using an ensemble improves accuracy and reduces generalization error limiting the impact of error causing factors like noise, bias and variance.

It is crucial to understand how a model arrives at its final prediction especially in domains like healthcare. Machine learning models, especially deep architectures, are regarded as black boxes because the way in which these models arrive at their final predictions is not explicit. This is attributed to the complex architecture of these models and the difficulty in understanding their internal working. EXplainable Artificial Intelligence (XAI) is a field of research that adds transparency to the working of the models by explaining, visualizing and interpreting the results. Today, image visualization techniques like Gradient weighted Class Activation Mapping (Grad-CAM) [Sel17] and SHapley Additive exPlanations (SHAP) [Lun17] are used to generate visual explanations for any CNN-based model.

In this study, a novel model called Ensembled CNNs with eXplainable Artificial Intelligence (ECNNXAI) is proposed, which is able to accurately classify colon histopathology images as well as visually explain decisions taken at every level of the stacked ensemble. ECNNXAI contains three pre-trained CNN models as its base classifier and a machine learning model as the meta classifier. Explainable AI (XAI) techniques like Grad-CAM and SHAP are used to provide model interpretability. The following are the contributions of this paper: (1) This study performs multiclass image classification on colon histopathology images. (2) A novel approach called ECNNXAI is put forth with the aim to enhance overall classification accuracy by using stacked ensemble techniques to combine three individually trained CNN models. (3) Visual explanations are produced using XAI techniques like Grad-CAM and SHAP.

The structure of this paper is as follows: Section 2 elaborates on the previous works, Section 3 describes the dataset. Section 4 discusses our proposed model. Section 5 provides the implementation details and evaluation metrics. Section 6 describes the results and discussions. Section 7 provides conclusions and suggests future works.

2 LITERATURE REVIEW

This section elaborates the previous work done on colon histopathology images using various techniques. Zhu et al. [Zhu21] developed an Easy/Hard/Noisy (EHN) image detection model accompanied by a CNN classification model. The EHN model utilized the sample training history to separate the useful hard samples from the detrimental noisy data. It was then incorporated into a self-training algorithm to gradually correct label errors and reduce noise rate. A Noise Suppressing and Hard Enhancing (NSHE) strategy was also suggested to train the noise robust model using the generated almost clean dataset. ResNet-32 was used as the backbone of the classification model for the Chaoyang dataset. Kadian et al. [Kad23] used the pipeline model proposed in [Zhu21] and replaced the backbones with different models like ResNet-34, Cross-Covariance Image Transformer (XCiT), SqueezeNet, and MobileNet. These models were integrated individually with this architecture that incorporated data cleaning. A two-phased architecture was utilized where Phase I generated a dataset that was almost clean through label correction, and the Phase 2 utilized the dataset generated for obtaining a classification model that was robust. It was found that the MobileNet model performed the best on the Chaoyang dataset.

Tepe and Bilgin [Tep22] used Graph Neural Networks (GNNs) to classify the tissue types from the Chaoyang dataset. The construction of a super-pixel graph from an image was the first step in this process that was followed by the application of the GNN models to the constructed graph. The study experimented with the Graph Convolution Network (GCN), Graph Isomorphism Network (GIN), and the Graph Attention Network (GAT) models. Out of the models experimented, the GIN model performed the best.

For computer vision related tasks, the Vision Transformer (ViT) is becoming popular, however pure ViT models do not work well on small datasets. The work by Li et al. [Li22] suggested locality guidance for enhancing the performance of ViT on small datasets. This approach involved using a lightweight ResNet-56 that was trained on the exact same dataset on which the ViT was trained. The local information extracted by the CNN was then combined with the global information extracted from the ViT. This approach enabled the ViT to learn and use both local and global information for the classification of the Chaoyang dataset.

Three primary architectures were used by Nergiz [Ner22] to benchmark the ResNet-18 model for Chaoyang dataset classification: Single Learning (SL), Centralized Learning (CL), and Federated Learning (FL). The traditional FL failed to converge the models on a highly biased dataset to produce good results. As a result, a brand-new Federated Neural Style Transfer

(FNST) technique was put forth that federated the traditional Neural Style Transfer (NST) algorithm and generated synthetic images. The ResNet-18 model was used to test the SL, CL, and FL architectures. The synthetic images produced by the proposed FNST method were also utilized to compare with pure FL findings. The results demonstrated that medical institutions, particularly those that specialize in treating uncommon diseases or medical problems, can effectively apply the FNST algorithm .

Zeid et al. [Zei21] proposed the Compact Convolutional Transformers (CCT) model for ColoRectal Cancer (CRC) tissue classification. The CCT used a convolutional based patching technique that preserves local information and was capable of encoding relationships between the patches. The images were passed to a convolutional layer before passing it to the transformer encoder. This way the CCT was able to combine the advantages of both CNN and transformers. The CCT model outperformed the ViT model. Albashish [Alb22] proposed an ensemble model that was built on four pre-trained CNN models namely DenseNet-121, MobileNetV2, InceptionV3 and the VGG-16 model for classifying colon histopathology images. A blockwise fine-tuning approach was used and additional drop out and dense layers were incorporated to improve the colon image analysis. The ensemble learning methods used were majority voting and product rule. The model that used the product rule achieved better performance as compared to the model that used majority voting.

3 DATASET DESCRIPTION

The Chaoyang dataset introduced by Zhu et al. [Zhu21] is a publicly available dataset which contains images of colon slides scanned at ×20 magnification and collected from the Chaoyang hospital. The images are of size 512×512 and in jpg format. All images in the dataset belong to one of the four categories namely normal, serrated, adenocarcinoma and adenoma. Figure 1 shows images belonging to each of the four classes. Three qualified pathologists collectively assigned labels to the images. The images on which all the three pathologists unanimously agreed on are added to the test set. Rest of the images are added to the training set with labels suggested by one of the pathologists randomly selected. The achieved training set consists of 4021 images and the test set consists of 2139 images in total. Figure 2 depicts the distribution of the four classes in both the training and test dataset.

4 METHODOLOGY

The proposed ECNNXAI is shown in Fig.3. Initially, the images from the dataset are pre-processed. The preprocessed images are then fed to the stacked ensemble model consisting of base and meta classifiers. The XceptionNet, InceptionV3 and DenseNet-121 models are used as the base classifiers and logistic regression model is used as the meta classifier. The meta classifier generates the final predicted class label. By combining predictions from multiple different base models, the strengths of one model can compensate for the weaknesses of another, resulting in a more robust and less biased overall prediction. Grad-CAM is used for visualizing and interpreting the predictions made by the base classifiers. SHAP offers visual explanations for the ensemble model and helps to understand how each base classifier influences the final label predicted.

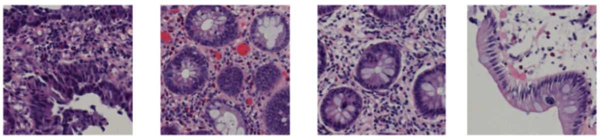

(a) (b) (c) (d)

Figure 1: Sample images from the dataset. (a) Adenocarcinoma; (b) Adenoma; (c) Normal; (d) Serrated

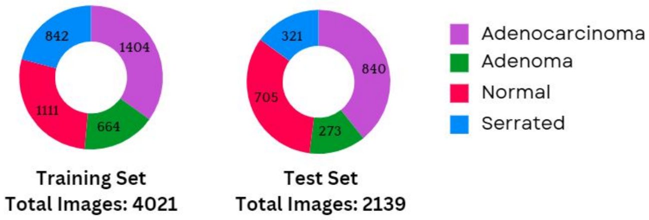


Figure 2: Dataset Distribution

4.1 Data Pre-processing

The training dataset exhibits a visible class imbalance as seen in Fig.2 which can affect the performance of the model. Hence, in order to prevent the model from overfitting and improve its generalization capability the training dataset is enhanced by augmentation techniques such as right-angled rotations and vertical and horizontal flip. These methods are applied to generate new samples for all classes and to increase the number of samples in each class to 1500. All images are then resized to 256×256. Unique numerical identifiers from 0 to 3 are assigned to the class labels adenocarcinoma, adenoma, normal and serrated respectively.

4.2 Base Classifiers

This work uses three pre-trained CNNs namely XceptionNet [Sze16], InceptionV3 [Cho17] and DenseNet-121 [Hua17] as the base classifiers. Pre-trained CNNs are models that are designed and trained for one purpose but can be retrained with little effort on another dataset for a closely related task. CNNs are made by stacking convolution layer, pooling layer, flatten layer, fully connected layer and an output layer. A CNN can

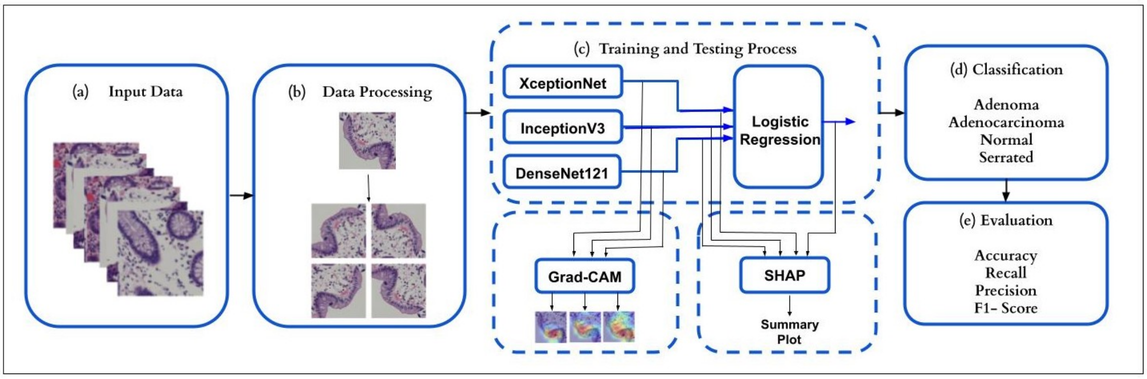


Figure 3: Structure of ECNNXAI

have multiple convolution and pooling layers making the architecture deep. The convolution layers are the key feature extractors of a CNN. The pooling layers are tasked to reduce the number of parameters thereby reducing complexity and improving efficiency. The output of the final pooling layer will be a set of 2-D feature maps. The flatten layer converts the 2-D feature maps into a 1-D feature vector. The fully connected layer is the classification layer that classifies the input image based on the features extracted from the previous layers. The arrangement of the nodes in a fully connected layer is such that all nodes of one layer are connected to each node of the next layer. Finally, the output layer provides the results of classification. The base classifiers in this work are chosen experimentally as explained in Section 6.1.

4.2.1 InceptionV3

Designed by Google, InceptionV3 [Cho17] is a CNN particularly developed for classification of images, and is part of the Inception family. It is 48 layers deep and consists of 23,626,728 parameters. It consists of multiple layers of convolution having various sizes of filters (1×1, 3×3, 5×5) as well as pooling layers for extracting the hierarchical features of the input image. Various techniques are utilized in the InceptionV3 for improving accuracy and efficiency. It enables feature capture at multiple scales using its parallel convolutions having different sizes of filters. The use of 1×1 convolutions for factoring larger convolutions help reduce the cost of computation. Factorized convolutions as well as parallel operations help maintain optimal balance between different cell regions in the colonoscopy images.

4.2.2 XceptionNet

XceptionNet [Sze16], also developed by Google, is an extension of the Inception architecture leveraging separable depth-wise convolutions for performance and computation enhancement. A series of normal convolution layers, depth-wise separable convolutions, pooling layers and residual connections make up its 71 layers deep architecture. Depth-wise separable convolutions enable factorizing of the standard convolution to two separate operations, that are the depth-wise convolutions and pointwise convolutions. The number of parameters are reduced through this separation as well as there is reduction in the complexity of computation, along with allowing more learning of the discriminative features. Despite having almost equal number of parameters as of InceptionV3, the XceptionNet architecture is able to perform better because of the capacity increase due the effective use of the parameters of the model.

4.2.3 DenseNet-121

Densely connected convolutional networks, or DenseNet [Hua17] for short, presents a novel design with the motive of enhancing feature propagation, encouraging feature reuse, and overcoming the vanishing-gradient issue. It is a deep pre-trained CNN model consisting of 121 layers in totals. A unique feature of DenseNet is its dense connectivity where every layer transfers its feature maps to every layer that comes after it and receives feature maps from every layer that comes before it. The network is also made up of numerous layers that are tightly connected within each of its dense blocks. Feature maps from earlier layers are concatenated within a dense block and then passed on to later layers. Reusing features helps improve gradient flow, promote feature propagation, and facilitates the acquisition of more discriminating features. The DenseNet uses batch normalization and ReLU activations as well. Finally, a global average pooling layer is employed followed by a fully connected layer for classification.

4.3 Meta Classifier

The logistic regression (LR) model [Pen02] is used as the meta classifier. It uses a statistical approach. It models the probability of a given input belonging to a certain class through the use of the sigmoid (logistic) function. Logistic regression algorithm is employed often for data having linear relationship between features and the target variable or for datasets that are relatively simple. Results obtained through logistic regression are interpretable, enabling the understanding of the influence of the individual feature on the final prediction. Low cost of computation of logistic regression makes it suitable for cases where the datasets are large and computational resources are few in comparison to models that are complex.

4.4 Explainability

In this work, explainability is incorporated for understanding the working of the base classifiers and the meta classifier using Grad-CAM and SHAP.

4.4.1 Gradient-weighted Class Activation Mapping (Grad-CAM) for base classifiers

The decisions of deep CNNs for the tasks of image classification are visualized and understood through the Grad-CAM. Grad-CAM helps increase the transparency of the model by indicating the features of the image that are most crucial for determining the class label by the model. It works as follows: The input image is passed through the CNN and the target class score gradients are computed using the feature maps of the final convolution layer. Rich spatial information is lost in the fully connected layer, therefore the final convolution layer offers high-level semantics and detailed spatial information. Global average pooling of the gradients is performed to identify the importance of every feature map based on the target class. The result of the pooling operation is a heat map that shows the significance of different regions in the input image for the predicted class. The heat map generated is overlapped onto the original input image to indicate the areas focused by the network while the particular class predictions are made. High intensity regions in the heat map indicate greater significance for the selection of the target class than the lower intensity ones. Fig.4. illustrates the working of Grad-CAM.

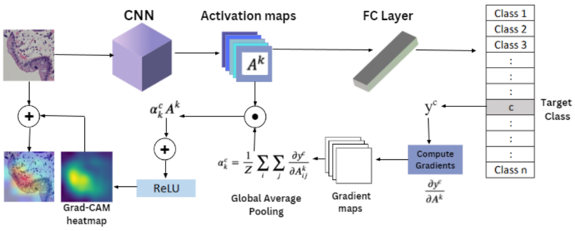


Figure 4: Working of Grad-CAM

4.4.2 SHAP for the level 1 classifiers

Ensemble models are more complex and can be challenging to interpret. SHAP employs cooperative game theory to give each attribute a value that represents how much of an impact it has on the final output. In this work, this property is used to provide explanations on how the contributions from the individual classifiers influence the overall result. This aids in determining the relative significance of every base model in the ensemble.

Following are the steps to calculate the shapley values for the base classifiers in the ensemble: (1) Initially, choose one base classifier as the classifier of interest (*CoI*). (2) List all possible classifier combinations excluding the *CoI*. (3) Compute the ensemble's prediction both with and without the *CoI* for each subset. (4) Determine the marginal contribution (*MC*) of the *CoI* to the final prediction by subtracting the above two values. (5) Compute the average *MC* of the *CoI* (*AvgMC_CoI*) to all possible subsets as the mean of the *MC*s of the *CoI* over all possible subsets. The above steps 1-5 are repeated for each base classifier in the ensemble. The *AvgMC_CoI* is given by Eq. 1 where N is the set of all base classifiers, s is the subset of base classifiers that does not include the *CoI*, $f(s)$ is the ensemble prediction for the subset s, $f(s \cup i)$ is the ensemble prediction for the subset s plus *CoI* and $|\cdot|$ denotes the number of elements in a set.

$$AvgMC\_CoI = \sum_{s \subseteq N|i} \frac{|s|!(|N|-|s|-1)!}{|N|!}[f(s \cup i) - f(s)] \tag{1}$$

4.5 Training the ECNNXAI

In this work, the pre-trained CNNs are accessed from the applications module of the Keras library. The top layers of the CNN models are set to false to build our own classification block. This classification block is common for all the models and it consists of a flatten layer followed by two fully connected layers having 512 and 256 neurons respectively and ReLU activation function. This is then followed by an output layer which consists of 4 neurons representing the four target classes in the colon histopathology dataset with softmax activation function.

The training set is further split in the ratio of 80:20 into the train and validation sets. All the base classifiers are trained using the following hyperparameters: 100 epochs, batch size of 64, Adam optimizer and learning rate of 0.001. A stacking classifier is then created using StackingClasssifier from the sklearn.ensemble module with the InceptionV3, XceptionNet and DenseNet-121 as the base classifiers and Logistic regression as the meta classifier.

In this work, pre-processed images from the training set are the inputs for the base classifier's training. Each base classifier predicts the class label of the input images in the training dataset. The final predictions of the base classifiers are formulated into a meta dataset having 4800 rows, 3 independent features (predictions of the base classifiers), and one target attribute. The actual class label of the input images becomes the target attribute. The LR model is then trained on this data. After training the LR model, the test set of the Chaoyang dataset is fed to the ECNNXAI. The images pass through the three base classifiers which provide their predictions. Finally, the predictions from the base classifiers are fed to the LR model which provides the final predictions by combining the predictions of the base models. This helps improve generalization capacity and prediction performance.

5 IMPLEMENTATION AND EVALUATION

This work is implemented in the Jupyter Notebook environment using Python with Tensorflow version 2.1.0. and Keras version 2.3.1. A V100-PCIE-32GB GPU with Ubuntu operating system was used. Let true positives (TP) be the correct predictions of a class of interest, true negatives (TN) denote the correct negative predictions with respect to a class of interest, false positives (FP) denote the number of predictions where samples of other classes are incorrectly predicted as a class of interest, and false negatives (FN) denote the number of predictions where samples of a class of interest are incorrectly predicted as belonging to other classes. All models are evaluated using four evaluation metrics namely accuracy, recall, precision and F1 score as presented in Table 1.

Metric	**Formula**
Accuracy	(TP+TN)/(TP+FP+TN+FN)
Recall	TP/(TP+FN)
Precision	TP/(TP+FP)
F1 score	(2×Precision×Recall)/(Precision+Recall)

Table 1: Evaluation metrics

6 RESULTS AND DISCUSSIONS

6.1 Experiments for choosing the base classifiers

To choose the base classifiers for the ensemble model, six pre-trained models like VGG-16 [Sim14], ResNet-50 [He16], EfficientNetb0 [Tan19], InceptionV3 [Cho17], XceptionNet [Sze16] and DenseNet-121 [Hua17] are trained and tested using the same hyperparameters (epochs: 100, optimizer: Adam, learning rate: 0.001, batch size: 64) with the Chaoyang dataset. According to Table 2, the DenseNet-121 is the best performing pre-trained model achieving an accuracy of 70.1%, precision of 75.67%, recall of 62.97% and F1 score of 68.74%. The XceptionNet is the second best performing model that achieves an accuracy of 68.56%, precision of 73.87%, recall of 59.89% and F1 score of 66.15%. The InceptionV3 is the third best performing model with an accuracy of 67.25%, precision of 72.86%, recall of 55.64% and F1 score of 63.09%. Thus, the three best performing models: XceptionNet, InceptionV3 and DenseNet-121 are then selected to be the base classifiers for our ECNNXAI.

6.2 Ablation study

Experiments are conducted in order to understand the importance of all base classifiers in the ECNNXAI. This is done by forming all possible 2-subsets of the base classifiers and comparing their results against the ECNNXAI which is a 3 classifier combination. The different base classifier combinations that we experimented are: XceptionNet + InceptionV3 + LR (X+I+LR), XceptionNet + DenseNet-121 + LR (X+D+LR), DenseNet-121 + InceptionV3 + LR (D+I+LR). As seen from Table 3 and Table 4, the ECNNXAI outperformed all base classifier combinations.

In order to assess the importance of the logistic regression meta classifier of the ECNNXAI, it is swapped with other popular machine learning algorithms like the naive-Bayes [Wic21] and the decision tree [Cos23] classifiers to create ensemble-NB and ensemble-DT respectively. As seen from Table 4, the ECNNXAI outperformed the ensemble-NB and ensemble-DT ensemble models.

6.3 Performance of the proposed model

It could be inferred from Table 2, Table 3 and Table 4 that the ECNNXAI comprising of a logistic regression model stacked on top of the InceptionV3, XceptionNet and DenseNet-121 outperformed the individual CNNs and all other combinations of the base classifiers. It could be demonstrated from the results that utilizing different base classifiers with varying strengths and weaknesses helped build a stronger classifier model. This is because the different base classifiers perform well in certain areas of the feature space. The overall robustness of ECNNXAI is enhanced and a wider range of patterns are captured helping it to achieve better performance with an accuracy of 72.83%, precision of 77.78%, recall of 66.52% and F1 score of 71.71%.

6.4 Evaluation of explainability

In this work, the use of Grad-CAM highlights how specific regions from the colonoscopy images are used by the base classifiers in order to arrive at the final predictions thereby enhancing the interpretability. This is crucial to understand the classification decisions made by the base classifiers and enables specialists in the field to understand areas focused by the network for predictions. Figure 5 displays the specific regions in sample images belonging to different classes that contribute to the final decisions made by the base classifiers.

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 score (%)**
Resnet-50	61.87	63.92	52.3	57.52
VGG-16	62.74	66.42	51.90	58.26
EfficientNetb0	65.21	68.28	53.77	60.16
InceptionV3	67.25	72.86	55.64	63.09
XceptionNet	68.56	73.87	59.89	66.15
DenseNet-121	**70.1**	**75.67**	**62.97**	**68.74**

Table 2: Experimental results for choosing the base classifiers

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 score (%)**
X+I+LR	63.19	68.13	50.33	57.81
D+I+LR	63.85	70.49	**53.05**	**60.53**
X+D+LR	**64.24**	**71.43**	51.85	60.08

Table 3: Experimental results for evaluating the importance of the combinations of the base classifiers. X denotes XceptionNet, I denotes InceptionV3, D denotes DenseNet-121 and LR denotes Logistic Regression.

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 score (%)**
Ensemble-DT	70.79	76.88	62.91	69.2
Ensemble-NB	72.04	77.73	64.17	69.95
ECNNXAI	**72.83**	**77.78**	**66.52**	**71.71**

Table 4: Experimental results to assess the importance of logistic regression (meta classifier)

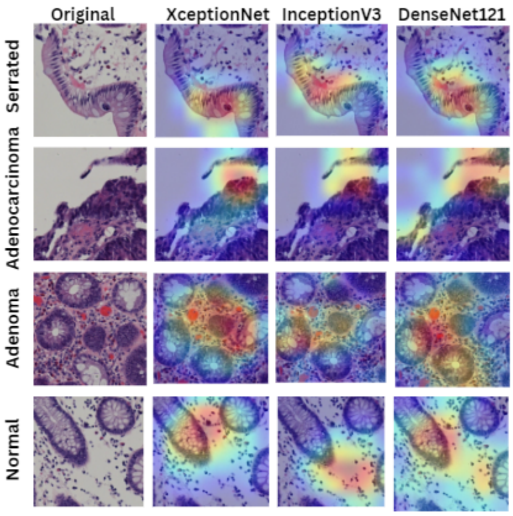


Figure 5: Grad-CAM visualization of sample images belonging to different classes and the regions focused by the base classifiers

Summary plot of SHAP offers a clear and understandable illustration of how each individual base classifier influences the output of the ECNNXAI. It is a horizontal bar chart where every bar represents a base classifier. The impact of each base classifier on the output of the model is shown by the length of the bar. Longer bar in SHAP summary plot indicate that the impact of a classifier is more on the final output generated while shorter bar indicate that the impact of the classifier is lesser. The base classifiers are arranged from highest impact to lowest impact according to how important they are for the model while making the final decision.

Figure 6 illustrates that the DenseNet-121 model has the highest impact on the final prediction of the ECNNXAI, which is followed by the XceptionNet and InceptionV3. From Fig. 6, it is observed that the DenseNet-121 has the highest influence on the predictions of ECNNXAI for class 1 (adenoma) and class 2 (normal) and the XceptionNet has the highest influence on the predictions of ECNNXAI for class 0 (adenocarcinoma) and class 3 (serrated).

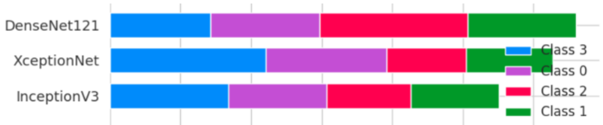


Figure 6: Summary plot showing impact of base classifiers on final predictions of the ECNNXAI

7 CONCLUSION

In this work, we proposed an ensemble network called the ECNNXAI for classifying colon histopathology images into one of the four target classes namely, adenoma, adenocarcinoma, serrated and normal. Three pre-trained CNN models were used as the base classifiers for the stacked ensemble model while the logistic regression model as the meta classifier. Combining the three CNNs using an ensemble model increased the overall performance, the generalization ability to unseen data and the reliability of predictions while reducing the impact of biases as the models errors. Explainable AI techniques like Grad-CAM and SHAP provided interpretation and aided the understanding of the predictions made by the models at various levels. The proposed model can accurately classify colon histopathology images and identify the critical regions in the images that correspond to the cancer types. Future research would examine the optimal number of base classifiers needed in an ensemble model for the classification of images related to colon histopathology and the use of various different deep learning architectures as base classifiers in an ensemble.

8 REFERENCES

[Alb22] Albashish, D. Ensemble of adapted convolutional neural networks (CNN) methods for classifying colon histopathological images. PeerJ Computer Science, 8, p.e1031, 2022.

[Cho17] Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition pp. 1251-1258, 2017.

[Cos23] Costa, V.G. and Pedreira, C.E.. Recent advances in decision trees: An updated survey. Artificial Intelligence Review, 56(5), pp.4765-4800, 2023.

[Ham20] Hameed, Z., Zahia, S., Garcia-Zapirain, B., Javier Aguirre, J. and Maria Vanegas, A. Breast cancer histopathology image classification using an ensemble of deep learning models. Sensors, 20(16), p.4373, 2020.

[He16] He, K., Zhang, X., Ren, S. and Sun, J.. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[Hua17] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q.. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700-4708, 2017.

[Jun21] Junayed, M.S., Anjum, N., Noman, A. and Islam, B.. A deep CNN model for skin cancer detection and classification, 2021.

[Kad23] Kadian, V., Singh, A. and Sharma, K.. A Robust Colon Cancer Detection Model Using Deep-Learning. In 2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC), pp. 665-670, IEEE, 2023.

[Kad20] Kadhim, M.A. and Abed, M.H.. Convolutional neural network for satellite image classification. Intelligent Information and Database Systems: Recent Developments 11, pp.165-178, 2020.

[Li22] Li, K., Yu, R., Wang, Z., Yuan, L., Song, G. and Chen, J.. Locality guidance for improving vision transformers on tiny datasets. In European Conference on Computer Vision, pp. 110-127, Cham: Springer Nature Switzerland, 2022.

[Lun17] Lundberg, S.M. and Lee, S.I.. A unified approach to interpreting model predictions. Advances in neural information processing systems, 30, 2017.

[Mhe22] Mhedhbi, M., Mhiri, S. and Ghorbel, F.. A new deep convolutional neural network for 2D contour classification, 2022.

[Ner22] Nergiz, M.. Collaborative Colorectal Cancer Classification on Highly Class Imbalanced Data Setting via Federated Neural Style Transfer Based Data Augmentation. Traitement du Signal, 39(6), 2022.

[Pav18] Pavlyshenko, B., August. Using stacking approaches for machine learning models. IEEE second international conference on data stream mining & processing (DSMP) (pp. 255-258). 2018.

[Pen02] Peng, C.Y.J., Lee, K.L. and Ingersoll, G.M.. An introduction to logistic regression analysis and reporting. The journal of educational research, 96(1), pp.3-14, 2002.

[Sel17] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D.. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision pp. 618-626, 2017.

[Sen22] Sengoz, N., Yigit, T., Ozlem, O. and Isik, A.H.. Importance of preprocessing in histopathology image classification using deep convolutional neural network. Advances in Artificial Intelligence Research, 2(1), pp.1-6, 2022.

[Sim14] Simonyan, K. and Zisserman, A.. Very deep convolutional networks for large-scale image recognition, 2014.

[Sze16] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z.. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818-2826, 2016,

[Tan19] Tan, M. and Le, Q.. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114), PMLR, 2019.

[Tep22] Tepe, E. and Bilgin, G.. Graph neural networks for colorectal histopathological image classification. In 2022 Medical Technologies Congress (TIPTEKNO), pp. 1-4, IEEE, 2022.

[Wic21] Wickramasinghe, I. and Kalutarage, H.. Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. Soft Computing, 25(3), pp.2277-2293, 2021.

[WHO23] World Health Organization. (2023, July 11). Colorectal cancer. Retrieved from https://www.who.int/news-room/fact-sheets/detail/colorectal-cancer

[Zei21] Zeid, M.A.E., El-Bahnasy, K. and Abo-Youssef, S.E.. Multiclass colorectal cancer histology images classification using vision transformers. In 2021 tenth international conference on intelligent computing and information systems (ICICIS), pp. 224-230, IEEE, 2021.

[Zhu21] Zhu, C., Chen, W., Peng, T., Wang, Y. and Jin, M.. Hard sample aware noise robust learning for histopathology image classification. IEEE transactions on medical imaging, 41(4), pp.881-894, 2021.

Polychromatism of all light waves and a new approach to the interpretation of fluorescence mechanisms; Idea of the fluorescent color monitor

Justyna Niewiadomska-Kaplar

tab edizioni
Viale Manzoni 24c
00185, Rome, Italy
micromacro@ymail.com

ABSTRACT

In scientific literature it is considered that only white light is polychromatic and that it is composed of countless monochromatic waves. This research on light vision mechanisms in biosystems and on the mechanisms of formation of deficits in color discrimination reveals that not only white light is polychromatic but all light waves are. This hypothesis brings numerous consequences, two of which we want to address in this work:
- what does the measurement of wavelength mean when the radiation is polychromatic;
- the presence of wavelengths shorter or longer than the incident one must not mean Stokes and anti-Stokes emissions but can be interpreted as the selective absorption/reflection mechanism of a compound radiation.

Based on these considerations, the mechanisms of fluorescence are interpreted as:
1. Pseudo-fluorescence formed by the selective absorption/reflection of **consecutive monochromatic components** of polychromatic blue/violet light – **spectral colors** production;
2. Fluorescence formed by the selective reflection of **non-consecutive monochromatic components** of polychromatic blue/violet light – **fluorescent color** production.

This new approach to the interpretation of fluorescence mechanisms is the theoretical starting point for project a fluorescent colors monitor.

Keywords

Wavelength of polychromatic radiations, Stokes and anti-Stokes emissions or the selective absorption/reflection of polychromatic radiations.

1. INTRODUCTION

This research on light vision mechanisms in biosystems and on the mechanisms of formation of deficits in color discrimination [Nie22b] reveals that not only white light is polychromatic but all light waves are. In this work, two consequences of this hypothesis will be analyzed:
- The measurements of the polychromatic wavelength in spectrophotometric analysis should be considered as an average of its monochromatic components;
- The presence of wavelengths shorter or longer than the incident one must not mean Stokes and anti-Stokes emissions but can be interpreted as the selective absorption/reflection mechanism of a compound radiation.

The objective of this article is to mention a new theoretical approach which concerns the composition of light radiation to address the problems concerning the production of color in monitors with an innovative spirit. In particular, a fluorescent monitor could be useful in signage in conditions of excessive sunlight which causes "blinding" of the monitors.

2. STATE OF ART

In scientific literature it is considered that only white light is polychromatic and that it is composed of countless monochromatic waves. Therefore, the emission of a light wave other than the exciting wave considered monochromatic is referred to as fluorescence or phosphorescence. The fluorescence mechanism is defined as follows: “Property of many solid, liquid and gaseous bodies by which, when they are struck by radiation, they re-emit other radiations with a very short delay (less than 10-8 s), the wavelength of which depends of the exciting radiation and by the nature of the substance (...); the emitted radiations can have a frequency equal to that of the incident radiations (we then have f. by resonance), or, as more often happens, a lower frequency, so that the emission lines and bands for f. are shifted towards longer wavelengths with respect to the lines and absorption bands (Stokes law). In the second case it can happen, and this is what happens in some solids, that the excited atom interacts with the surrounding atoms, in emitting for f. for return to the fundamental state, and that the energy of one or more atomic vibrational quanta also contributes to

https://www.doi.org/10.24132/CSRN.3401.30

the energy of what is emitted; the frequency of f radiation is then greater, and the wavelength shorter, than that of the absorbed radiation: thus there is an evident exception to Stokes' law (➔ Stokes, sir George Gabriel), and the lines, or bands, corresponding to the aforementioned frequencies take the qualification of anti-Stokes lines (or bands).”[1] [Enc13]

3. POLYCHROMATIC CONTENT OF THE LIGHT WAVES OF THE VISIBLE

3.1. Selection of 4 monochromatic waves of the visible and perception of 3 fundamental colors

In this research work it is hypothesized that the spectrum of white light is composed of aggregations of only 4 monochromatic waves: *magenta UV*[2] 384 nm, cyan 432 nm, yellow 576 nm and *magenta IR* 768 nm.

magenta IR	yellow	cyan	*magenta UV*
432 THz	576 THz	768 THz	864 THz
768 nm	576 nm	432 nm	384 nm

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 due to the homologation of frequency multiples by the brain. [Nie22b]

3.2. Composition of 5 bi-chromatic waves of the visible and the additive synthesis of the visible radiation

The spectrum of white light is composed of aggregations of only 4 monochromatic waves: *magenta UV* 384 nm, cyan 432 nm, yellow 576 nm and *magenta IR* 768 nm, grouped in 5 **bi-chromatic** waves. The monochromatic waves will be marked with -1/2, to indicate that they form half of the content of the bright waves, of which unit constitutes a bi-chromatic wave.

single bi-chromatic waves
monochromatic contents [nm]

288	384	432	576	768
384	432	576	768	864

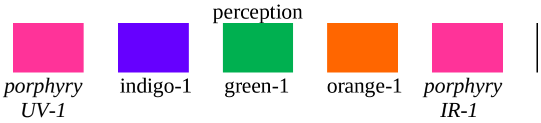
3.2.1. Synthesis of 5 single bi-chromatic waves: indigo-1, green-1, orange-1, *porphyry UV*-1, *porphyry IR*-1

Due to the mechanism of additive synthesis these bi-chromatic waves are perceived as: orange (*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).

single bi-chromatic waves [nm]
monochromatic contents

288	384	432	576	768
384	432	576	768	864

perception

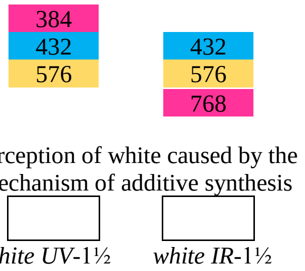


Single bi-chromatic waves will be marked with -1, to indicate that they constitute the fundamental (dual) unit of electromagnetic radiation. The system is called ***PIGOP*** (*P*orphyry, Indigo, Green, Orange, *P*orphyry) and with a different name it wants to differentiate itself from the RGB system.

3.2.2. Synthesis of white 1½

White is synthesized when the quantity of heterogeneous monochromatic waves is equal:

- *magenta UV*-1/2 + cyan-1/2 + yelow-1/2 = *white UV*-1½,
- cyan-1/2 + yelow-1/2 + *magenta IR*-1/2 = *white IR*-1½.


384
432
576
432
576
768
perception of white caused by the mechanism of additive synthesis
white UV-1½
white IR-1½

But since the spectral radiation are bi-chromatic, white is synthesized as part of radiation composed of at least two bi-chromatic waves. (See paragraph 5).

3.2.3. Synthesis of 7 double bi-chromatic waves: purple-2, cyan + white (cyan-2), yellow + white (yellow-2), red-2, magenta + white (magenta-2), *white UV-2, white IR-2;* Distinction between spectral and non-spectral colors

Synthesis between two consecutive bi-chromatic waves (as in the prism) form: purple-2, cyan-½+ white-1½, yellow-½+white-1½ and red-2. In this research the aggregations of consecutive monochromatic and consecutive bi-chromatic waves will be called *spectral colors* to distinguish them from the *non-spectral* colors formed by aggregations

---

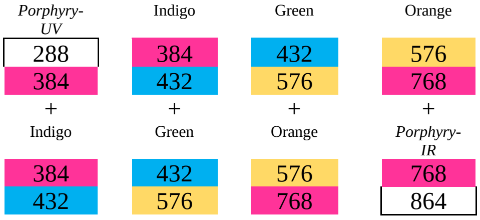
[1] Author's translation.

[2] The denomination of the waves that have wavelengths 384 and 768 nm as *magenta UV* or *IR* and *porphyry UV* and *IR* is proposed from the author.

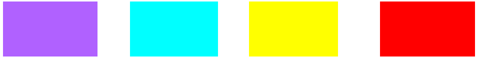
of non-consecutive monochromatic or bi-chromatic waves. We find *spectral* colors for example in stellar radiation and in the decomposition of this radiation. On the other hand, *non-spectral* colors are found, for example, in the superposition of indigo and orange radiations, in reflected radiations in which central monochromatic or bi-chromatic components have been absorbed or in deficits of color vision.

Synthesis between two non-consecutive bi-chromatic waves produces: magenta + white, *white UV*, *white IR*. Double bi-chromatic waves will be marked with -2, to indicate presence of 2 fundamental unit of electromagnetic radiation.

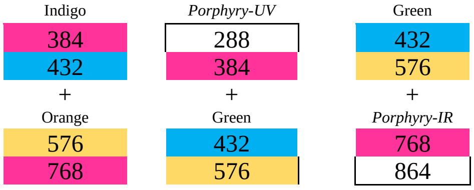
Synthesis between consecutive 2 bi-chromatic waves (as in the prism)



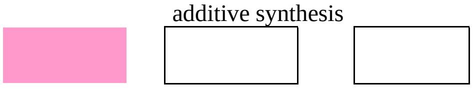
Perception of colors caused by the mechanism of additive synthesis



Synthesis between 2 non-consecutive bi-chromatic waves



Perception of colors caused by the mechanism of additive synthesis

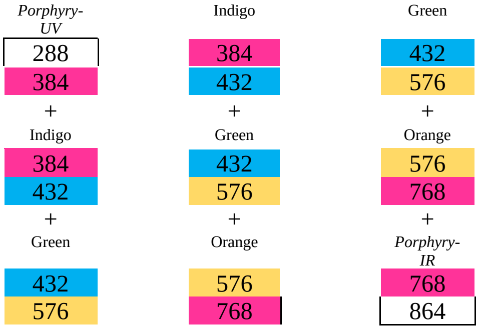


3.2.4. Synthesis of triple bi-chromatic heterogeneous waves

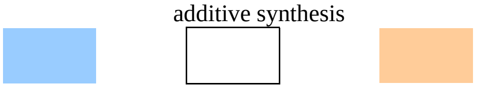
Synthesis of triple bi-chromatic heterogeneous waves (*porphyry IR*-1, orange-1, green-1 and indigo-1, *porphyry UV*-1) form: indigo-1+*white UV*-2, white-3 and orange-1+*white IR*-2.

The aggregations of different 5 bi-chromatic waves together with black form all the existing chromatic shades.

Synthesis between 3 consecutive bi-chromatic waves (interval between stars colors)



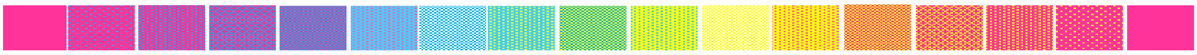
Perception of colors caused by the mechanism of additive synthesis



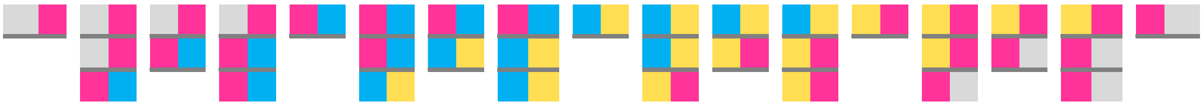
4. POLYCHROMATIC WAVES AND STATISTICAL WAVELENGTH

If we start from the hypothesis of this research that light radiations are polychromatic, the wavelengths should be considered as a statistical average between the monochromatic components of the radiations.

The following table illustrates the aggregations of the bi-chromatic waves: *porphyry IR* and *UV*, orange, green, indigo, their monochromatic contents, 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 consisting of the average of the monochromatic components. This calculation is useful for spectral radiation (aggregation of consecutive bi-chromatic waves) and wavelength calculated in this way will be called the statistical wavelength. (Compare wavelength computation from RGB [Ska23a].



monochromatic contents of polychromatic waves

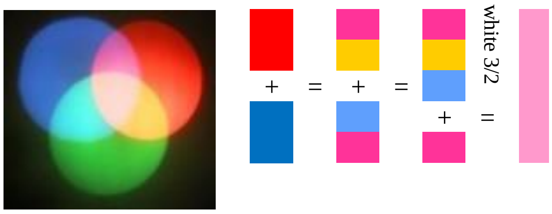


Average measure of the wavelengths of the polychromatic consecutive waves consisting of the average of the monochromatic components: 288, 384, 432, 576, 768, 864 [nm]

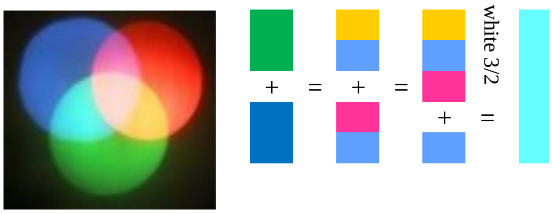
porphyry UV -1	fuchsia -3	violet -2	plum -3	indigo -1	blue -3	cyan -2	turquoise -3	green -1	lemon -3	yellow -2	amber -3	orange -1	pumpkin -3	red -2	wine -3	porphyry IR -1
336	360	372	384	408	440	456	472	504	560	588	616	672	720	744	768	816

5. OVERLAPPING OF THE RED, GREEN AND INDIGO LIGHTS AS EVIDENCE OF THE COMPOSITE NATURE OF THESE RADIATIONS

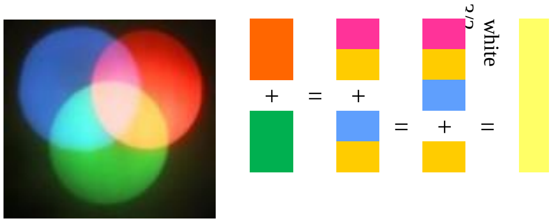
The formation of the magenta, cyan and yellow during the overlapping of the red, green e indigo lights is one of the evidence of the bi-chromatic composition of these radiation provided by the *PIGOP* model, as shown in the following figures:



The formation of magenta clear during the overlap of red and indigo **<u>is possible only if</u>** magenta is one of the components of red and indigo.

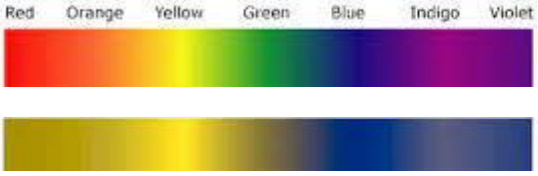


The formation of cyan clear during the overlap of green and indigo **<u>is possible only if</u>** cyan is one of the components of green and indigo.



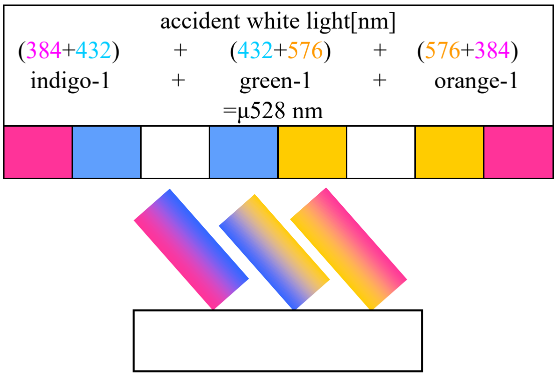
The formation of yellow clear during the overlap of red and green **<u>is possible only if</u>** yellow is one of the components of red and green.

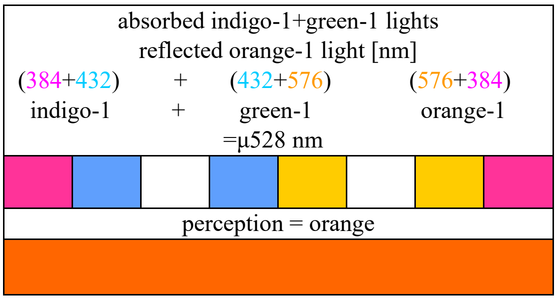
Another evidence of the composite nature of red, green and blue is the confusion between green and red by some types of color blindness in which both colors are perceived as yellow + black. So yellow must be part of both red and green, as show the table below. [Nie23a]

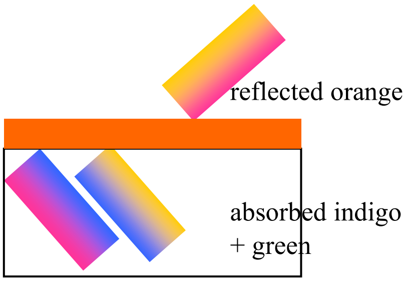


6. ABSORPTION/REFLECTION MECHANISM AND FLUORESCENCE ACCORDING TO THE APPROACH OF THIS RESEARCH

The selective absorption/reflection mechanism refers only to white light since in the literature only white light is considered polychromatic. This mechanism is broadly described as the absorption of a part of the countless monochromatic waves of the incident white light radiation and the reflection of the remaining part, as illustrated in the following table.



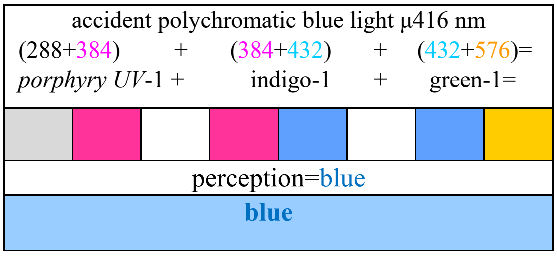

absorbed indigo-1+green-1 lights
reflected orange-1 light [nm]
(384+432) + (432+576) (576+384)
indigo-1 + green-1 orange-1
=µ528 nm
perception = orange

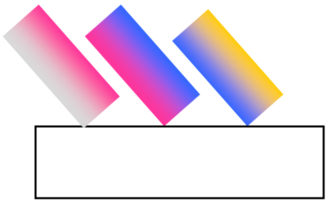


As a consequence of the thesis that all light radiations are polychromatic and that light radiations are aggregations of dual, bi-chromatic waves composed of 4 monochromatic waves: *magenta UV* 384 nm, cyan 432 nm, yellow 576 nm and *magenta IR* 768 nm, two types of selective absorption/reflection mechanisms can be identified:

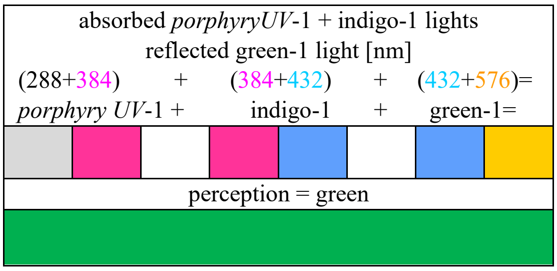
- *spectral* absorption/reflection when consecutive monochromatic or bi-chromatic reflected waves have the wavelength greater or shorter than the incident one,
- *non spectral* absorption/reflection when non-consecutive monochromatic or bi-chromatic reflected waves have wavelength longer or shorter

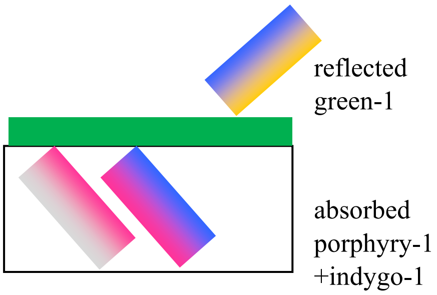
than the incident one; In this case the difference with respect to the *spectral* colors is perceived.
The following tables illustrate both mechanisms with the example of polychromatic blue incident light with a statistical wavelength of 416 nm.


accident polychromatic blue light µ416 nm
(288+384) + (384+432) + (432+576)=
porphyry UV-1 + indigo-1 + green-1=
perception=blue
blue

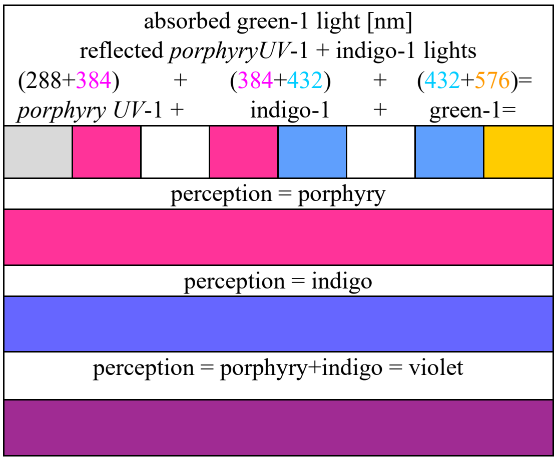


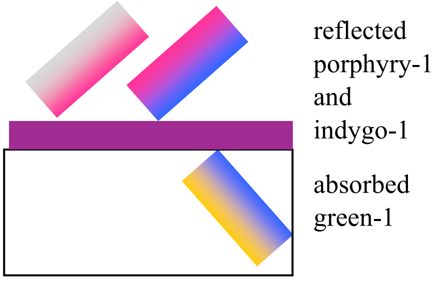
1. *Spectral* absorption/reflection when wavelength of reflected consecutive bi-chromatic waves is **greater than the incident** polychromatic blue **light**.


absorbed *porphyryUV*-1 + indigo-1 lights
reflected green-1 light [nm]
(288+384) + (384+432) + (432+576)=
porphyry UV-1 + indigo-1 + green-1=
perception = green

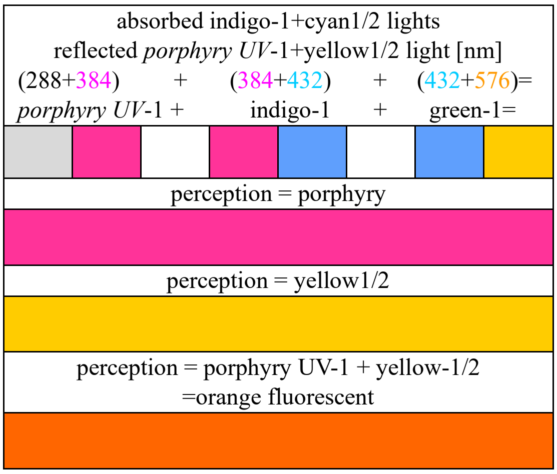


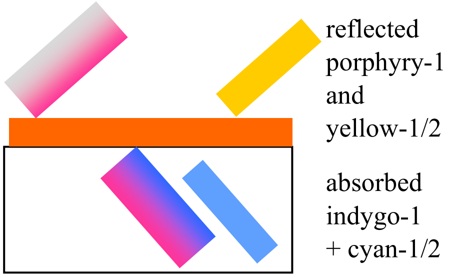
This type of reflection/absorption is called Stokes emission in the literature.
2. *Spectral* absorption/reflection when wavelength of reflected consecutive bi-chromatic waves are **shorter than the incident** polychromatic blue **light**.


absorbed green-1 light [nm]
reflected *porphyryUV*-1 + indigo-1 lights
(288+384) + (384+432) + (432+576)=
porphyry UV-1 + indigo-1 + green-1=
perception = porphyry
perception = indigo
perception = porphyry+indigo = violet



This type of reflection/absorption is called anti-Stokes emission in the literature.
3. *Non spectral* absorption/reflection when non-consecutive monochromatic or bi-chromatic reflected waves have wavelength longer or shorter than the incident one; In this case the difference with respect to the *spectral* colors is perceived.


absorbed indigo-1+cyan1/2 lights
reflected *porphyry UV*-1+yellow1/2 light [nm]
(288+384) + (384+432) + (432+576)=
porphyry UV-1 + indigo-1 + green-1=
perception = porphyry
perception = yellow1/2
perception = porphyry UV-1 + yellow-1/2
=orange fluorescent



When *spectral* or *non-spectral* reflected radiation:
- replaces *magenta IR* 768 nm with *magenta UV* 384 nm or *Porphyry UV* µ336 nm,

- have the *porphyry UV-1* component or its ultraviolet part of 288 nm,

fluorescent colors are formed, and the perceptual system perceives greater energy content respect the same *spectral* color.

In the following tables there are examples of fluorescent radiation that can be obtained from the reflection of non-consecutive waves and from the replacement of magenta *IR* and/or *porphyry IR* with magenta *UV* and/or *porphyry UV*.

Scheme of the composition of monochromatic and bi-chromatic waves reflected by exciting light blue-3 416 nm formed fluorescent colors

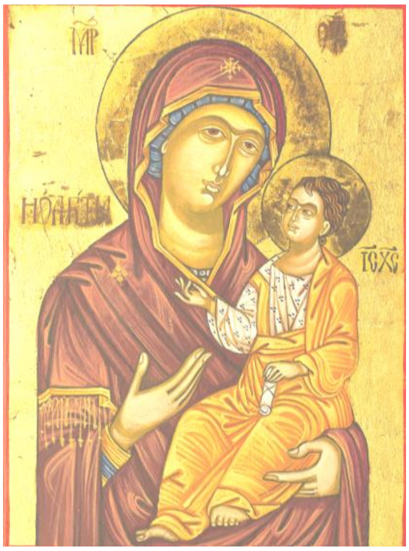
cyan-5/2 fluorescent	orange-3/2 *fluorescent*	yellow UV-1 *fluorescent*	electric blu-3 *spectral*	*white* UV-2 *fluorescent*	red UV-2 *fluorescent*
🟦	🟧	🟨	🟦	⬜	🟥
Monochromatic contents of reflected waves [nm]:					
288	288	288	288	288	288
384	384		384	384	384
			384		384
432			432		
432			432	432	
576	576	576	576	576	576
Statistics wavelengths not useful for spectrographic measures but only to interpret energy content [nm]					
422,4	416	432	416	420	408
Statistic wavelengths of the spectral colors [nm]					
456	672	576		528	744

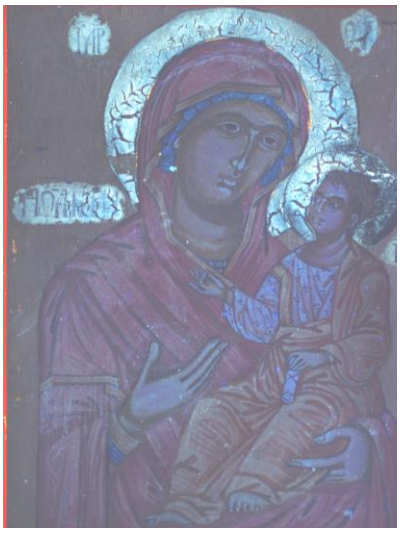
Scheme of the composition of monochromatic and bi-chromatic contents of spectral e fluorescent orange

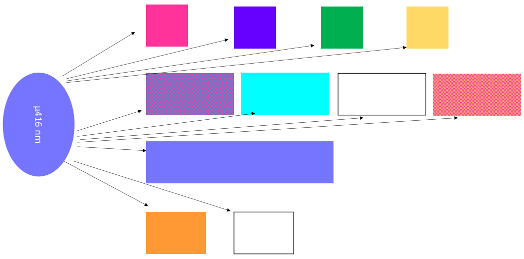
whole spectrum	orange-3/2 *fluorescent*	orange-1 *fluorescent*	orange-3/2 *fluorescent*	*orange-1 spectral*	orange-3/2 *fluorescent*
	🟧	🟧	🟧	🟧	🟧
Monochromatic contents of reflected waves [nm]:					
288	288				288
384	384	384	384		
384					
432					
432					
576	576	576	576		
576				576	576
768				768	768
768					
864			864		
Statistics wavelengths not useful for spectrographic measures but only to interpret and compare energy content [nm]					
	416	432	608	672	736

7. BRIEF MENTION OF VISIBLE FLUORESCENCE INDUCED BY UV RADIATION AS A MULTISPECTRAL INVESTIGATION TECHNIQUE

Visible fluorescence induced by UV radiation is too broad a topic to be addressed in this article. We only want to mention the fact that the radiations used for this technique have a range of around 300-440 nm, therefore according to the theses of this research, which states that light radiations are polychromatic and therefore the wavelengths must be considered an average statistics among the components of a radiation, these radiations between 300-440 nm contain cyan 432 nm, and magenta 384 nm and often also yellow 576 nm. Therefore the "emission" of magenta, violet, blue, white, cyan and even green radiations hues after illumination of an object with for example violet hues could be a simple reflection of its various chromatic content. The table below illustrates icons illuminated with white polychromatic light (a) with violet polychromatic light (b) [Ros12] c. colors that can be reflected from a surface illuminated with a violet light lamp with an ultraviolet component.


a.


b.



c.

8. FLUORESCENT MONITOR IDEA

According to the theoretical approach presented in this research, to obtain fluorescent colors on monitors in the RGB system, the simplest way would be to replace *magenta IR* 768 nm and *porphyry IR* 864/768 nm in red light with *magenta UV* 384 nm and *porphyry UV* 288/ 384 nm as shown in the following table:

Generic composition of bi-chromatic lights in the RGB system

single bi-chromatic waves [nm]
monochromatic contents

	384	432	576	768
	432	576	768	864
		perception		
	🟪 indigo-1	🟩 green-1	🟧 orange-1	🟥 *porphyry IR-1*

Composition of bi-chromatic lights in fluorescent monitor.

single bi-chromatic waves [nm]
monochromatic contents

288	384	432	576	
384	432	576	**384**	
		perception		
🟥 *porphyry UV-1*	🟪 indigo-1	🟩 green-1	🟧 orange-1	

CONCLUSIONS

The idea, born from studies on the measurement of light in biosystems, that not only white light is polychromatic and that light radiations are dual, i.e. composed of two consecutive monochromatic waves between *magenta IR* 768, yellow 576 nm, cyan 432 nm and *magenta UV* 384 nm, form an alternative perspective on the physics of light, making the theoretical explanation of light phenomena, such as fluorescence, simple and giving applied sciences a new theoretical tool for new technological solutions.

REFERENCES

[Nie22b] Niewiadomska-Kaplar J., *Meccanismi della visione del colore e discromatopsie*, Aracne (2022)
[Enc13]
https://www.treccani.it/enciclopedia/fluorescenza
[Ska23a] Skala V., *Multispectral Image Generation from RGB Based on WSL Color Representation: Wavelength, Saturation, and Lightness*, Computers, (2023)
[Nie23a] Niewiadomska-Kaplar J., *Polychromatism of all light waves: new approach to the analysis of the physical and perceptive color aspects*, Computer Science Research Notes (2023)
[Ros12] http://www.giadarossi.com/?p=2224

Impact of Calibration Matrices on 3D Monocular Object Detection: Filtering, Dataset Combination and Integration of Synthetic Data

Alexandre Evain
ESIGELEC
Technopole du Madrillet
Avenue Galilee
France (FRA), 76800, Saint-Etienne-du-Rouvray
alexandre.evain@groupe-esigelec.org

Redouane Khemmar
ESIGELEC
Technopole du Madrillet
Avenue Galilee
France (FRA), 76800, Saint-Etienne-du-Rouvray
redouane.khemmar@esigelec.fr

Mathieu Orzalesi
SEGULA Technologies
19 rue d'Arras
France (FRA), 92000, Nanterre
mathieu.orzalesi@segula.fr

Sofiane Ahmedali
Universite d'Evry Val d'Essonne
2 Rue du Facteur Cheval
France (FRA), 91000, Evry-Courcouronnes
sofiane.ahmedali@univ-evry.fr

ABSTRACT

In traditional 2D object detection, augmenting datasets typically enhances model precision. However, 3D estimations from a 2D image are dependent on the camera's focal length, meaning that differences in focal length may undermine distance estimation, object dimension estimation, and subsequent 3D position estimation. In this article, we attempt to evaluate the impact of different calibration matrices on 3D monocular object detection. Firstly, we assess the impact of different calibration matrices within the same dataset by comparing the performance of filtered, non-filtered, and normalized datasets using the NuScenes dataset as a base. Our results show that filtering the dataset to only keep images sharing the same focal lengths results in increased depth and dimension estimations but at the expense of the other metrics. Then, we investigate the impact of dataset combination on 3D monocular object detection, focusing on the integration of datasets with varying focal lengths and matrices. Leveraging the NuScenes dataset, this time augmented with additional synthetic data from GTA, we evaluate the efficacy of dataset combination in improving model performance across a range of metrics. Contrary to our initial expectations, incorporating additional datasets does not consistently result in 2D performance improvements depending on their visual appearance, but also does not always result in decreased 3D performance either, despite their different focal lengths providing the model with contradictory 3D visual information, as long as the data contained is accurately labeled, showing that dataset combination has the potential to improve 3D monocular object detection.

Keywords

3D Monocular Object Detection, Dataset combination, Computer Vision, Camera Calibration Matrix, Focal Length, Dataset Filtering, Dataset Normalisation

1 INTRODUCTION

1.1 Dataset Combination

In machine-learning object detection, data availability is often the primary bottleneck in achieving optimal model performance, especially for real-life applications where diverse scenarios must be accurately captured. While numerous datasets (such as [1, 2, 3, 4]) exist for training machine learning models, each comes with its own set of limitations, necessitating the strategic combination of datasets to address these constraints effectively. Firstly, existing datasets exhibit variations in terms of the covered conditions and object classes. As an example, certain datasets might focus solely on daytime, clear weather conditions and do not have scenarios such as nighttime or adverse weather conditions. Other datasets are limited by the object class they cover, resulting in models able to detect cars but not buses as another example ([1] covers 3 classes while [2] cover 9). In addition, domains are also covered inequally by the existing datasets: While certain domains such as road situations might boast an abundance of datasets,

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.

https://www.doi.org/10.24132/CSRN.3401.31

others like railroad scenarios suffer from scarcity. Furthermore, variations in the quality of images and annotations across datasets pose another challenge, as some datasets have their own specific image ratios or limited image quality.

Given these limitations, dataset combination emerges as a viable strategy to address the shortcomings inherent in individual datasets: Firstly, by aggregating multiple datasets, we can mitigate the incompleteness of coverage by incorporating diverse conditions, object classes, and domains into the training data, thereby enhancing the model's ability to generalize across a broader spectrum of scenarios. Then, dataset combination can increase the model's robustness. By amalgamating datasets with different scenarios, we can increase the representation of real-world scenarios.

Another interesting aspect of dataset combination is that it allows using synthetic datasets. These datasets, like [5], though providing perfect annotations, are still visually distinct from real-world images, potentially compromising their utility in practical applications. This limitation can be alleviated by combining synthetic datasets alongside real-world datasets. This way, we not only enrich the latter with additional scenarios but also imbue the synthetic datasets with greater realism. This amalgamation helps bridge the gap between synthetic and real-world data, enhancing the model's adaptability to real-life scenarios.

1.2 Focal Length and Contradictory Visual Information

In traditional 2D object detection tasks, the primary objective revolves around accurately identifying objects within an image. However, transitioning to 3D monocular object detection introduces additional challenges, such as estimating objects' distances, dimensions, sizes, and orientations.

While augmenting the dataset might bolster the 2D aspect of detection, the same approach may not yield commensurate improvements for the 3D predictions. This discrepancy is caused by the relationship between the camera's focal length, its field of view, the scene geometry, and the resulting image. Unlike in 2D detection, where object appearance suffices, the 3D estimations are fed potentially contradictory information:

- The size of an object within an image is dependant not only on its true dimensions but also on its distance from the camera and the camera's focal length, as expressed in the Equation (1):

  $$d = \frac{f \cdot \mathrm{H}}{h} \tag{1}$$

  With h the heights of the object in pixels, H the actual width of the object, and f the camera's focal length.

  As a result, two images portraying objects of apparently identical dimensions might convey disparate distance estimations if captured using cameras with different focal lengths.

- In addition, the camera's field of view also affects the orientation estimations as well as the positions of the objects within the image. This effect is very noticeable when using wide-angle cameras, leading to side distortion, and method such as [6, 7] solve this problem by either making a FOV-independant detection model or by using sensor fusion.

- Finally, inaccuracies in distance or dimension estimation can reverberate through subsequent stages of 3D position estimation.

Some existing detection methods like [8, 9] take the focal length into account in their detection models. However, we aim to investigate the impact of different calibration matrices and dataset combinations on conventional 3D object detection models that do not explicitly consider focal length, thus highlighting the importance of these factors in enhancing detection performance. We aim to evaluate this impact through two primary avenues: Firstly, we investigate the ramifications of incorporating images captured using diverse camera setups within a single dataset, each equipped with its unique calibration matrix. Then, we assess the consequences of amalgamating datasets sourced from disparate sources, each characterized by distinct calibration matrices. We aim to determine the relationship between dataset composition and the efficacy of 3D monocular object detection and determine whether dataset composition can be overcome on the dataset level without directly modifying the object detection models.

In summary, our work has the following contributions:

- We examine how different calibration matrices within datasets affect 3D detection, and we assess

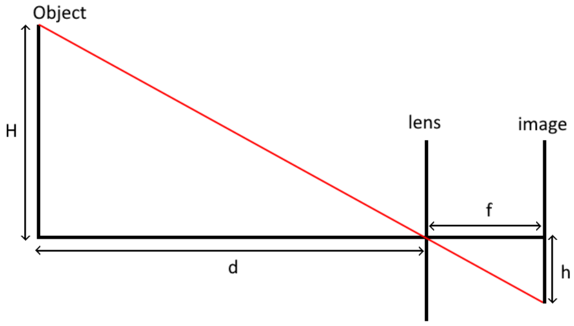


Figure 1: Relation between the camera's focal length f, the object's height in the image h, the real object's height H and the object's distance to the camera d, as explained in the Equation (1)

the efficacy of filtering methods in addressing contradictory information from diverse calibration matrices.

- We analyze combining datasets for 3D monocular object detection, understanding how different focal lengths affect model performance and 3D estimations.
- We analyze the effects of the visual normalization of the camera calibration matrices through distortion, particularly in the context of synthetic datasets, on prediction accuracy and model performance.

2 RELATED WORK

2.1 Monocular 3D Object Detection taking focal length into account

Monocular 3D object detection research has explored various methodologies to address inherent challenges in autonomous driving perception tasks. All the methods in this section focus on taking into account the focal length in the model itself for more accurate 3D predictions; we aim to differ from their approach by instead acting on the datasets themselves to see the effects of different focal lengths, of dataset filtering/normalization/combination on 3D object detection without modifying the object detection method itself. Our focus is to see what can be done to improve the detection while only changing the dataset.

The most recent method is MonoGDG[6], which proposes a geometry-guided domain generalization framework, addressing gaps at both camera and feature levels by incorporating geometry-based image reprojection and feature disentanglement techniques. The paper addresses most of the limitations of dataset combination by taking Focal Length, FOV Distortion, FOV Range, Camera Orientation, and Image Appearance into account in its architecture.

Another method, ODD-M3D[10] proposes object-wise dense depth estimation, improving depth estimation accuracy by randomly sampling points from the bounding box area of each object, and then using these using pre-generated sampled points for their depth estimation method, instead of relying on a single center point.

MonoEdge[8] proposes utilizing local appearance cues, particularly the edges of 3D bounding boxes, to estimate depth and global yaw angle directly from object appearance in images, enabling object depth and yaw angle derivation without requiring absolute size or position information and bypassing the need for explicit camera intrinsic parameters as well.

MonoUNI[11] introduces a unified optimization target, normalized depth, which addresses discrepancies between vehicle and infrastructure-side detection due to variations in pitch angle and focal length.

Advancements in depth estimation techniques, such as those explored by Deep Optics[12], integrate optics and image processing to improve depth estimation performance, with implications for 3D object detection tasks.

Flexibility and adaptability are crucial considerations, with approaches like Objects Are Different[13] offering frameworks that explicitly account for truncated objects and adapt multiple approaches for object depth estimation.

Additionally, incorporating motion cues for depth estimation and object detection presents promising avenues. Monocular 3D Object Detection with Depth from Motion[14] explores synergies between camera ego-motion and monocular understanding to improve accuracy and robustness in object detection tasks.

2.2 Datasets

The field of 3D object detection in computer vision has experienced notable advancements, driven by the availability of diverse datasets catering to various aspects of the task. These datasets have played a crucial role in benchmarking algorithms and propelling progress in the domain. However, each dataset typically focuses on specific scenarios, sensor modalities, or annotation techniques, leaving certain aspects of 3D object detection unexplored. In this section, we review related work spanning the development of diverse datasets for 3D object detection.

The KITTI dataset family [1, 15, 16, 17] has been instrumental in driving progress in various computer vision tasks relevant to autonomous driving. It introduced fundamental benchmarks for stereo, optical flow, visual odometry, and 3D object detection. KITTI-360 further extends this by focusing on suburban driving scenarios, while Virtual KITTI leverages computer graphics to propose an efficient real-to-virtual world cloning method.

Road datasets such as nuScenes [2] and Rope3D [18] provide diverse and challenging data for advancing roadside perception and autonomous driving technologies. These datasets offer extensive annotations and analysis for object detection and tracking, capturing diverse scenes and environmental conditions.

Efforts like A*3D and H3D [19, 20] aim to provide challenging real-world datasets with diverse scenes, varying weather conditions, and dense annotations, pushing the boundaries of autonomous driving research into more challenging environments.

Enhancing 2D datasets for 3D object detection, Cityscapes 3D [21] and PASCAL3D+ [22], augment existing datasets with 3D annotations, providing richer annotations and increasing variability for studying 3D detection and pose estimation.

Leveraging computer games for dataset creation has emerged as a cost-effective alternative to manual data collection. Approaches like "Ground Truth from Computer Games" and "Free Supervision From Video Games" [5] demonstrate the feasibility of extracting pixel-level semantic labels and ground truth annotations from video games in real time, providing visually realistic images for training models on large-scale datasets.

3 METHOD

3.1 Dataset Configurations

Table 1: NuScenes[2] Matrix 1. The first three columns are the camera's rotation matrix (3,3), and the last column (1,3) is its translation matrix.

1252	0.0	826	0.0
0.0	1252	469	0.0
0.0	0.0	1.0	0.0
0.0	0.0	0.0	1.0

To understand the impact of having different camera calibration matrices present within the same dataset on 3D monocular object detection, we decided to use the NuScenes dataset[2] trimmed down to focus on three distinct classes (cars, pedestrians, and cyclists), because NuScenes images have three different camera calibration matrices:

- Matrix 1, with occurrences totaling 16,443 instances, constituting 56.71% of the dataset.
- Matrix 2, featuring 12,082 occurrences, representing 41.67% of the dataset.
- Matrix 3, featuring 468 occurrences, representing 1.61% of the dataset.

To systematically assess the influence of these matrices, we initiated our analysis by creating a validation split consisting of 2000 images using only Matrix 1. All our models have been tested on this single validation split.

Then, we created the different training sets: First, we formed a base training from NuScenes set comprising 5000 images. Within this set, 2886 images corresponded to Matrix 1 such as 2, while 2114 images were associated with Matrix 2 (no image used Matrix 3). Since NuScenes is a sequential dataset, we made sure that all the images from the validation and the training set come from different sequences, to avoid training our model on images too similar to the ones used for our validation. Then, to see the effect of the absence of Matrix 2 images, we made a filtered training set containing only the 2886 images belonging to Matrix 1 while all images associated with Matrix 2 were omitted from this subset. Finally, we created a normalized training set to see the potential influence of normalization techniques. Here, all 2886 images associated with Matrix 1 were retained, while the 2114 images linked to Matrix 2 underwent normalization procedures.

Table 2: Calibration Matrices of the Training and Validation Sets.

Dataset	**Matrix 1**	**Matrix 2**	**Matrix 3**	**Total**
NS3	2886	2114	-	5000
NS3 Filt	2886	-	-	2886
NS3 Norm	2886 + *2114(n)*	-	-	5000
NS3 GTA	2886	2114	2500	7500
NS3 GTA Norm	2886 + *2500(n)*	2114	-	7500
NS3 VAL	2000	-	-	2000

Figure 2: Image of the NuScenes[2] dataset using the Matrix 1.

Having established the effect of different matrices within the NuScenes dataset, we then proceeded to investigate the impact of dataset combinations on 3D monocular object detection. To achieve this, we incorporated another dataset, GTA[5], a fully synthetic dataset that is visually distinct from NuScenes while also presenting images with a different camera calibration matrix. Initially, we examined the effects of a simple combination without any further changes, by combining the base NuScenes dataset with 2500 images sourced from GTA, adhering to its native calibration matrix (Matrix 4). Then, we combined the base NuScenes dataset with a normalized version of GTA. In this scenario, 2500 images from GTA underwent normalization procedures to align with the calibration Matrix 1.

3.2 Model Configuration & Evaluation Method

For our 3D monocular object detection tests, we employed a homemade version of YOLOv7 modified to do 3D Monocular Object Detection, which we called MYv7. We used a modified method of [23] to adapt YOLOv7 from 2D to 3D monocular object detection. Previous observations we made seemed to indicate that, dataset combination yields inferior results compared to training without any form of dataset augmentation. However, these initial findings also suggested that dataset combination could reach its maximum accuracy at a higher epoch than regular model training and that this maximum was greater than the regular model's.

Table 3: Effect of different matrices within the same dataset on 2D and 3D object detection metrics on the car class.

Model	**Epochs**	**P**	**R**	**mAP @0.5**	**mAP @0.95**	**Depth Err.**	**CS**	**DS**	**OS**
NS3	250	0.687	0.785	0.748	**0.445**	0.0465	**0.935**	0.869	**0.953**
NS3 Filt	250	**0.699**	0.774	**0.749**	0.442	**0.0462**	0.931	**0.871**	0.949
NS3 Norm	250	0.652	**0.789**	0.737	0.435	0.0482	0.643	0.745	0.596
NS3	1000	0.68	**0.856**	0.794	**0.534**	0.0391	**0.948**	0.89	**0.978**
NS3 Filt	1000	0.736	0.828	**0.797**	0.529	**0.0379**	0.946	**0.891**	0.975
NS3 Norm	1000	**0.779**	0.8	**0.797**	0.531	0.0388	0.637	0.805	0.773
NS3	2000	0.795	0.808	0.782	**0.548**	0.0366	**0.954**	0.9	0.984
NS3 Filt	2000	**0.801**	0.804	0.782	0.545	**0.0353**	0.953	**0.901**	**0.985**
NS3 Norm	2000	0.782	**0.821**	**0.788**	**0.548**	0.0371	0.643	0.83	0.842
NS3	4000	0.843	0.776	**0.768**	**0.558**	0.036	**0.959**	0.909	**0.988**
NS3 Filt	4000	**0.857**	0.768	0.764	0.55	**0.0333**	0.958	**0.911**	0.987
NS3 Norm	4000	0.827	**0.792**	0.767	0.557	0.0352	0.65	0.85	0.88
NS3	6000	0.862	0.762	**0.756**	**0.555**	0.0351	**0.96**	0.914	**0.991**
NS3 Filt	6000	**0.889**	0.74	0.748	0.547	**0.033**	**0.96**	**0.916**	0.987
NS3 Norm	6000	0.869	**0.765**	0.755	**0.555**	0.0357	0.657	0.857	0.894
NS3	MAX	0.856	**0.766**	**0.765**	**0.557**	0.0358	0.959	0.91	0.988
NS3 Filt	MAX	**0.877**	0.748	0.747	0.547	**0.0334**	**0.96**	**0.916**	**0.987**
NS3 Norm	MAX	0.868	0.765	0.754	0.554	0.0355	0.658	0.858	0.896

To ensure a fair comparison among different models, we decided to evaluate them at their peak performance, determined by the maximum accuracy they could attain regardless of epoch. However, achieving this pinnacle necessitated extensive training durations, with model maximums usually reached after 5000-6000 epochs. Consequently, we had to reduce the dataset size and employ a smaller model variant, specifically the Tiny model. While this inherently caps the performance potential compared to larger models, even with these changes the training process still extends over three months. This means that replicating the experiment with the entire NuScenes dataset or with heavier models is not practical.

The evaluation itself is done using the usual 2D metrics (Precision, Recall, and Average Precision (AP) at IoU thresholds of 0.5 and 0.95) combined with further metrics tailored for each specific 3D estimation, these metrics are the Depth Error, the Center offset & Dimension Score defined by [24] and the Orientation Score. This has a two-fold use: firstly, this grants us insights into how each 3D estimation is affected by dataset filtering/normalization/combination, and it allows us to tailor the model learning to focus on a specific metric if needed. We assess Depth Error using metrics such as Absolute Relative Error (Abs Rel), Squared Relative Error (SRE), Root Mean Square Error (RMSE), and logarithmic RMSE (log RMSE).

4 EXPERIMENTAL RESULTS

4.1 Different Matrices within the same dataset

As we can see in Table 3, filtering the dataset increases the accuracy of both depth and dimension estimations. This improvement can be attributed to eliminating contradictions introduced by having varying focal lengths within the same dataset. Since focal length particularly affects the depth and dimensions estimations, it is logical that these two metrics are the most improved ones by the filtering. The improved depth estimations are further confirmed by the use of RMSE metrics in Table 4.

Conversely, the non-filtered regular dataset outperforms the filtered version in mAP@95 and center position & orientation estimations. The superior performance in mAP@95 can be attributed to the larger quantity of data, as additional properly labeled data invariably benefits 2D object detection tasks. Additionally, the enhanced CS and OS metrics can be attributed to the importance of data volume outweighing the impact of differing focal lengths on these specific estimations.

However, the normalized dataset demonstrates poor performance across all metrics except Recall and depth estimation. Despite removing the different focal lengths while keeping additional data, the distortions resulting from normalization and reprojections lead to

Table 4: Effect of different matrices within the same dataset on depth estimation results on the car class.

Method	**Depth RMSE**									
Epochs	250	500	1000	1500	2000	3000	4000	5000	6000	MAX
NS3	2.89	2.77	2.46	2.34	**2.22**	2.19	2.17	2.11	2.05	2.13
NS3 Filt	**2.81**	2.93	**2.37**	**2.31**	2.23	**2.06**	**2.01**	**1.92**	**1.96**	**2.01**
NS3 Norm	2.97	**2.68**	2.5	2.33	2.26	2.17	2.09	2.05	2.06	2.04

Table 5: Effect of dataset combination on depth estimation results on the car class.

Method	Depth RMSE									
Epochs	250	500	1000	1500	2000	3000	4000	5000	6000	MAX
NS3	2.89	2.77	2.46	2.34	**2.22**	2.19	2.17	2.11	**2.05**	2.13
NS3 GTA	**2.88**	**2.71**	**2.41**	**2.28**	2.26	**2.16**	**2.04**	**2.06**	2.09	**2.06**
NS3 GTA Norm	2.97	**2.71**	2.58	2.48	2.4	2.3	2.27	2.32	2.3	2.27

a significant decrease in prediction accuracy across all metrics. The decreased depth error seen in Table 4 does show that getting rid of the different focal lengths does help the depth estimation. However, the loss in image quality due to normalization cannot be compensated for by removing contradictory information.

The qualitative results given by Figure 3 further confirm these observations: Compared to the left image (NS3 model), the center image (NS3 Filt model) has more accurate bounding box sizes and positions, however, the predictions made by the base model on the left have more accurate orientation estimation.

4.2 Dataset Combination

Based on the results in Table 6, incorporating an additional dataset with its own focal length and matrix did not consistently lead to a complete decrease in 3D metrics. This unexpected outcome suggests that the accuracy of the synthetic 3D position data, despite introducing contradictory focal length information, may partially compensate for such discrepancies. Utilizing a dataset with precise ground truth compared to the image may allow the accuracy of the additional data to offset the negative impact of conflicting focal lengths and matrices. As we can see in Table 5, the depth estimations in the combined models were often better than the ones made by the regular model.

Further deviating from expectations, introducing the new synthetic dataset appears to have had a detrimental effect on 2D metrics, contrary to the usual anticipation of higher results with additional information. This unexpected observation is likely attributed to the visual appearance of the synthetic dataset. Despite its photorealism, it remains too far from real-life images to significantly enhance 2D object detection performance.

Moreover, the normalization process once again results in excessive distortion, hindering the attainment of satisfactory results. While applied to synthetic data, normalization yielded improved mAP@0.5 results, the substantial decrease in other 3D metrics outweighs this improvement. Consequently, the overall impact of normalization on synthetic data appears unfavorable, underscoring the importance of considering the trade-offs between data preprocessing techniques and resultant performance metrics.

Another result of the dataset combination that we can see from the qualitative evaluation in Figure 3 is the difference in labeling between datasets: in the GTA dataset, even cut-off objects are labeled, while they are not in the NuScenes dataset. This means that once these datasets are combined, our NS3 GTA model can detect cut-off cars using information from the GTA dataset, while these are not part of the NuScenes label, which means that the quantitative evaluation considers these detections as erroneous. This effectively means that even if the labels themselves are correct, both datasets must have similar criteria for object labelization to avoid contradicting each other.

Table 6: Effect of dataset combination on 2D and 3D object detection metrics on the car class.

Model	Epochs	P	R	mAP @0.5	mAP @0.95	Depth Err.	CS	DS	OS
NS3	250	**0.687**	0.785	0.748	0.445	**0.0465**	0.935	0.869	**0.953**
NS3 GTA	250	0.679	**0.798**	**0.749**	**0.447**	0.0472	**0.937**	**0.87**	0.95
NS3 GTA N	250	0.628	0.78	0.721	0.421	0.0491	0.643	0.745	0.609
NS3	1000	0.68	**0.856**	**0.794**	**0.534**	0.0391	0.948	0.89	**0.978**
NS3 GTA	1000	0.7	0.85	**0.794**	0.532	**0.0386**	**0.949**	**0.891**	0.976
NS3 GTA N	1000	**0.744**	0.817	**0.794**	0.521	0.0427	0.622	0.79	0.763
NS3	2000	**0.795**	0.808	0.782	**0.548**	0.0366	**0.954**	**0.9**	0.984
NS3 GTA	2000	0.772	0.821	0.785	**0.548**	**0.0363**	**0.954**	**0.9**	**0.985**
NS3 GTA N	2000	0.757	**0.824**	**0.796**	0.544	0.0405	0.633	0.815	0.825
NS3	4000	0.843	0.776	0.768	**0.558**	0.036	**0.959**	**0.909**	**0.988**
NS3 GTA	4000	**0.861**	0.762	0.764	0.55	**0.0347**	**0.959**	0.907	0.987
NS3 GTA N	4000	0.821	**0.788**	**0.783**	0.556	0.0393	0.643	0.836	0.867
NS3	6000	0.862	0.762	0.756	0.555	0.0351	0.96	0.914	0.991
NS3 GTA	6000	0.865	0.766	0.755	0.549	0.035	0.959	0.912	0.989
NS3 GTA N	6000	0.835	0.778	0.774	0.556	0.0395	0.65	0.846	0.879
NS3	MAX	0.856	0.766	0.765	**0.557**	0.0358	0.959	0.91	0.988
NS3 GTA	MAX	**0.873**	0.756	0.755	0.55	**0.0351**	**0.96**	**0.912**	**0.989**
NS3 GTA N	MAX	0.839	**0.771**	**0.773**	0.551	0.0392	0.654	0.845	0.886

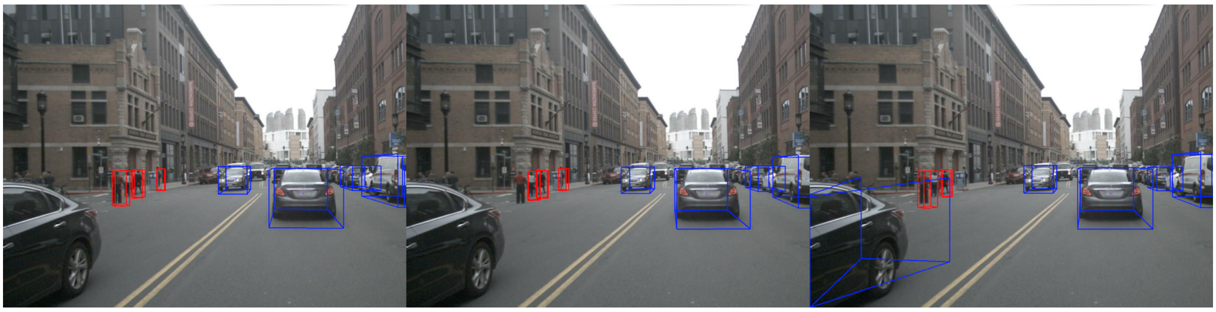


Figure 3: Comparison of 3D predictions on the Nuscenes[2] dataset. Left: NS3 model, Center: NS3 Filt model, Right: NS3 GTA model.

5 DISCUSSION

In this study, we explored the complexities surrounding the presence of several camera calibration matrices within the same dataset, as well as dataset combination and its impact on 3D monocular object detection.

Firstly, our analysis focused on the comparative performance of filtered and non-filtered datasets within the context of 3D monocular object detection. Filtering the dataset, which aimed to remove contradictions introduced by different focal lengths within the same dataset, yielded positive effects on depth error and dimension estimations as expected. However, our results show that the non-filtered regular dataset often outperformed its filtered counterpart in metrics such as mAP@95 and center position & orientation estimations. This discrepancy can be attributed to the larger quantity of data within the non-filtered dataset, which benefits 2D object detection by providing additional properly labeled data.

While dataset combination offers a promising strategy for addressing the limitations inherent in individual datasets, its efficacy varies depending on several factors. Despite anticipating improved 2D detection results with the addition of new synthetic data, we observed a negative impact on 2D metrics. This outcome suggests that although synthetic datasets offer perfect annotations, their visual dissimilarity from real-world images can compromise their utility in practical applications. Using synthetic datasets in combination with real ones does not always result in increased performance, and careful consideration must be given to dataset composition to ensure alignment with the objectives of the object detection task.

Another finding is the overwhelmingly negative effect of normalization of the camera matrix through artificial distortion, both on real and synthetic datasets, whether it affected the core training data or additional data. While normalization may improve certain metrics, such as mAP@0.5, not only are these improvements inconsistent, but its adverse effects on other 3D metrics did outweigh the benefits, and models trained on normalized datasets performed poorly in most cases.

On a more positive note, incorporating additional datasets with their own focal lengths and matrices did not consistently result in a complete decrease in 3D metrics. It seems that datasets with precise ground truth compared to the image allowed the accuracy of the additional data to partially compensate for the introduction of contradictory information induced by different focal lengths.

6 CONCLUSION

We can conclude that having different camera focal lengths within a training set does not inherently decrease the performance of a 3D monocular object detection model. While filtering the dataset results in more accurate depth and dimension estimations, it is at the expense of other results as filtering gets rid of useful data. Introducing additional data from other datasets does not necessarily reduce the accuracy of the model's 3D estimations as long as this data contains precise ground truth, the visual appearance of this new data matters a lot for 2D object detection. Finally, attempting to normalize the focal length through artificial distortion just provides unreliable data for 3D estimations.

7 ACKNOWLEDGMENTS

This research is funded and supported by SEGULA Technologies. We would like to thank SEGULA Technologies for their collaboration and for allowing us to conduct this research. We would like to thank also the engineers of the Autonomous Navigation Laboratory (ANL) of IRSEEM for their support. In addition, this work was performed, in part, on computing resources provided by CRIANN (Centre Regional Informatique et d'Applications Numeriques de Normandie, Normandy, France).

8 REFERENCES

[1] A. Geiger, P. Lenz, *et al.*, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361, IEEE, 2012.

[2] H. Caesar, V. Bankiti, *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] M. Cordts, M. Omran, *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[4] M. Everingham, L. Van Gool, *et al.*, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[5] S. R. Richter, V. Vineet, *et al.*, "Playing for data: Ground truth from computer games," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pp. 102–118, Springer, 2016.

[6] F. Yang, H. Chen, *et al.*, "Geometry-guided domain generalization for monocular 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 6467–6476, 2024.

[7] M. Furst, R. Jakkamsetty, R. Schuster, and D. Stricker, "Learned fusion: 3d object detection using calibration-free transformer feature fusion," 2023.

[8] M. Zhu, L. Ge, *et al.*, "Monoedge: Monocular 3d object detection using local perspectives," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 643–652, 2023.

[9] X. Shi, Q. Ye, *et al.*, "Geometry-based distance decomposition for monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15172–15181, October 2021.

[10] C. Park, H. Kim, J. Jang, and J. Paik, "Odd-m3d: Object-wise dense depth estimation for monocular 3d object detection," *IEEE Transactions on Consumer Electronics*, 2024.

[11] J. Jinrang, Z. Li, and Y. Shi, "Monouni: A unified vehicle and infrastructure-side monocular 3d object detection network with sufficient depth clues," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[12] J. Chang and G. Wetzstein, "Deep optics for monocular depth estimation and 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[13] Y. Zhang, J. Lu, and J. Zhou, "Objects are different: Flexible monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3289–3298, June 2021.

[14] T. Wang, J. Pang, and D. Lin, "Monocular 3d object detection with depth from motion," in *European Conference on Computer Vision*, pp. 386–403, Springer, 2022.

[15] Y. Liao, J. Xie, *et al.*, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2022.

[16] A. Gaidon, Q. Wang, *et al.*, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4340–4349, 2016.

[17] Y. Cabon, N. Murray, and M. Humenberger, "Virtual kitti 2," *arXiv preprint arXiv:2001.10773*, 2020.

[18] X. Ye, M. Shu, *et al.*, "Rope3d: The roadside perception dataset for autonomous driving and monocular 3d object detection task," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21341–21350, 2022.

[19] Q.-H. Pham, P. Sevestre, *et al.*, "A*3d dataset: Towards autonomous driving in challenging environments," in *2020 IEEE International conference on Robotics and Automation (ICRA)*, pp. 2267–2273, IEEE, 2020.

[20] A. Patil, S. Malla, *et al.*, "The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9552–9557, IEEE, 2019.

[21] N. Gahlert, N. Jourdan, *et al.*, "Cityscapes 3d: Dataset and benchmark for 9 dof vehicle detection," *arXiv preprint arXiv:2006.07864*, 2020.

[22] Y. Xiang, R. Mottaghi, *et al.*, "Beyond pascal: A benchmark for 3d object detection in the wild," in *IEEE winter conference on applications of computer vision*, pp. 75–82, IEEE, 2014.

[23] A. Mauri, R. Khemmar, *et al.*, "Lightweight convolutional neural network for real-time 3d object detection in road and railway environments," *Journal of Real-Time Image Processing*, vol. 19, pp. 499–516, Jun 2022.

[24] H.-N. Hu, Q.-Z. Cai, *et al.*, "Joint monocular 3d vehicle detection and tracking," 2019.

Spotlight Control for Real-Time Targeting

Mika Benjamin Heinemann, Thomas Kernbauer, Philipp Fleck, Clemens Arth
Institute of Computer Graphics and Vision, University of Technology, Graz
heinemann, kernbauer, philipp.fleck, clemens.arth@tugraz.at

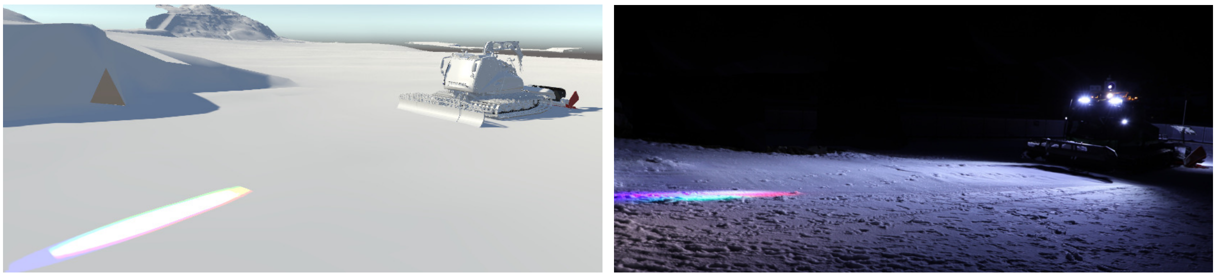


Figure 1: The spotlight control via reconstruction-based targeting (left) and the corresponding result atop a real snow groomer (right). The light cone of the spotlight can be controlled with the user's input in the reconstruction.

ABSTRACT

Off-road heavy machinery such as snow groomers or excavators, often operate in low-light and hazardous environments. In this work, we explore the development of an intelligent camera-spotlight system with automatic and manual control to illuminate points of interest, such as obstacles or individuals at risk. We implement a prototype as proof of concept and integrate our workflow using a standard lighting protocol and a single-board computer. The presented calibration of the camera and spotlight ensures high precision in the desired use cases. In addition to testing the prototype on a real snow groomer, we evaluated its performance in terms of accuracy and repeatability. Overall, we showcase the usability of a commercially available spotlight in the context of spatial augmented reality in heavy machinery applications.

Keywords

Augmented Reality, Spatial Augmented Reality, Spotlight

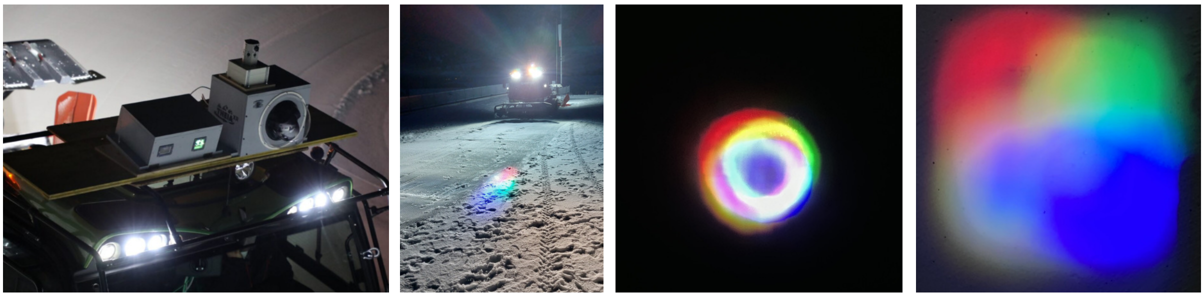
1 INTRODUCTION

Heavy machinery operators need a diverse set of skills, including a thorough understanding of their surroundings and the ability to make rapid decisions. From construction to more specialized domains, such as snow grooming, off-highway vehicles are often deployed in diverse weather conditions and challenging environments. To optimize the efficient and safe use of heavy machinery, operators depend not only on their expertise but also on a wide array of sensors and complex auxiliary systems. Recently, Augmented Reality (AR) has played a vital role in empowering operators to improve their situational awareness, making it an interesting potential component of vehicle operation [Sitompul and Wallmyr, 2019]. However, these solutions typically rely on immersive technology, such as head-mounted displays (HMDs) or head-up displays (HUDs). Less obtrusive, yet lightweight, and effective solutions to support operators in their tasks are rare.

This work introduces a novel prototype, designed for off-highway heavy machinery, which addresses visibility challenges commonly encountered by operators. Our prototype includes, among other components, a commercially available spotlight, a camera, and a single-board computer (SBC). The primary workflow of our system is to automatically identify points of interest, *e.g.* obstacles, animals or humans, within the machine's operational environment and illuminate them. The utilization of image-based detection and targeted illumination offers an approach to aid operators in challenges associated with poor visibility without steering away their attention from their main task, which manual steered spotlights would do.

Although our prototype is usable in various domains, we have tested our concept on a snow groomer, as shown in Figure 2. With these machines, the slopes are prepared, usually during the night, as skiers and snowboarders use the pistes during the day. Therefore, proper artificial lighting is necessary for workers to navigate the slope and avoid accidents involving various objects, including humans and animals. In this on-site evaluation, the spotlight not only efficiently illuminated critical areas of interest, but also demonstrated its ability to enhance overall operational efficiency and safety.

In general, we propose an intelligent spotlight control system that is capable of alleviating the responsibilities of heavy machinery operators by illuminating their surroundings and utilizing detections to illuminate possible threats automatically. We propose two types of



(a) (b) (c) (d)

Figure 2: (a) Our prototype (besides other devices) mounted atop a snow groomer. (b) The light cone is visible in front of the vehicle. The spotlights cone on a flat surface in a distance of 1 meter (c) and 4 meters (d).

control, *i.e.* a control based on a reconstructed 3D environment and an image-based approach that utilizes the prototype's camera. For reconstruction-based control, we use a 3D simulation as depicted in Figure 1. To correctly localize our reconstruction w.r.t. the real world, we incorporate information from the control area network (CAN) bus of the vehicle. Besides this simulation-based control, we provide an image-based alternative to manually (or automatically) detect and shine on points of interest in the image. In addition, we evaluate the accuracy of our prototype and the overall workflow. The main contributions of this work are:

- A novel spotlight prototype that can be controlled via image- or simulation-based input.
- A registration routine to find the transformation between the prototype's main components, *i.e.* the spotlight and the camera.

2 RELATED WORK

Even though the expression capacity of our spotlight is intrinsically restricted, our work is related to the field of spatial augmented reality (SAR), AR in heavy machinery and projector-camera systems. Therefore, we provide a summary of methods and seminal work in these fields, which influenced our implementation and design considerations.

2.1 Spatial Augmented Reality

Raskar *et al.* [Raskar et al., 1999] pioneered SAR as a camera projector setup for surface extraction and virtual object rendering, highlighting its independence from HMD. Subsequent SAR research focuses on realistic projections in static scenes [Raskar et al., 2001], requiring both the projector and the model to remain static. Today, SAR approaches are found in various research endeavors, ranging from smart manufacturing [Uva et al., 2018] to robot communication [Coovert et al., 2014]. In addition to projection-based SAR, some works employ laser projections, *i.e.* Schwerdtfeger *et al.* [Schwerdtfeger et al., 2008], and Glossop and Wang [Glossop and Wang, 2003]. Lately, Kernbauer *et al.* [Kernbauer et al., 2024] introduced a laser-based projector-camera system for heavy machinery operations.

2.2 AR in Heavy Machinery

AR is widely used to assist heavy machinery operators [Sitompul and Wallmyr, 2019]. This includes methods to improve remote vehicle operation, *e.g.* in forklift [Sarupuri et al., 2016] or forestry [Palonen et al., 2017] applications. Predominantly, video-based AR or see-through devices such as HMDs and HUDs [Santana-Fernández et al., 2010, Palonen et al., 2017] are utilized. Furthermore, in-cabin support is provided by applications using diminished reality (DR), as in [Aromaa et al., 2020]. Introduced in the 1990's by Mann [Mann, 1999], DR enables the selective removal or reduction of elements from the user's real-world environment.

2.3 Projector-Camera Calibration

Since we want to control the spotlight using the camera as sensor input, we need a mapping between the coordinate systems of the camera and the spotlight. This problem is studied by calibration methods of projector-camera setups, such as structured light systems [Sadlo et al., 2005, Yamauchi et al., 2008, Liao and Cai, 2008, Kimura et al., 2007]. Our prototype can be classified as a projector-camera setup with a naive projector. Therefore, the calibration approaches introduced in these works, are closely related to viable calibration mechanisms for our prototype. Besides the utilization of precalibrated cameras, they use calibration patterns to compute the correspondence between the projector and the camera. Note that these calibration approaches depend on the accuracy of the camera calibration, as errors in the camera calibration can lead to misalignments between the camera and the projector.

A different approach was introduced by Moreno and Taubin [Moreno and Taubin, 2012], which directly establishes correspondences between projector pixels and

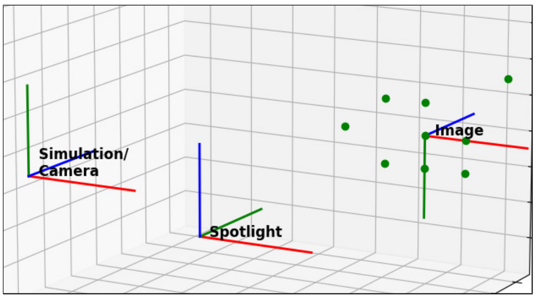


Figure 3: The utilized calibration pattern in 3D space and the coordinate systems corresponding to the simulation/camera, the spotlight, and the image frame, respectively. The red, green, and blue axes correspond to the x, y, and z direction of each coordinate system.

3D world points using local homographies and therefore avoids the need for a precalibrated camera.

However, projector-camera systems necessitate a more sophisticated calibration approach, as projectors in these systems can project more complex and delicate patterns than our spotlight. Hence, we introduce a simple but accurate blob-based calibration approach with a precalibrated camera, as described in Section 3.1.

3 CAMERA-SPOTLIGHT SYSTEM

Since our system is essentially a projector-camera system, we describe our concepts for calibrating and obtaining control signals for the spotlight. More precisely, given a position in a sensor device, *i.e.* the camera, we provide a conversion to calculate the control signal for the spotlight to shine its light cone in the right direction.

3.1 Spotlight Calibration

To register the camera and the spotlight in relation to each other, we need to establish correspondences between the projected pattern and the corresponding image obtained with the camera. Moreover, determining the position of a camera with respect to known 3D points within its field of view is a well-established problem [Marchand et al., 2015], referred to as camera localization or camera pose estimation. To solve this problem, usually patterns in an image and their corresponding location in 3D (*i.e.* in the real world) are leveraged to calculate the position and rotation of the camera. Therefore, given the 3D positions of the projected light cone on a surface and the corresponding 2D positions as the pixel coordinates of the centers of the light cone in the image plane, the Perspective-n-Point (PnP) algorithm by Fischler and Bolles [Fischler and Bolles, 1981] is utilized. Note that the calibration pattern must have distinct characteristics that remain unequivocally identifiable, even in the presence of variations in rotation, reflection, and translation. Furthermore, the utilized spotlight has to be able to generate the selected pattern. Since our spotlight is essentially capable of producing only circular blobs in the form of light cones on target surfaces, we opt to sequentially generate an asymmetric circular blob pattern, as shown in Figure 3. Even though only four distinct points are required to solve the pose estimation, more correspondences yield a more robust result. Therefore, we project a pattern of nine blobs. We found that further increasing the number of points in the pattern resulted in only marginal improvements, generally within a one-millimeter margin.

3.2 Position to Control

Assuming a given target position $p_{\text{target}} \in \mathbb{R}^3$ and the spotlight position $p_{\text{spotlight}} \in \mathbb{R}^3$, the directional vector $\vec{V}_{\text{dir}} \in \mathbb{R}^3$ is given by

$$\vec{V}_{\text{dir}} = p_{\text{target}} - p_{\text{spotlight}}. \tag{1}$$

Since the spotlight is controlled with two 16-bit numbers, corresponding to its pan (yaw) and tilt (pitch), we need to adapt the 3D directional vector to meet the spotlight's requirements by the subsequent computation of both rotation angles. Here, the pan-direction corresponds to a rotation around the y-axis (green) and the tilt corresponds to a rotation around the z-axis (blue) in the spotlight coordinate frame in Figure 3. More precisely, we project $\vec{V}_{\text{dir}}$ onto the target plane and use trigonometric functions to calculate the corresponding angle. *E.g.* to compute the rotation angle in the yaw direction, *i.e.* $\theta_{\text{pan}} \in \mathbb{R}$, we omit the target height and only consider the plane in which the spotlight is rotating. Formally,

$$\theta_{\text{pan}} = \operatorname{atan2}\left(\det(\vec{V}_0, \vec{V}_{\text{pan}}), \vec{V}_0 \cdot \vec{V}_{\text{pan}}\right), \tag{2}$$

with $\vec{V}_0, \vec{V}_{\text{pan}} \in \mathbb{R}^2$ being the projected current position- and the target vectors, respectively. Finally, we obtain the discrete control values $c_{\text{pan}}, c_{\text{tilt}} \in [1, 2^{16})$ by a simple conversion

$$c_{\text{pan}} = \theta_{\text{pan}} \frac{2^{16}-1}{c_{\text{pan,max}}} \quad \text{and} \tag{3}$$

$$c_{\text{tilt}} = \theta_{\text{tilt}} \frac{2^{16}-1}{c_{\text{tilt,max}}}, \tag{4}$$

with $c_{\text{pan,max}}, c_{\text{tilt,max}} \in \mathbb{R}$ being the maximum possible respective angles in radians.

4 PROTOTYPE

Our main contribution is a working prototype that can automatically and manually shine on interesting objects within its field of view. The main components are listed in Table 1 and include a camera module, a commercially available disco spotlight, and a control unit (*i.e.* a Raspberry Pi). The final prototype is depicted in Figure 4.

Spotlight	*PUZILOZA ZQDMX512*
Camera	*Arducam AR0134-C*
SBC	*Raspberry Pi 4B*
DMX Shield	*CQRobot DMX Shield MAX485*

Table 1: Main hardware components integrated into our prototype.

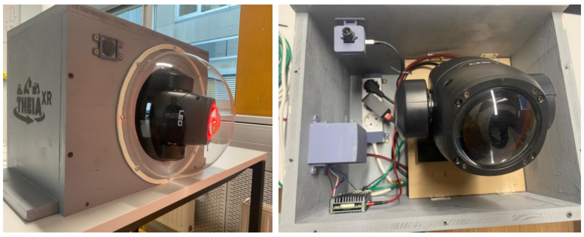


Figure 4: (Left) The assembled prototype in its sealed and finished case. (Right) The prototype without the frontal wood panel and the acrylic half dome.

4.1 Software Control

To control the spotlight, we support multiple scenarios, *i.e.* a camera-based as well as a simulation-based approach. Both approaches presume different levels of environmental knowledge. However, the spotlights control input is in any case a 3D position information in world coordinates.

4.1.1 Image-based

The camera-based control of the moving head can be applied automatically, *i.e.* by object detection or by manual user input on the camera image. Nevertheless, since we need a 3D target position in the spotlight-coordinate system, both methods require assumptions about the environment. These assumptions are use-case dependent. For example, in the field of pedestrian detection, the output of object detection methods (*i.e.* the position, dimensions, and class of the detected object in the image space) could be fused with the assumption of the average height of the detected class to obtain a 3D position. In addition, assuming that the detected object is on a planar ground plane and a known 3D mounting position of our spotlight system, one could utilize a homography to estimate the 3D position of the detected object.

4.1.2 Simulation-based

Alongside camera-based control, simulation-based control can be applied. For this, a reconstruction of the environment and a registered prototype within this environment are necessary. The selected target positions in the simulation can naively be translated into real-world target positions. For example, in the context of heavy machinery, the vehicle must initially be localized within an environmental reconstruction. Using the information provided in the CAN bus of these vehicles, accurate localization is possible, even in harsh environments. This input yields a manual control over the light cone, overwriting the automatic aiming via image-based control.

4.2 Hardware

The hardware of our prototype is divided in three main parts, indicated by their respective color of shape in Figure 5. The first category entails the parts that provide data for the spotlight control, *i.e.* the user interface and the camera. The second category corresponds to the units that process the input signal and compute suitable control values for the spotlight, namely an SBC and a digital multiplex (DMX) component. The third category involves the spotlight itself. Our hardware components are protected by a robust casing.

4.2.1 Casing and Protection

The spotlight's external casing is built with 10 millimeter plywood. To guard the electrical components from water ingress, we apply water-proof varnish on the wood (hence the gray appearance in Figure 4). Furthermore, the casing is equipped with ports for power and Ethernet connections. Both the camera and the spotlight are encapsulated with acrylic glass, to ensure usability while keeping the components protected.

4.2.2 Camera

In the hardware setup for this research project, a pivotal component is the camera. We utilize the *Arducam* model *AR0134-C* [Arducam, 2024], a low-cost high frame global shutter camera module. This camera module produces images with a resolution of 1280×960 pixels and yields a maximum frame rate of 54 frames per second. Furthermore, the camera is compatible with various microcontroller platforms.

4.2.3 Spotlight

The used spotlight has a 100-watt light-emitting diode (LED) light source, able to perform a 540° pan rotation and a tilt of up to 180°. As shown in Figure 1 and Figure 2, this commercially available spotlight is powerful enough to be visible even in the headlights of a snow groomer. Note that the collimator lens of the spotlight is not adjustable. Therefore, the focus point of the light cone does not change.

4.2.4 Control Signal Flow

The camera shares a universal serial bus (USB) connection with the SBC. The external computer communicates via the message queue telemetry transport (MQTT) protocol with the SBC. Since our spotlight follows the DMX lighting protocol, we employ a receiver/transmitter device on the general purpose input/output (GPIO) pins of the SBC. This

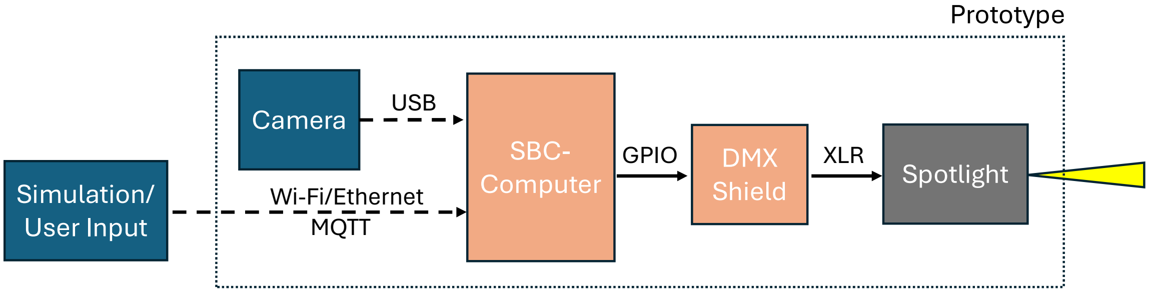


Figure 5: Schematic description of the workflow of our prototype. Note that there are different input modalities, *i.e.* user input or camera input. The SBC on board converts the signal into discrete spotlight control values. Finally, the DMX shield forwards the values to the spotlight via an electrical connection used commonly in stage lighting, *i.e.* an XLR connection.

device, *i.e.* the DMX shield, is necessary to send the control signals to the spotlight. We employ a level converter for the data transfer between the DMX shields voltage and the SBC output. Note that this is the only required add-on for the DMX shield to work properly with the SBC. The spotlight is connected to the DMX shield via an electrical connection used commonly in stage lighting, *i.e.* an XLR cable.

The DMX protocol has stringent timing requirements. Therefore, we leverage the capabilities of an external library, specifically the open lighting architecture (OLA) [Newton, 2024]. Our implementation process follows a multithreaded approach: a buffer is created to hold the control data, while another thread repeatedly transmits these data to the spotlight. Consequently, any modifications made to the data are reflected in the subsequent transmission, ensuring dynamic and responsive integration with the DMX protocol.

5 EVALUATION

In this section, our goal is to assess the practical effectiveness of our prototype. Therefore, we provide an objective evaluation of both the repeatability and accuracy of the spotlight, as well as a heuristic test to estimate the usability of our overall workflow.

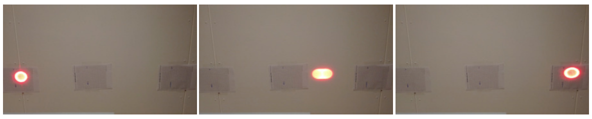
5.1 Accuracy and Repeatability

This test aims to assess the long-term accuracy and stability of a spotlight system, focusing in particular on potential deviations induced by repetitive or high-speed movements, called *overshooting*. Overshooting occurs when the spotlight head extends beyond the intended target position due to momentum, potentially leading to inaccuracies and misalignment.

Our experiment involves the spotlight system facing a wall at a fixed one-meter distance, with grid papers located at predefined target positions. To quantify the spotlight's accuracy, we utilize high-resolution video. Exemplary images of this test are depicted in Figure 6. The test consists of randomly pointing in various directions before returning to the target positions. The results of this evaluation are shown in Table 2. With 100% speed, *i.e.* approximately 300°/ in the pan- and 200°/ sec in the tilt direction, the spotlight loses its localization and misses given target positions by a large margin. However, with a speed reduction of 50%, the targets are hit accurately, even after 400 repetitions.

Test Description		Error [m] at % Speed		
Test Run	Repetitions	50%	75%	100%
1	100	0.0	0.0	0.41
2	200	0.0	0.04	0.62
3	400	0.0	0.13	0.65

Table 2: Test results of our accuracy and repeatability test with random rotations. With increasing speed, the spotlight becomes inaccurate. All results are targets w.r.t. on a plane with a distance of 1 meter from the prototype.



(a) 00:22.50 (b) 00:22.75 (c) 00:23.00

Figure 6: These image series show exemplary screenshots of our evaluation. The spotlight changes its target from the general starting point to the rightmost target in images (a) to (c). Using timestamps, we measure the speed of the spotlight. Furthermore, we measure the accuracy using grid paper on the wall.

5.2 Heuristic Real World Test

As the previous test provides insight into the functionality of hardware components within a controlled environment, we want to evaluate the performance of the entire set-up under typical operational conditions.

Our test setup is depicted in Figure 7. Given that the spotlight would be placed on top of heavy machinery,

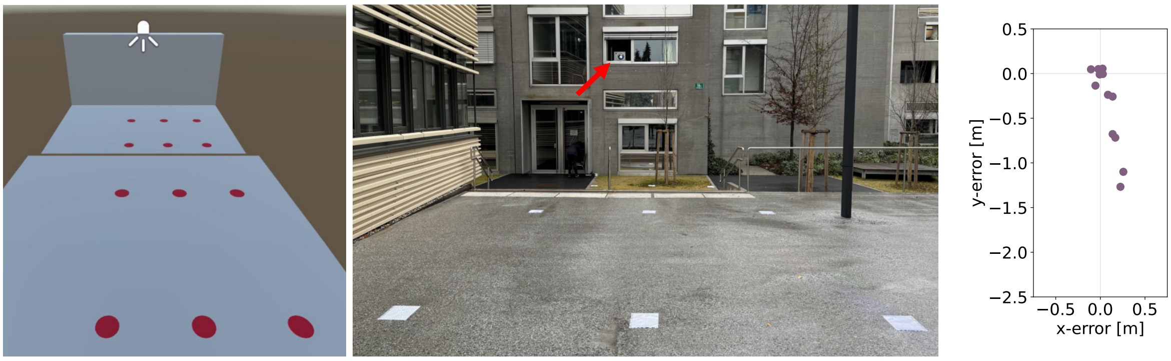


Figure 7: (Left) A simple reconstruction of the test environment in *Unity*. (Middle) Real-world setup of our test corresponding to the reconstruction. The prototype's position is indicated with a red arrow. (Right) Mean of all deviations from the target location. The error increases with distance to the spotlight due to the cumulative nature of errors in the aiming angle. Note that all targets were still illuminated as the spotlights light cone increased with the distance.

we placed it at a height of 4.5 meters. Subsequently, we placed 12 targets on the floor to simulate operational scenarios. The 3×4 grid of targets is placed close to each other, 3 meters from the wall, with a 2-meter gap between them. Furthermore, the final two rows of targets were located at elevated positions. To test simulation-based control, we constructed a rudimentary terrain model to simulate real-world conditions and facilitate interactions with the targets.

The results of this evaluation can be seen in Figure 7. Due to the increased distance and therefore the increase in the elliptic light cone on the projection plane (see Figure 2), our prototype is capable of enlightening all targets, even if the center point of the light cone would not hit the targets accurately from far distances. However, as the target is still lit by the spotlight, the impact on the usability is negligible in our test. Although we meticulously measured the distances between the targets and carefully constructed our terrain model, potential inaccuracies between the reconstruction and the real-world test setup are inevitable. Since we evaluate both the accuracy and repeatability, we assume that the deviation between the projection center and the target positions is due to these inaccuracies.

5.3 Limitations

The primary limitation of our prototype is due to the incorporated hardware components. The chosen spotlight was not manufactured with our use-case in mind. As a result, small angular errors in the spotlight's orientation which are not problematic in its initial area of application may lead to serious consequences, *i.e.* inaccuracies of the light cone especially at greater distances. This issue could potentially be mitigated with the usage of alternative hardware, *e.g.* a modified spotlight with more robust or precise actuators.

Additionally, the prototype's functionality is constrained by the capabilities of the camera sensor. In conditions with poor visibility, such as low light or adverse weather, alternatives like thermal imaging sensors could offer viable solutions.

Moreover, in the simulation-based control input scenario, the accuracy depends on the quality of the 3D reconstruction and the precise localization within it. The availability of accurate 3D reconstructions might impede the practical implementation of our prototype using this control input. However, this challenge can potentially be overcome by utilizing the camera as an input device. There, the prototype's ability to automatically detect and illuminate points of interest relies on estimating the 3D position of target objects, which inherently introduces ambiguity. Employing a different sensor, such as a light detection and ranging (LiDAR) device, could potentially mitigate this issue.

6 CONCLUSION

In this paper, we introduce a novel prototype designed to aid operators in heavy-machine applications. We propose a camera-based spotlight system, to facilitate automatic detection and lighting and therefore enhance visibility in various conditions. Furthermore, we provide a calibration workflow to accurately register the components in the prototype w.r.t. each other and propose two control modalities, *i.e.* a simulation-based and an image-based procedure. Furthermore, we evaluate the device on the basis of accuracy and repeatability. Even though the commercially available spotlight yields inaccuracies at high speeds, its precision is sufficient to be employed on such a device.

Overall, we introduce a novel camera-spotlight prototype with different input modalities. Our simple calibration routine is sufficiently accurate such that our prototype is capable of operating in real-world scenarios, as we demonstrate our concept on a real snow groomer with different input modalities.

Future work may include the incorporation of other sensing technology, *e.g.* thermal cameras or LiDAR devices, to propel the prototypes' application possibilities.

7 ACKNOWLEDGMENTS

Funded by the European Union under Grant Agreement No. 101092861. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

8 REFERENCES

[Arducam, 2024] Arducam (2024). 1.2MP ONSemi AR0134/AR0135. https://www.arducam.com/products/camera-breakout-board/global-shutter-camera/1-2mp-onsemi-ar0134-ar0135/. Online; accessed October 19, 2023.

[Aromaa et al., 2020] Aromaa, S., Goriachev, V., and Kymäläinen, T. (2020). Virtual Prototyping in the Design of See-Through Features in Mobile Machinery. *Virtual Reality*, 24:23–37.

[Coovert et al., 2014] Coovert, M. D., Lee, T., Shindev, I., and Sun, Y. (2014). Spatial Augmented Reality as a Method for a Mobile Robot to Communicate Intended Movement. *Computers in Human Behavior*, 34:241–248.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395.

[Glossop and Wang, 2003] Glossop, N. D. and Wang, Z. (2003). Laser Projection Augmented Reality System for Computer-Assisted Surgery. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pages 239–246.

[Kernbauer et al., 2024] Kernbauer, T., Tschulik, M., Fleck, P., and Arth, C. (2024). Spatial Augmented Reality for Heavy Machinery using Laser Projections. In *Proceedings of the International Workshop on eXtended Reality for Industrial and Occupational Supports*, pages 98–109.

[Kimura et al., 2007] Kimura, M., Mochimaru, M., and Kanade, T. (2007). Projector Calibration using Arbitrary Planes and Calibrated Camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Liao and Cai, 2008] Liao, J. and Cai, L. (2008). A Calibration Method for Uncoupling Projector and Camera of a Structured Light System. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 770–774.

[Mann, 1999] Mann, S. (1999). Mediated Reality. *Linux Journal*, 1999(7).

[Marchand et al., 2015] Marchand, E., Uchiyama, H., and Spindler, F. (2015). Pose Estimation for Augmented Reality: A Hands-on Survey. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2633–2651.

[Moreno and Taubin, 2012] Moreno, D. and Taubin, G. (2012). Simple, Accurate, and Robust Projector-Camera Calibration. In *Proceedings of the IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 464–471.

[Newton, 2024] Newton, S. (2024). Open Lighting Architecture. https://www.openlighting.org/ola/. Online; accessed August 24, 2023.

[Palonen et al., 2017] Palonen, T., Hyyti, H., and Visala, A. (2017). Augmented Reality in Forest Machine Cabin. *International Federation of Automatic Control*, 50(1):410–417.

[Raskar et al., 1999] Raskar, R., Welch, G., and Fuchs, H. (1999). Spatially Augmented Reality. *Augmented Reality: Placing Artificial Objects in Real Scenes*, pages 64–71.

[Raskar et al., 2001] Raskar, R., Welch, G., Low, K.-L., and Bandyopadhyay, D. (2001). Shader Lamps: Animating Real Objects With Image-Based Illumination. In *Proceedings of the Eurographics Workshop*, pages 89–102.

[Sadlo et al., 2005] Sadlo, F., Weyrich, T., Peikert, R., and Gross, M. (2005). A Practical Structured Light Acquisition System for Point-Based Geometry and Texture. In *Proceedings of the Symposium on Point Based Graphics*, pages 89–99.

[Santana-Fernández et al., 2010] Santana-Fernández, J., Gómez-Gil, J., and del Pozo-San-Cirilo, L. (2010). Design and Implementation of a GPS Guidance System for Agricultural Tractors Using Augmented Reality Technology. *Sensors*, 10(11):435–447.

[Sarupuri et al., 2016] Sarupuri, B., Lee, G. A., and Billinghurst, M. (2016). Using Augmented Reality to Assist Forklift Operation. In *Proceedings of the Australian Conference on Computer-Human Interaction*, pages 16–24.

[Schwerdtfeger et al., 2008] Schwerdtfeger, B., Pustka, D., Hofhauser, A., and Klinker, G. (2008). Using Laser Projectors for Augmented Reality. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 134–137.

[Sitompul and Wallmyr, 2019] Sitompul, T. A. and Wallmyr, M. (2019). Using Augmented Reality to Improve Productivity and Safety for Heavy Machinery Operators: State of the Art. In *Proceedings of the International Conference on Virtual-Reality Continuum and its Applications in Industry*.

[Uva et al., 2018] Uva, A. E., Gattullo, M., Manghisi, V. M., Spagnulo, D., Cascella, G. L., and Fiorentino, M. (2018). Evaluating the Effectiveness of Spatial Augmented Reality in Smart Manufacturing: A Solution for Manual Working Stations. *International Journal of Advanced Manufacturing Technology*, 94:509–521.

[Yamauchi et al., 2008] Yamauchi, K., Saito, H., and Sato, Y. (2008). Calibration of a Structured Light System by Observing Planar Object from Unknown Viewpoints. In *Proceedings of the IEEE International Conference on Pattern Recognition*.

Real-Size Experience for Virtual Try-On

FuChe Wu
Providence University
200, Sec. 7, Taiwan Boulevard,
Shalu Dist., Taichung City
43301 , Taiwan
fcwu@gm.pu.edu.tw

Wei-Cheng Chen
HOKEI CORP.
No. 103, Daming 1st Rd.,
Tanzi Dist., Taichung City
427007, Taiwan
frank@mail.hokei.com.tw

Andrew Dellinger
Elon University
100 Campus Drive
Elon, NC 27244
adellinger@elon.edu

Han-Wen Cheng
HOKEI CORP.
No. 103, Daming 1st Rd.,
Tanzi Dist., Taichung City
427007, Taiwan
eve@mail.hokei.com.tw

ABSTRACT

In this system, we have established an e-commerce platform that allows users to virtually try on products and experience their sizes. The virtual try-on experience goes beyond just exploring the styles of the items; size is a crucial factor. However, achieving consistent sizes between the real and synthesized worlds requires a calibration process. In a virtual try-on e-commerce system, the challenge lies in enabling users to perceive the true-to-life size experience. Camera calibration plays a pivotal role in this process. By obtaining the parameters of the camera, it becomes possible to display accurate sizes that align with the user's proportions. We propose a straightforward calibration method that leverages a mobile web interface to acquire the camera parameters. This approach ensures that users are provided with realistic sizes, enhancing their virtual try-on experience. Additionally, through coordinate transformations, we convert the obtained parameters into the three.js framework, allowing virtual objects to be tried on in a virtual setting. Our system encompasses the virtual try-on of items such as hats, glasses, earrings, rings, and watches. Users have the flexibility to choose different sizes, enabling them to explore various fitting effects. The URL for accessing the demonstration site is https://showcase.id-yours.com/main/id-yours-glasses.

Keywords

Virtual Try-On, Camera Calibration, Augmented Reality

1. INTRODUCTION

The advent of virtual try-on technologies has revolutionized the way consumers experience and engage with e-commerce platforms, particularly in the fashion and accessory industries. Ensuring an authentic real-size experience is crucial for building user confidence and facilitating informed purchasing decisions. In this context, camera calibration plays a pivotal role in accurately representing object sizes within the virtual environment.

This paper introduces a streamlined approach to camera calibration, leveraging a mobile web application for efficient parameter acquisition. By implementing coordinate transformation techniques, the obtained parameters seamlessly integrate with three.js, enabling users to virtually try on accessories such as hats, glasses, earrings, rings, and watches in their true sizes. The system also empowers users to explore different size options, enhancing the overall virtual try-on experience. The proposed methodology addresses the challenges associated with real-size representation in virtual try-on system, contributing to the advancement of immersive and reliable virtual try-on solutions in the e-commerce landscape.

Camera calibration is a fundamental process in computer vision and imaging technology that involves determining the intrinsic and extrinsic parameters of a camera. The intrinsic parameters include focal length, optical center, and lens distortion, while extrinsic parameters involve the position and orientation of the camera in the 3D world. The goal of camera calibration is to establish a mathematical relationship between the 3D world and the 2D image captured by the camera.

2. Previous Work

Due to the maturity of AR filter tools such as Spark AR or Lens Studio, and the stability of deep learning in body feature tracking with technologies like mediapipe, the virtual try-on has witnessed a plethora of commercial applications. Renowned brands such as Prada, Marc Jacobs, L'Oréal, Nike, Baume & Mercier, Ray-Ban, Sephora, among others, have also ventured into online virtual try-on services in recent years. To address the size issue, it is necessary to provide a known object for calibration, such as a credit card. However, this is limited to the moment of capturing the photo, as without camera calibration, the distance between objects cannot be freely adjusted. To solve this issue, some people may opt for depth cameras [Yan14, Aze16]. However, this approach is not as convenient for applications on regular user smartphones. Yu et al. [Yu23] addresses this issue by first calibrating the smartphone. The rear camera captures the environment to determine the phone's position. When waving in front of the front camera to capture a portrait, the person's position can be obtained. However, since smartphone calibration is required beforehand, it may not be suitable for the general user.

Camera calibration has a history spanning several years. The standard practice involves using a known-sized planar target as a calibration tool [Hei97, Stu99, Zha00, Mei07]. However, for the average user, obtaining and using such tools can be challenging. Therefore, we aim to simplify the calibration process by using a straightforward hand-waving gesture, treating our hands as known planar targets for camera calibration.

While deep learning has become prevalent in recent years, with numerous outstanding studies utilizing it for camera calibration[Ken15, Bog18, Lee21, Pon22, Jin23], the majority of successful results are observed in outdoor settings with roads and tall buildings. In indoor scenarios where there are windows, tables, and chairs providing reference points, better outcomes can be achieved. However, when the background consists of blank walls, the obtained calibration values may not be as satisfactory.

.

3. Camera Calibration

FacialSCDnet [Ber22] estimates the distance between the face and the camera based on facial features. Chen et [Che21, Che22] al.'s research, utilizing the Mano [Rom22] project, allows the acquisition of hand shape and size, applicable on mobile devices. Inspired by their work, can we also use a simple waving gesture to calibrate our camera? While the face can also serve as a calibration pattern, instructing users to move their faces to different positions in front of the camera is relatively challenging. However, using hands provides higher degrees of freedom. It allows for variations in height, distance, and angle, minimizing the risk of encountering local minima in the obtained solutions.

As shown in Figure 1, we treat our hand as a planar pattern. Placed at different positions and distances on the screen, it serves to calibrate the camera's parameters. However, since our hand is not truly planar, a plane is used to approximate the landmarks on the hand's features before calibration. These landmarks are then projected onto the plane to obtain the camera's intrinsic and extrinsic parameters.

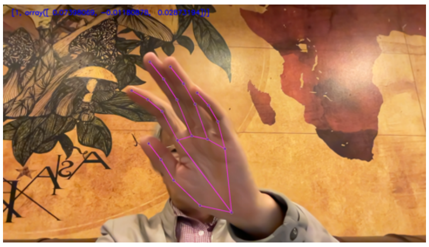


Figure 1. Waving as a calibration pattern

Also, if the user's hand positioning is incorrect, such as the inability to detect a flat plane, we will exclude that frame's image data to avoid affecting the final calibration accuracy.

After obtaining the projection matrix of the camera, we can understand the relationship between the 3D space and its projection in the 2D image space. However, the correct size is still unknown. To address this issue, for a more accurate approach, one can use a credit card as a ruler. If precision is not a critical requirement, estimating finger sizes from training data is an acceptable alternative.

4. Coordinate transform

After obtaining the camera parameters through OpenCV, applying them to OpenGL for rendering involves a coordinate transformation process, as described by Costa et al[Cos19].

As illustrated in Figure 2, the coordinate system on the left represents OpenCV, where y is downward, and z is forward. In the middle, the coordinate system is for OpenGL, with y pointing upward, and -z pointing forward. On the right is the Normalized Device Coordinates after passing through the Clip-Space Frustum.

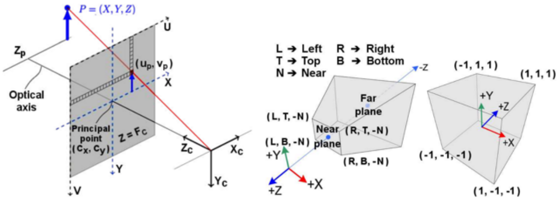


Figure 2. The difference in coordinate systems between OpenCV and OpenGL.

Therefore, we can derive the OpenGL projection matrix from the intrinsic parameters of OpenCV, while also accounting for the axis direction.

Assuming that the intrinsic parameters of OpenCV are

$$K = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix}$$

Here, f_x and $f_y$represent the focal lengths in the x and y directions, while c_x and c_y denote the principal points in the x and y directions.

$$P_{opengl} =$$

$$\begin{bmatrix} \frac{2f_x}{width} & 0 & \frac{width - 2C_x}{width} & 0 \\ 0 & \frac{2f_y}{height} & -\frac{height - 2C_y}{height} & 0 \\ 0 & 0 & -\frac{Z_{far} + Z_{near}}{Z_{far} - Z_{near}} & -\frac{2Z_{far}Z_{near}}{Z_{far} - Z_{near}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Where P_{opengl} is the OpenGL projection matrix, $width, height$ represent the width and height of the image, respectively. Z_{far}, Z_{near} denote the far and near planes of the frustum, respectively.

Through this relationship, the parameters corrected by OpenCV can easily combine the camera's 2D space with the given spatial coordinates. As shown in Figure 3, we can display virtual cubes using three.js at the intersections of the image pattern.

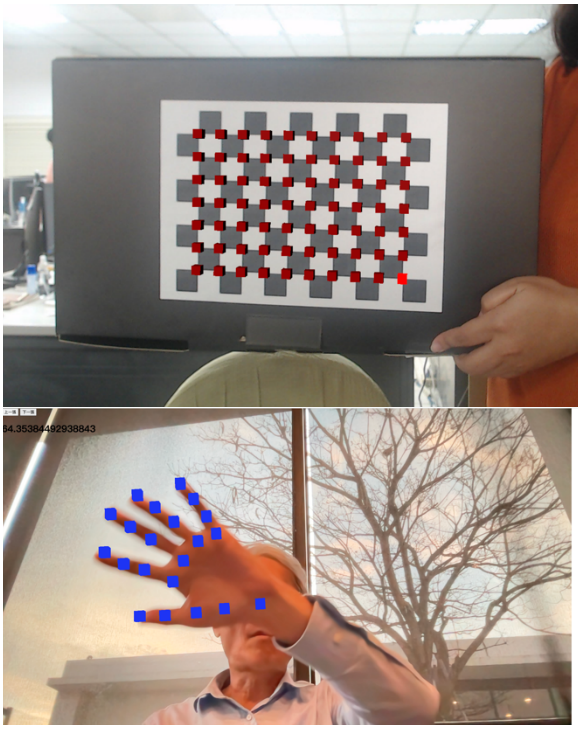


Figure 3. Place a cube at each intersection point after calibration

We also tested the camera calibration method in different environments. As shown in Figure 4, when the ambient light is dim, the virtual cubes displayed in three.js exhibit positional deviations. This indicates that when the camera uses color or brightness-sensitive features during calibration, the strength and quality of the light can affect the accuracy of the calibration.

For this reason, the current solution involves converting the image to grayscale, then calculating the total sum of pixel values to estimate the average light intensity. Following this, each RGB channel is normalized based on this light intensity, and the normalized values are constrained within the valid range (0 to 255). Finally, these channels are merged back together to produce the adjusted image.

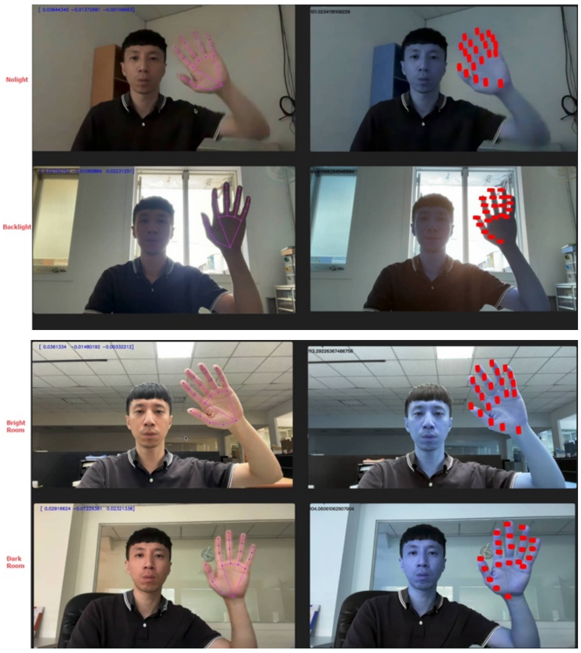


Figure 4. Different environment to place a cube at each intersection point

5. Implementations

After obtaining the calibration parameters of the camera, we can easily integrate virtual objects with images. Our current implementation primarily focuses on wearable items for the head and hands. For the head, we can simulate hats, glasses, and earrings, while for the hands, we have implemented rings and watches. To address occlusion issues when wearing these items, we have incorporated a generic model of a person's head. This virtual head model serves three purposes: first, to determine the alignment between virtual objects and a real person's head; second, to establish the relative positioning between wearable items and the person; and third, to address occlusion relationships between wearable items and the person's head.

To achieve efficient inference on mobile devices or web applications, we have adopted the Mediapipe solution. Within this framework, the entire process, from image preprocessing, head tracking, landmark detection, to the inference of the three-dimensional position of the head, is structured as a set of calculators. These calculators can be organized into a graph pipeline, and the results are encapsulated as WebAssembly for utilization in the front-end of web applications. We have established a WebGL environment and employed three.js for rendering.

The head itself is a rigid body, with facial landmarks using the nose as the local origin. Apart from facial expressions, there aren't many degrees of freedom. Therefore, simulations for trying on hats, glasses, and earrings on the head are relatively straightforward. Obtaining the transformation for try-on points, whether in translation or rotation, poses minimal challenges, as shown in Figures 5, 8, and 9.

Also, the detectable range starts from the camera and extends up to 2 meters for accurately recognizing a complete face, which is considered the normal tracking range. From 2 to 4 meters, the stability of detection gradually decreases. Beyond 4 meters, tracking will fail.

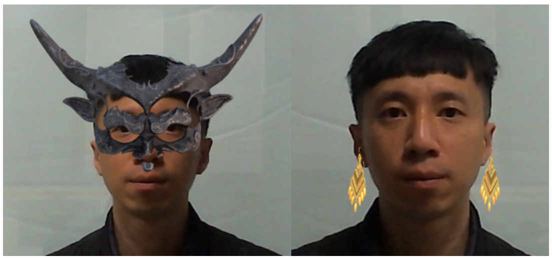


Figure 5. Trying on face masks and earring

In addition, we have incorporated physics simulation into the earring try-on application, sequence of images as shown in Figures 6.

We created an earring model using bones of a 3D model, associating various parts of the earring with the bone to control its deformation and movement. Next, rigidbody components were added to the earring model to impart physical properties such as gravity and collision in Figure 7. Finally, constraints were applied to restrict the movement of the earring model in the physics simulation, such as adding rotational constraints to mimic realistic swaying effects. In this physics simulation, the skeleton is used to define the shape and structure of the earring. The rigidbody represents the physical properties of the earring, such as mass, inertia, and collision attributes. Constraints are used to define the relationship and constraints between the earring and the head or other objects, such as how the earring should connect to the head or whether the earring can rotate. This application of physics simulation enhances the realism and immersion of the try-on experience, allowing users to better experience the appearance and feel of the earrings in a real environment.

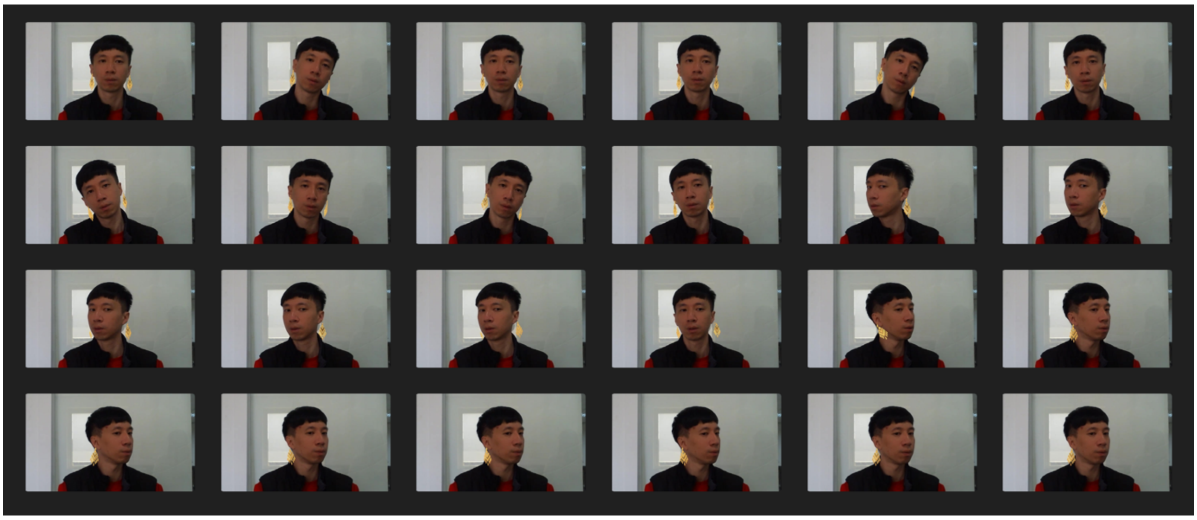
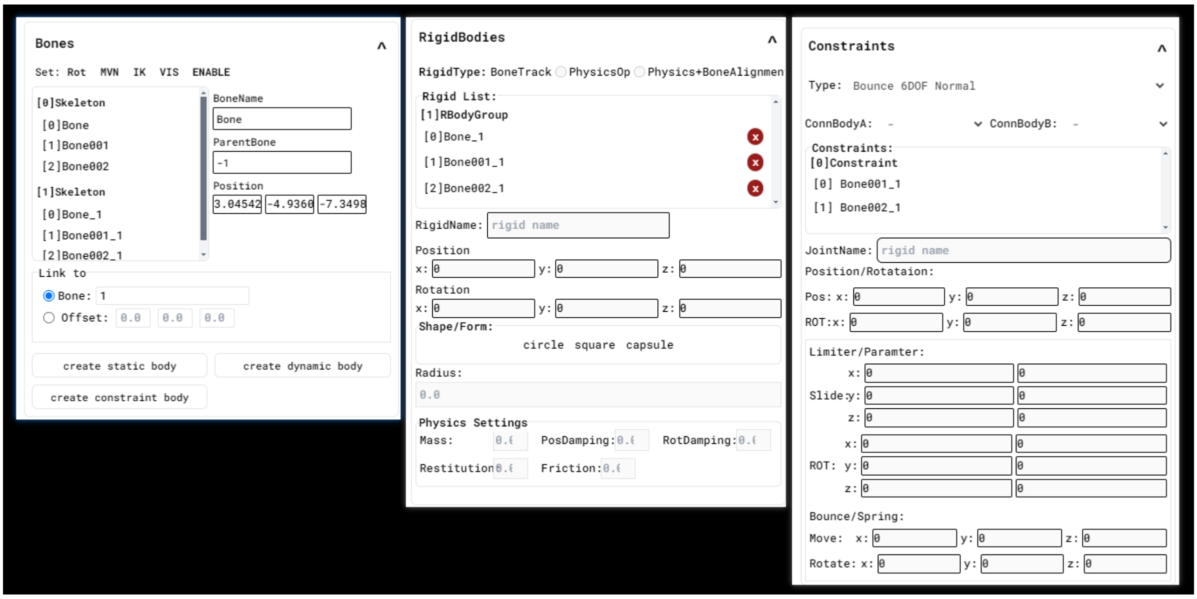


Figure 6. Earring physics simulation



Bones

[0]Skeleton
[0]Bone
[1]Bone001
[2]Bone002
[1]Skeleton
[0]Bone_1
[1]Bone001_1
[2]Bone002_1

BoneName: Bone
ParentBone: -1
Position: 3.04542 -4.9360 -7.3498

Link to
● Bone: 1
○ Offset: 0.0 0.0 0.0

RigidBodies

RigidType: BoneTrack ○ PhysicsOp ○ Physics+BoneAlignmen

Rigid List:
[1]RBodyGroup
[0]Bone_1
[1]Bone001_1
[2]Bone002_1

RigidName: rigid name
Position x: 0 y: 0 z: 0
Rotation x: 0 y: 0 z: 0
Shape/Form:
Radius: 0.0

Physics Settings
Mass: 0.(PosDamping: 0.(RotDamping: 0.(
Restitution 0.(Friction: 0.(

Constraints

Type: Bounce 6DOF Normal
ConnBodyA: - ConnBodyB: -

Constraints:
[0]Constraint
[0] Bone001_1
[1] Bone002_1

JointName: rigid name
Position/Rotataion:
Pos: x: 0 y: 0 z: 0
ROT:x: 0 y: 0 z: 0

Limiter/Paramter:
Slide x: 0 0
Slide y: 0 0
Slide z: 0 0
ROT: x: 0 0
ROT: y: 0 0
ROT: z: 0 0

Bounce/Spring:
Move: x: 0 y: 0 z: 0
Rotate: x: 0 y: 0 z: 0

Figure 7. Physics configuration

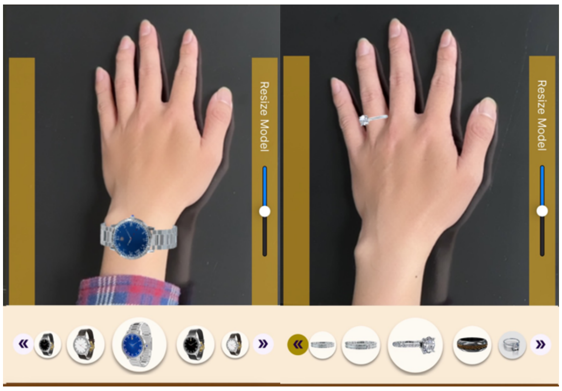


Figure 8. Trying on watches and rings

However, when it comes to trying on accessories on the hands, the presence of numerous joints introduces increased degrees of freedom. For example, in the case of trying on a ring, we not only need to acquire the position of the metacarpophalangeal (MCP) joint but also determine its rotation. The approach involves obtaining the positions of the wrist, index finger MCP joint, and pinky MCP joint—considering these three landmarks as representing the palm and treating them as a plane. Subsequently, we calculate the rotation of the ring finger relative to this plane. Calculating the angle between two vectors can be done using the arccosine (inverse cosine) function, which is based on the law of cosines. As illustrated in Figure 8, this method demonstrates the result of trying on a ring.

The hand landmarks do not include the wrist part, making trying on a watch quite challenging. To determine the fitting point of the watch and its rotation angle, we segmented the skin area using the Mediapipe Selfie Segmentation solution. Next, we applied Zhang et al.'s thinning method in OpenCV to obtain skeletons, allowing us to find suitable points as try-on candidate points. The search starts from the unique position of hand landmark 0, exploring skeleton points based on their neighborhood relationships. An ideal try-on point must satisfy two conditions: being distant from other landmarks (i.e., not in the direction of the palm and fingers) and having sufficient distance. By considering these two conditions, we can identify the most suitable point. The identified dorsal part of the hand, its skeleton, and the final try-on point are illustrated in Figure 9. Once this point is found, we extract a 3D point from the pose skeleton, whose projection onto the 2D image closely matches our identified 2D try-on point. The orientation of the watch face must be perpendicular to the tangent direction of the skeleton and closest to the normal vector of the back of the hand. The concept is illustrated in Figure 10.

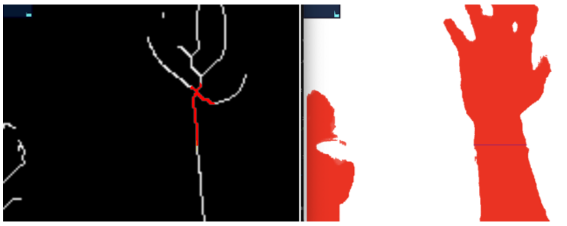


Figure 9. Locating the try-on point for watches from the hand's skeleton

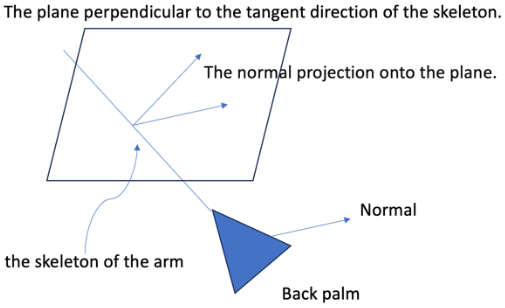

The plane perpendicular to the tangent direction of the skeleton.
The normal projection onto the plane.
Normal
the skeleton of the arm
Back palm

Figure 10. The normal vector of the back of the hand projected onto the plane perpendicular to the skeleton.

6. Result

To ensure consistency between the try-on size and the real size, we create virtual objects in proportion to their real-world counterparts. When trying on virtual objects, we can also wear real-world objects of known size to verify the size consistency between the real and virtual. As shown in Figure 11, we first try on virtual glasses and then wear real glasses on top. We can observe that the sizes of both are perfectly consistent.

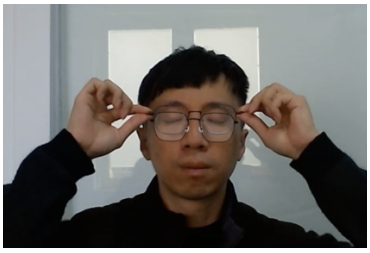


Figure 11. Comparing the size difference between real and virtual glasses

In Figure 12, we demonstrate trying on hats of different sizes. As shown in the display, we can observe the visual differences between wearing a smaller hat and a larger hat. By capturing two images at the same position, we can see that the lower edge of the larger hat appears larger than that of the smaller hat.

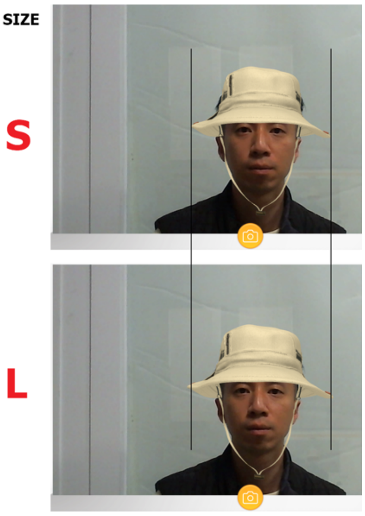


Figure 12. Comparing synthesized hats of different sizes

The current system still has some imperfections that can be improved. For example, when the user's head rotation angle is relatively large, as shown in Figure 13, the occlusion effect on the hair and nose is not well-executed, resulting in some flaws being visible. To address this issue, we plan to adopt recent advancements in image generation techniques. For instance, when generating portraits, Wang et al. [Wan23] faced challenges with hair, and they addressed this by employing UNet to refine the final results. However, UNet can be computationally intensive. While Jia et al. [Jia23]demonstrated its application on mobile devices, it might not be entirely suitable for real-time synthesis. We are currently working on how to achieve real-time performance for this solution.

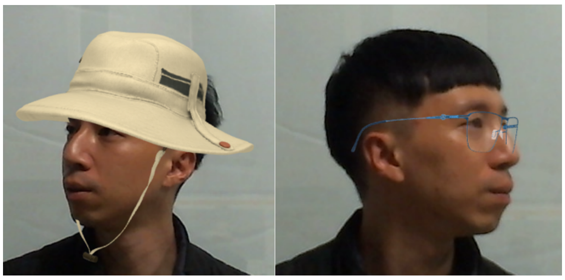


Figure 13. Imperfections during head rotation

Since our system is implemented in the web frontend, it is compatible with various operating systems such as iOS, Android, and Windows, and browsers like Chrome or Safari. Through testing on individuals with different body shapes, both males and females, our system consistently provides a realistic sense of size when interacting with the camera.

7. Conclusion

In the context of augmented reality applications, a better understanding of the real world leads to more accurate synthesis of virtual objects that match the real-world environment. In this paper, we propose a straightforward method for calibrating the user's camera, allowing for consistent results in terms of synthesized and real object sizes, regardless of their dimensions.

Furthermore, we demonstrate how to configure the virtual camera's projection matrix after obtaining the camera's intrinsic parameters. This ensures a consistent performance between the real and virtual cameras, enhancing the natural appearance of synthesized objects in the real environment.

When it comes to determining reference points for synthesized objects, we present an efficient method for reverse engineering 3D positions and orientations from 2D image reference points. Leveraging proximity relationships from the segmentation's skeleton, we locate the necessary points and find suitable solutions in 3D through vector projection.

Also, since virtual try-on will involve issues of portrait rights and privacy, to avoid controversy over user data being transmitted to the server, we also use MobileNet for training the AI model, allowing the AI model to run on the client-side, and it can even be used offline.

Lastly, we integrate a virtual try-on system, showcasing headwear trials for hats, glasses, and earrings, as well as hand trials for rings and watches. We demonstrate trial results of different sizes and compare them with real-size trials. Cross-testing on various devices and users validates the system's ability to provide a realistic sense of size in the try-on results.

8. ACKNOWLEDGMENTS

This work was supported by the National Science and Technology Council, Taiwan, R.O.C., under grant no. NSTC 112-2221-E-126-003.

9. REFERENCES

[Yan14] Yang, Y.-I., Yang, C.-K. and Chu, C.-H. A virtual try-on system in augmented reality using RGB-D cameras for footwear personalization. *Journal of Manufacturing Systems*, 33, 4 (2014), 690-698.

[Aze16] Azevedo, P., Dos Santos, T. O. and De Aguiar, E. *An augmented reality virtual glasses try-on system*. IEEE, City, 2016.

[Yu23] Yu, R., Wang, J., Ma, S., Huang, S. X., Krishnan, G. and Wu, Y. *Be Real in Scale: Swing for True Scale in Dual Camera Mode*. IEEE, City, 2023.

[Hei97] Heikkila, J. and Silvén, O. *A four-step camera calibration procedure with implicit image correction*. IEEE, City, 1997.

[Stu99] Sturm, P. F. and Maybank, S. J. *On plane-based camera calibration: A general algorithm, singularities, applications*. IEEE, City, 1999.

[Zha00] Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 11 (2000), 1330-1334.

[Mei07] Mei, C. and Rives, P. *Single view point omnidirectional camera calibration from planar grids*. IEEE, City, 2007.

[Ken15] Kendall, A., Grimes, M. and Cipolla, R. *Posenet: A convolutional network for real-time 6-dof camera relocalization*. City, 2015.

[Bog18] Bogdan, O., Eckstein, V., Rameau, F. and Bazin, J.-C. *DeepCalib: A deep learning approach for automatic intrinsic calibration of wide field-of-view cameras*. City, 2018.

[Lee21] Lee, J., Go, H., Lee, H., Cho, S., Sung, M. and Kim, J. *Ctrl-c: Camera calibration transformer with line-classification*. City, 2021.

[Pon22] Ponimatkin, G., Labbé, Y., Russell, B., Aubry, M. and Sivic, J. *Focal length and object pose estimation via render and compare*. City, 2022.

[Jin23] Jin, L., Zhang, J., Hold-Geoffroy, Y., Wang, O., Blackburn-Matzen, K., Sticha, M. and Fouhey, D. F. *Perspective Fields for Single Image Camera Calibration*. City, 2023.

[Ber22] Bermejo, E., Fernandez-Blanco, E., Valsecchi, A., Mesejo, P., Ibáñez, O. and Imaizumi, K. FacialSCDnet: A deep learning approach for the estimation of subject-to-camera distance in facial photographs. *Expert Systems with Applications*, 210 (2022), 118457.

[Che21] Chen, X., Liu, Y., Ma, C., Chang, J., Wang, H., Chen, T., Guo, X., Wan, P. and Zheng, W. *Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration*. City, 2021.

[Che22] Chen, X., Liu, Y., Dong, Y., Zhang, X., Ma, C., Xiong, Y., Zhang, Y. and Guo, X. *Mobrecon:*

Mobile-friendly hand mesh reconstruction from monocular image. City, 2022.

[Rom22] Romero, J., Tzionas, D. and Black, M. J. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610* (2022).

[Cos19] Costa, C. M., Veiga, G., Sousa, A., Rocha, L., Sousa, A. A., Rodrigues, R. and Thomas, U. *Modeling of video projectors in OpenGL for implementing a spatial augmented reality teaching system for assembly operations*. IEEE, City, 2019.

[Wan23] Wang, L., Zhao, X., Sun, J., Zhang, Y., Zhang, H., Yu, T. and Liu, Y. StyleAvatar: Real-time Photo-realistic Portrait Avatar from a Single Video. *arXiv preprint arXiv:2305.00942* (2023).

[Jia23] Jia, H., Wang, Q., Tov, O., Zhao, Y., Deng, F., Wang, L., Chang, C.-L., Hou, T. and Grundmann, M. *BlazeStyleGAN: A Real-Time On-Device StyleGAN*. City, 2023.

Genetic Subdivision Curve and Surface Reconstruction

Alexander Komar
a.komar@cgv.tugraz.at

Marco Riegler
marco.riegler@student.tugraz.at

Reinhold Preiner
r.preiner@cgv.tugraz.at

Ursula Augsdörfer
u.augsdorfer@cgv.tugraz.at

Institute of Computer Graphics and Knowledge Visualisation, Graz University of Technology, Austria

ABSTRACT

In this paper we employ a new genetic algorithm approach for CAD shape reconstruction, where a mathematical shape representation is reconstructed from point data. We reconstruct planar subdivision curves and 3D subdivision meshes from ordered input point data by fitting the corresponding subdivision control polygon or control mesh respectively, from which the smooth subdivision limit surfaces can be derived. For the reconstruction of curves the system estimates the number and position of control points required to approximate the curve closely. To reconstruct subdivision surfaces from points, the system determines a sequence of CAD operations which is subject to mutation in the course of a genetic optimization. We discuss implementation details of the proposed genetic algorithms and demonstrate our approach on a number of example data.

Keywords

Genetic Algorithms, Curve Reconstruction, Surface Reconstruction, Shape Code, Procedural Modelling

1 INTRODUCTION

To recover the shape of a curve or surface from data points is a challenging problem that appears frequently in a wide range of applications such as Computer-Aided Design (CAD), virtual reality and computer graphics, data visualization and medical imaging. Shape reconstruction in CAD refers to the process of generating a smooth and accurate representation of a curve or surface from a set of data points. The goal is to transform discrete data into a smooth and continuous curve representation suitable for CAD modeling. Shape reconstruction has become a fundamental tool in reverse engineering, where dense data acquired from physical objects is converted to a digital representation [6].

In this context B-splines [2] are the preferred approximating functions due to their powerful mathematical properties and their wide support by CAD/CAM systems. The B-spline curve or surface approximates the control polygon given by linearly connected control points P_i. They are defined over a uniform or non-uniform knot sequence. The control points and the knots are the design freedoms to satisfy approximation requirements. If the degree is fixed and the knot vector is fixed and uniform an ordered set of control points fully define the smooth shape.

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.

Smooth curves may be derived from control polygons either through the analytic expression of a B-spline curve or recursively through subdivision [12]. For arbitrary control meshes smooth subdivision surfaces are derived via subdivision refinement rules [1], where regular regions of the surface will correspond to B-spline surfaces.

The reconstruction of subdivision control meshes is a well researched problem [17, 23, 24, 10]. Traditional methods often use rather complex, multivariate optimization processes to reverse the subdivision algorithm. However, the main limitation of this class of algorithms is that the resulting control meshes usually do not exhibit a topology and distribution of extraordinary vertices that would emerge if the model was manually designed using a CAD system. This makes it more difficult to continue the design on the reconstructed shape as it enters a design pipeline.

In this paper, we approach the subdivision shape reconstruction problem by formulating it as a genetic optimization process. Genetic algorithms are a popular tool for a variety of search and optimization problems as they offer the possibility to search through multiple solutions for a given problem in parallel. They are based on various natural principles found in the real world, such as evolution, natural selection and reproduction. In this paper we introduce a genetic approach for B-spline or subdivision curve reconstruction and another genetic approach to reconstruct subdivision surfaces.

https://www.doi.org/10.24132/CSRN.3401.34

In our work we use a genetic optimisation approach to not only reconstruct the mathematical representation of the shape, but also a set of CAD operations that define a modelling process to construct said representation. We discuss the algorithm for 2D reconstruction in Section 3, the method for 3D reconstruction in Section 4. We present our results in Section 5 and propose future work in Section 6.

2 RELATED WORK

Our work relates to two major classes of methods: Genetic Algorithms and Evolutionary Optimization.

2.1 Subdivision Curves and Surfaces

The principle idea of subdivision is to define a smooth curve or surface as the limit of an iterative subdivision process in which an initial coarse control polygon or mesh is repeatedly refined by introducing new vertices in each subdivision step. Careful selection of the refinement rules ensures that the initial coarse control polygon or mesh converges to a smooth limit curve or surface, referred to as the subdivision limit curve or surface. That is, a subdivision curve or surface is fully defined by its control cage and a subdivision algorithm. The concept of subdivision curves was introduced by Chaikin [12]. Soon after, this approach was extended to surfaces. The concept of subdivision surfaces was introduced by Catmull and Clark [1] and Doo and Sabin [5] independently in 1978. Catmull-Clark subdivision is most commonly found in a wide range of CAD systems and is a standard representation in the entertainment industry. It is a generalization of uniform bi-cubic B-splines to arbitrary shapes and is based on quadrilateral meshes, like the Doo-Sabin algorithm. Generalizations to other domains were proposed by Loop [18] and Kobbelt [14] amongst others. Numerous different subdivision algorithms have emerged [19] based on a range of different types of control polyhedrons.

In this paper we propose a method able to automate the generation of control polygons and meshes for subdivision algorithms based on quadrilateral meshes which resemble manually created subdivision models. Throughout the paper we derive the corresponding limit curves using Chaikin [12] or uniform B-splines, and derive subdivision surfaces using the Catmull-Clark [1] subdivision.

2.2 Genetic Algorithms

There are many different types of evolutionary optimization, showing the potential and diversity in this area of optimization algorithms. In order to choose the best suited algorithm one has to look at different properties of the algorithms, for example their suitability for integer optimization or how they handle local optima. The fitness function, defined as a measure of the feasibility of a found solution, needs to be carefully designed as it guides the optimization to the global minimum.

Genetic Algorithms belong to the group of evolutionary algorithms, but differ in the reproduction process and gene encoding, e.g. genetic algorithms rely more on recombination than mutation for producing offspring, while evolutionary programming in contrast strongly relies on mutation. Evolutionary strategies use real-valued encoding for their genes [31].

Genetic algorithms were first introduced in 1970 by John Holland. They are employed to search and optimization problems where a large amount of variables are to be considered. They are based on evolution and natural genetics [27], modelling nature's principle of natural selection or "survival of the fittest". The measure of fitness varies with each application and is an important aspect in any evolutionary algorithm. While the weak are more prone to go extinct, the fitter individuals will be able to pass on their genetic information via reproduction [27]. As in nature, the offspring in genetic algorithms are also susceptible to mutation, which means that small parts of the inherited genes differ from their parents due to copying errors for example. We refer the reader to a comprehensive review on genetic algorithms and evolutionary optimization by Katoch et al. [13].

We chose genetic algorithms for our approach, since our goal was to not only be able to reconstruct the mathematical representation of the shape, but also define a set of CAD modelling operations which, applied to a template mesh, will yield the smooth shape which fits the data points.

2.3 Evolutionary Algorithms in Shape Reconstruction

Evolutionary approaches have been applied to curve and surface reconstruction using various types of algorithms.

Takeuchi et al. [28] proposed an algorithm able to reconstruct B-Spline surfaces from dense triangle meshes using Quadric Error Measures and elaborate ways of splitting triangle meshes into quad meshes. Xiyu et.al. [32] introduced a combination of genetic algorithms with Neural Networks to reconstruct 3D shapes. They used an encoding based on the principle genes, cells and organs. Combination operations were introduced to combine genes to cells and cells to organs or jelly. Galvez et al. [9] proposed genetic algorithms to reconstruct B-spline surfaces by first obtaining a parameterization for the surface and then determine the knot vectors and then calculating the control point positions according to the least squares approach. Galvez et al. improved their method to be able to execute in just one step [8]. They changed the optimization algorithm from a genetic approach to a particle swarm approach. They combined their previous 2 algorithms into one and solved the surface fitting problem by means of Least Squares. Sabsch et.al. [22] proposed an implementation of NSGA-II [3] to approximate different datasets with open B-spline curves of degree 3 with a fixed, clamped and uniform knot vector. The researchers optimized the number of control point positions and their position. Robustness to noise was shown, but not to sparsity or intersections. Wang et al. [30] used the Loop subdivision scheme in order to reconstruct cavities of the human body from volumetric data, gathered from different imaging modalities (MRI and CT). Rahamathulla and Misro [21] used genetic algorithms to aid in reconstruction of a craniofacial fracture. They propose an algorithm adjoining two curves together with a degree 5 B-spline curve with different continuity constraints. The number of control points was fixed. The input curves were extracted from a scan. Moulaeifard et al. [20] used Particle Swarm Optimization in order to reconstruct a geological model, which is watertight and controllable by a control mesh. They targeted the specific use case of geological simulations, therefor a low number of control vertices is desirable. Komar and Augsdörfer [15] used a Particle Swarm Optimization to approximate data using B-splines and rational B-splines. Similar to the approach presented here, they optimised control point positions to fit a B-spline curve to input data using a evolutionary approach different from the one explored in this paper.

In the following two sections we introduce a genetic algorithm which outputs a 2D vector of control point positions that define a clamped, uniform B-spline curve optimised to represent the input data. We also propose a genetic algorithm for surface reconstruction that takes a point cloud as an input and outputs a Catmull-Clark subdivision control mesh. In both cases, the genetic algorithms output a sequence of CAD operations to an initial template control polygon or mesh in order to create a CAD model of the control polygon or control mesh that approximates the input.

3 2D CURVE RECONSTRUCTION

To reconstruct subdivision curves the algorithm receives as input a two-dimensional array, containing x and y values of points along a planar curve. Each individual member of the population describes a control polygon. The control polygon is an ordered array of 2D control points, which defines a B-spline. The proposed algorithm also provides a list of operations, the chromosomes, that describe which operations were applied to an initial template polygon to arrive at the control polygon of the curve which optimally approximates the input data. In case of curves, the operations are limited to adding or deleting control points or translating existing control points.

The general structure of the genetic algorithm is based upon the algorithm proposed by Holland [11]. A genetic algorithm maintains a list of individuals, called *population*. The characteristics of each member of the population is defined by an individual *chromosome*, that corresponds to a solution to the given problem. Each individual is then ranked based on an objective value, its *fitness*, describing how well the proposed solution performs. In each generation, pairs of individuals are selected according to their fitness to produce offspring, whose properties are defined by a combination of its parents chromosomes, for the next round of evolution until a predefined termination criterion is met [29].

In our work each individual is represented as a sequence of CAD operations, which applied to the starting control polygon in order, gives the solution proposed by the individual. The initial population in genetic algorithms hugely influences its performance, that is a well conditioned starting population comes with a higher chance of finding a satisfying solution. One way to influence the quality of the first generation of individuals is the population size. In general, using too few individuals leads to worse solutions, while having a large population leads to higher computation time [4]. In our work we employ a mixed strategy: The algorithm is given a large initial population, thus increasing the initial diversity. During selection the population is then reduced to only 1/10th of the initial population, encouraging more diversity in the starting solutions. As a result, the first evolution round takes more time. To create the population, the algorithm creates 1000 individuals, all containing four to eight random CAD operations as genes in their chromosome. Each individual therefore represents a polygon which evolves to a control polygon of a uniform B-spline curve approximating the input data.

Each individual created is initialized with four control points. The first and last control point are placed on the first and last points of the input curve, as the curve generated by this algorithm is clamped. The two remaining control points then consist of the swapped x and y coordinates of the endpoints. The bounding box of the input curve is used to restrict the area in which control points can be moved or added. The quality of each individual is assessed by its fitness function.

The chromosome of each individual is a list of genes, which are the set of CAD operations applied to an initial template polygon. These operations can either be the addition of a new point to the current polygon or the translation of a point. If a new point is introduced, the index of the new point is added in the control point array together with its position. If the point is translated, its position is updated. The first and last control points are barred from being translated. Since they lie on the end points of the input point set.

The fitness function determines how well a candidate solution performs and is an important part in the selection mechanism. In the proposed algorithm this function takes an individual as input, derives the B-spline or subdivision curve using its control point positions and calculates the Euclidean distance between the samples lying on the reconstructed curve and the samples on the target curve, calculated index wise. We sample the curve to have the same number of points as the target curve, therefore the distance calculation can be carried out index-wise. The fitness value is then the sum over all euclidean distances. After calculating the fitness value for each member of the population, the algorithm can begin with the reproduction of the individuals for the next generation.

We use the roulette wheel selection to select parents for reproduction [13]. Each individual is given a weight based on its fitness value. After two parents have been selected, the algorithm computes a crossover which produces offspring by combining their genes. In our work single-point crossover [26] is used to combine two individuals to their offspring. Different length parents can also be combined, by choosing the crossover point according to the shorter length.

In the mutation operation, the algorithm loops over the chromosome and, for each gene, draws a random mutation probability value. After comparing different mutation probability values, a constant probability of 5% has been shown to perform well. If a gene is chosen to be mutated, the algorithm chooses with a probability of 1% to delete the current gene. Otherwise, the gene is mutated by replacing it with a randomly generated operation. After looping over all genes of the individual, another check is performed if a new operation should be appended to the list. This is also done, if a chromosome is left with less than three genes after too many deletions, as this would result in errors during a crossover operation.

The last step of the mutation function is to check whether all operations in the mutated individual are still valid. The validity might not be given if an operation that adds a control point is removed, as a move operation might access the control point later in the chromosome. The algorithm iterates through the chromosome, and checks whether each operation can be executed. If not, the operation is removed from the chromosome. Lastly, it also checks the length of the chromosome again, to ensure the required minimum length is still given, otherwise it would add as many random operations as needed to make sure crossover can be executed.

In this work the *steady-state* technique was used [25]. After the population is sorted according to the fitness value of each individual, the top 20% are moved into the next generation. The remaining 80% open spots are filled with offspring, created using the reproduction steps described above.

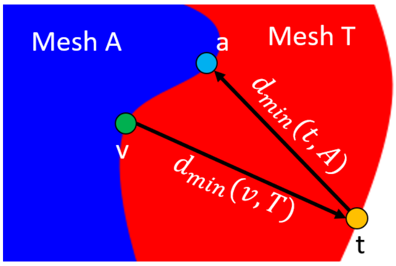


Figure 1: Illustration of the distance function from a single vertex a to the target mesh T. $d_{min}(x,M)$ represents the minimum distance from a vertex x to any vertex on a mesh M.

The steps above constitute the evolution of one generation. This process is repeated for as many generations as desired, or until a certain termination criterion is met. One typical termination criteria used is that one candidate solution satisfies a specified minimum threshold of the fitness value [16], which we employed in the optimisation presented in this paper.

4 3D SHAPE RECONSTRUCTION

Surface reconstruction closely follows the approach employed in genetic curve reconstruction. However, the design of 3D shapes on a CAD system involves numerous operations. We demonstrate our approach considering seven CAD operations defined in Blender [7], five of which are shown in Figure 2. However, our approach may be extended to more types of operations. The algorithm then derives control meshes by combining a set of modelling operations in a genetic approach. The set of CAD modelling operations, when applied in sequence to an initial template mesh, yield a subdivision control mesh which defines a smooth surface, the reconstruction solution. The resulting meshes, see Figures 5 and 6, exhibit features which are typically found subdivision surface which has been designed on a CAD system.

A good fitness function defines the quality criteria of the surface reconstruction. We employ the mixed mean distance between the limit surface corresponding to the found control mesh and the input sample point which belong to a target surface.

The first step in the evaluation of the fitness is the calculation of the mixed mean distance between the meshes, which is illustrated in Figure 1. For every vertex v on the mesh computed by the algorithm (Mesh A) the following is calculated. First, the closest vertex t by Euclidean distance on the target mesh (Mesh T) is found. Then, starting from the vertex t the closest vertex a by Euclidean distance on mesh A is found. The relevant part for the mixed mean distance is the Euclidean distance d_{min} from vertex v to vertex t. Finally, if vertex

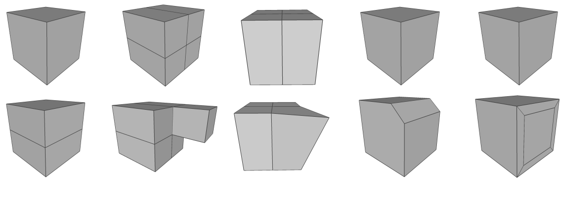

(a) [L] (b) [E] (c) [V] (d) [B] (e) [I]

Figure 2: Five of the seven operations used in our method. (a) Loop insertion, which introduces a new edge in the middle of an edge which is continued through the model, (b) face extrusion along the face normal, (c) vertex translation in a random direction, (d) bevelling of an edge and (e) face inset.

v and vertex a are not the same, the distance is set to the bounding box diagonal. The mean of this distance is calculated over all vertices of mesh A to mesh T and vice versa. The mixed mean distance is then the maximum of the two and the fitness is the inverse of the mixed mean distance.

The quality of the fitness function is essential. A simple mean distance measure proved insufficient to capture the error between meshes well and did not converge. One of the problems was overlapping regions in one of the shapes, which caused the simple mean distance to be very small. Various other distance measures could be used here, but need to be evaluated carefully. We observed that a Hausdorff distance, for example, is not expressive enough for this kind of optimization. It only captures the maximum distance and therefore, does not consider all parts of the mesh. This leads to small adjustments being discarded.

Our genome is a design code, comprised of a set of modelling operations applied to an initial template mesh. This template mesh may be any quadrilateral mesh and can be chosen by the user. We demonstrate our approach using a unit cube as an initial template mesh.

Evolving the sequence of modelling operations rather than the explicit shape representation itself leads to a procedural genome that reflects closely the variety of results produced by a design process using these modelling operations. To this end, we use seven types of operations widely used in CAD modelling. Five of these operations are shown in Figure 2:

Op: V An arbitrary vertex is translated. Parameters are the vertex and the vector of translation. The vector of translation is defined by three random samples chosen in the interval $[-bbox_d, bbox_d]$, where $bbox_d$ is the length of the bounding box diagonal of the mesh.

Op: E A face is extruded along its normal. Parameters are the face and the distance of extrusion. The maximum distance of extrusion is limited by the length of the bounding box diagonal.

Op: L An edge loop is inserted. Here, the parameter is a single edge. The edge loop operation is applied to the middle of an arbitrary edge but results in a insertion of a number of edges, each splitting along the middle edge of all edge in a mesh which are opposite to each other (Figure 2(a)). It is the only operation which affects the whole mesh. When the edge loop meets a non-quadrilateral face it terminates, creating a T vertex, that is added to the non-quadrilateral face.

Op: B Bevelling creates an angled face between two adjacent faces. Parameters are the edge between two faces and the length to be cut off. Applying the bevel operation on an edge with an extraordinary vertex, an additional extraordinary vertex with the same valence is created.

Op: I Face inset creates another face inside an existing face and connects the new vertices to the vertices of the outer face. Parameters are the face and length of inset.

Op: F Face translate displaces a face along its normal. No new vertices are added. Parameters are the face and the offset of the movement.

Op: S Scale operation scales the whole mesh by a factor. Parameter is the scaling factor.

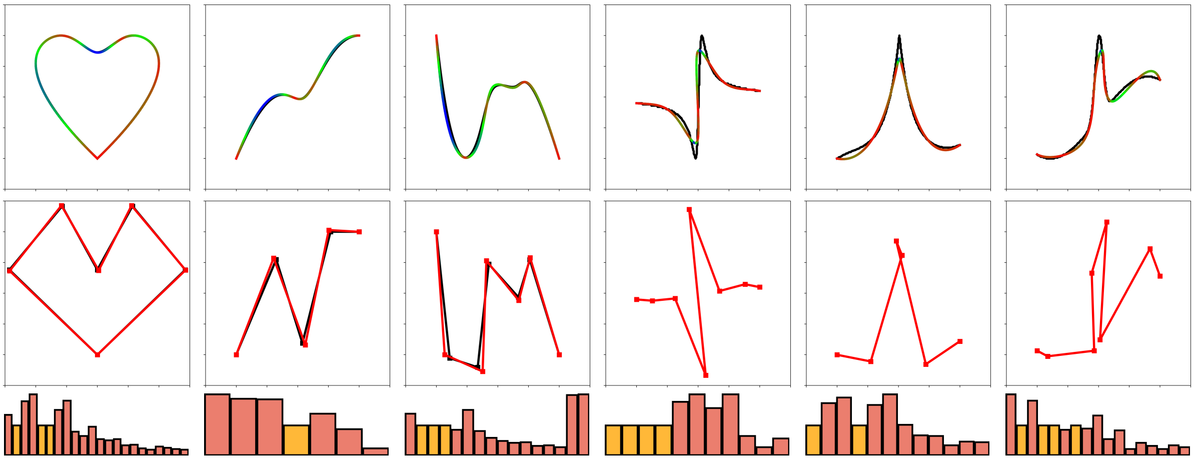
The genome is initialized with one or two random modelling operations which are sequentially applied to arbitrary parts of a mesh. Each operation builds on its predecessors and can change geometry newly introduced by previous operations.

5 RESULTS

Figures 3 and 4 show results derived by our genetic approach. To demonstrate our approach we reconstruct cubic B-spline curves from ordered points sampled from cubic B-spline curves, Figures 3(a) - (c), non-parametric curves, Figures 3(d) - (f), noisy data, Figure 4a, and data sampled at non-equidistant points along a B-spline curve, Figure 4b.

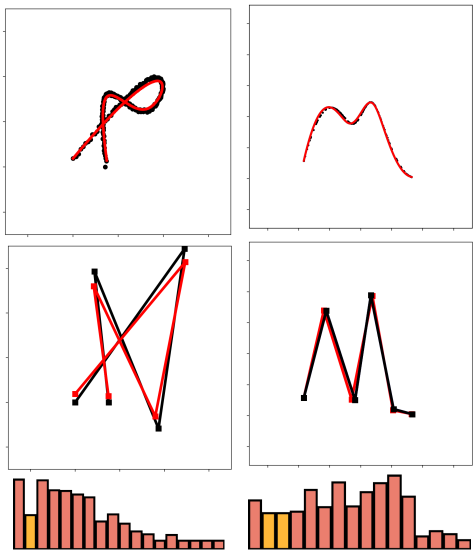
The top row of Figures 3 and 4 show the reconstructed curve (red or colored) plotted on top of the target B-spline curve (black). The colors of the reconstructed curve indicate how close it is to the target curve, scaled by the maximum distance measured. The color red corresponds to a smaller distance, whereas blue describes a larger distance.

The center row in Figures 3 and 4 shows a comparison between the control polygon used to create the target curve (black) and the control points generated by the



(a) Heart (open curve, $n = 7$) (b) Random curve ($n = 5$) (c) Random curve ($n = 7$) (d) Non-parametric curve f1 from [15] (e) Non-parametric curve f3 from [15] (f) Non-parametric curve f4 from [15]

Figure 3: Curves reconstructed using our algorithm. Top: target data points (black) and derived B-spline curve colored by the Euclidean distance to the target curve (red = low, blue = high). Center: result control polygon (red) overlaid on the control polygon of the reference B-spline curve sampled by the target points (black). Bottom: Visualization of the resulting chromosome as a sequence of genes. Yellow bar: vertex insertion operation. Red bar: vertex movement (bar height indicating movement distance).



(a) Noisy input (b) Sparse input

Figure 4: (a) Result after adding Gaussian noise ($\sigma = 0.7\%$ of the bounding box diagonal) to the target samples (black). (b) Reconstruction from sparse data, after randomly removing 75% of the point samples. Top: Result curve (red) overlaid on target points (black). Middle: Comparison of control polygons. Bottom: Chromosome visualization as in Figure 3.

algorithm (red). Because curves 3(d) - (f) are not derived from B-splines, a comparison is not possible.

As evident from these examples, the algorithm derives the same number of control points as was used to create the target curve and matches their positions closely, without any prior knowledge of the control polygon structure. Since the algorithm is not restricted on how many points are to be used in the solution, we expect curves to also be derived by more or less control points than the original target curve.

The bottom row in Figures 3 and 4 depicts the chromosome of each solution, consisting of operations that are applied to generate the control polygons shown below of each of them. The operations are applied to the template mesh in sequence starting from left. The last operation is on the very right. A vertex insertion is indicated by a yellow bar, while a vertex translation is shown in red. The size of the bar indicates the distance of the translation. We observe that translation operations applied to vertices at a later stage the modelling process tends to decrease. This is not explicitly encoded in the algorithm and shows that the algorithm tries to first move the control points reasonably close and then looks more closely to refine the position.

In Figure 4a the data sampled for curves has been altered by introducing noise to each data point. While the algorithm found control points close to those from which the target curve has been defined, they are not as precise as without noise. The average distance between our reconstructions from the noisy data points and the original curves, have an average Euclidean distance of 0.075.

To test the performance of our algorithm with sparse data points sampled from the target curve, we reduced the sampled points by 75% by randomly deleting data points. Results are displayed in Figure 4b. In the reconstruction the algorithm created control points similar to the ones used in the original input. The example results in an average Euclidean distance of around 0.02 measured from the generated to the original curve.

Although noisy data effects our algorithm more than sparse data, it is quite robust to both.

The parameters used in the calculation of the results of our 3D reconstruction shown on top in Figure 5 for the genetic algorithm were 100 individuals, 200 generations and 20 parents which were combined to generate up to 10 children. The operations of the 10 best individuals were mutated specifically. 10 newly created individuals were added per generation and a limit of 3 individuals with the same genome was enforced. For the bottom row of Figure 5 the parameters were the same with the exception of the number of iterations which was 1000 and the limit of individuals with the same genome which was 10.

Figure 5 show results of the algorithm performing the fitment of a straightforward control mesh to a limit surface. It has no information about the control mesh which was used to generate the target and compare their resulting limit surfaces to the given target. The coloring indicates the relative distance from the point on the generated mesh to the target. The red color indicates a small distance to the target. The closer the color is to blue the further the target mesh is away. In Figure 5 top we observed distances from 0 to 0.001 and a mean of 0.0001 for the best individual of the genetic algorithm. Execution time was 240s. In Figure 5 bottom we observed distances from 0 to 0.005 and a mean of 0.002 for the best individual of the genetic algorithm. Execution time was about 1400 seconds. The distances were normalized by the bounding box diagonal. The control meshes show that our algorithm is able to identify the specific operations needed to generate to target control polygon.

In Figure 6 we test our genetic approach with slightly more challenging reconstruction problems. The shape on top was specifically modelled to enforce the algorithm to reconstruct exact sets of operations. While the best individual in the first generation has an extrusion without a prior face inset, by the 50th generation the correct set was identified and only positions were refined afterwards. The shape on the bottom was produced using a random set of operations. After 1000 generations our method was able to find the combination of operations very similar to the ones used to create the mesh. However, an extra edge loop was inserted in the top part, which, after 1000 generations remained in the shape. This demonstrates a limitation of the genetic reconstruction approach.

6 CONCLUSION

In this paper we introduce a genetic approach to the reconstruction of B-spline or subdivision curves and subdivision surfaces. The genetic algorithm takes a list of data points as input and evolves a control polygon or control mesh by applying a set of operations. The subdivision curves or surfaces corresponding to the derived control polygon or mesh approximates the input data closely. The algorithm derives the number of control points required as well as their position to optimise the approximation to the input data.

We proposed a genetic algorithm based on a modelling process, which is novel in the context of reconstruction surfaces or curves.

For the case of planar curve reconstruction our proposed method using two operations showed some promising results in reconstruction. The algorithm handles noise and sparsity well, although it struggles with more complex shapes like in Figure 3 (f), where it does tend to get stuck in local optima.

For the reconstruction of highly complex curves we suggest a pre-processing step in form of a curve segmentation algorithm at high curvature region to split up the reconstruction problem. Additionally, to speed up reconstruction of highly complex curves, future work will include extending this algorithm by a pre-processing step which derives an approximate minimum to the number of control points required to reconstruct the curve.

The 3D case is not a straightforward extension of the 2D case and requires a larger set of operations. Although results were impressive for the examples given in this paper, the algorithm struggles with convergence for more complex shapes. Future work will focus on solving the convergence problem by employing a more sophisticated approach, e.g. using NSGA-II [3], and multiple fitness functions, constraining the algorithm to converge faster.

REFERENCES

[1] Catmull, E., Clark, J., 1978. Recursively generated b-spline surfaces on arbitrary topological meshes. Computer-aided design 10, 350–355.

[2] De Boor, C., Rice, J.R., 1968. Least squares cubic spline approximation i-fixed knots. International Mathematical and Statistical Libraries .

[3] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation 6, 182–197.

[4] Diaz-Gomez, P.A., Hougen, D.F., 2007. Initial population for genetic algorithms: A metric approach., in: Gem, Citeseer. pp. 43–49.

[5] Doo, D., 1978. A subdivision algorithm for smoothing down irregularly shaped polyhederons. Computer Aided Design , 157–165.

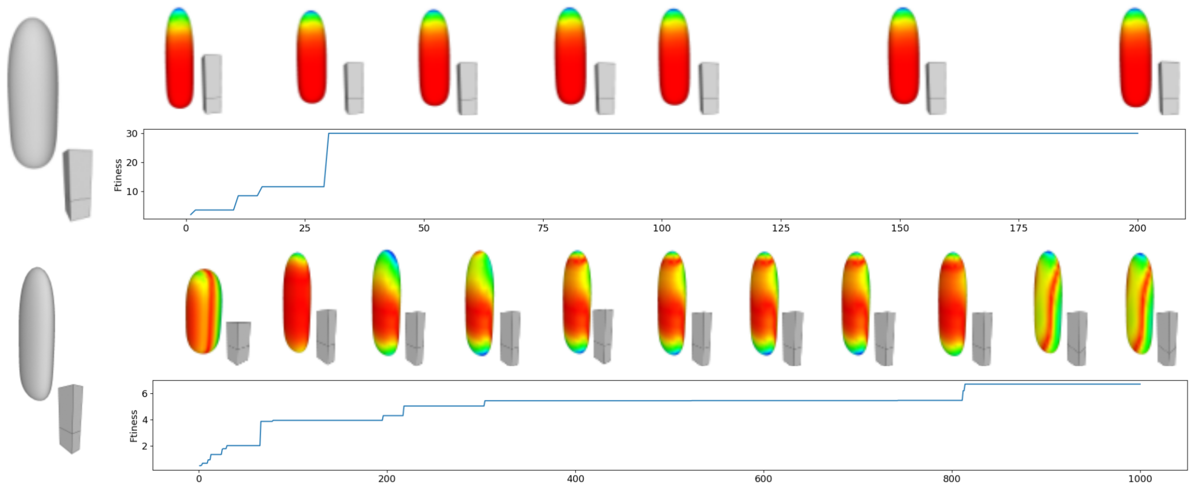


Figure 5: Comparison on the convergence behaviour of the algorithm using two examples. Top: cube extended in one direction. Bottom: cube flattened and then extended. For each, the fitness of the best solution is plotted as a function of generations. Results above the fitness graph show the best individual of the generation together with its limit surface. Limit surfaces are colored by the distance of a point on the mesh to the target mesh (red closer, blue further away). Since the the flattened shape needs very specific operations to be determined it takes many more generations as for the top shape.

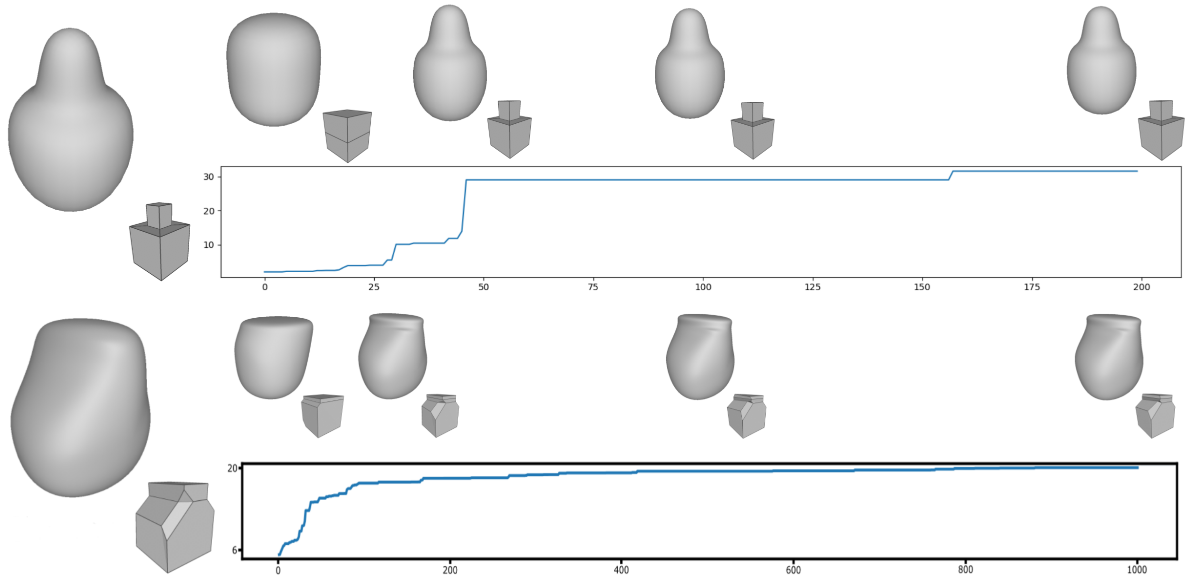


Figure 6: Two examples more complex examples of freeform geometry, generated using a random set of operations. Fitness and solutions are shown as in Figure 5.

[6] Farin, G.E., 2002. Curves and surfaces for CAGD: a practical guide. Morgan Kaufmann.

[7] Foundation, B., 2022. Blender python api (version 3.0.1). URL: `https://docs.blender.org/api/current/index.html`.

[8] Gálvez, A., Iglesias, A., 2012. Particle swarm optimization for non-uniform rational b-spline surface reconstruction from clouds of 3d data points. Information Sciences 192, 174–192.

[9] Gálvez, A., Iglesias, A., Puig-Pey, J., 2012. Iterative two-step genetic-algorithm-based method for efficient polynomial b-spline surface reconstruction. Information Sciences 182, 56–76.

[10] Hassan, M.F., Dodgson, N.A., 2005. Reverse subdivision, in: Advances in multiresolution for geometric modelling, Springer. pp. 271–283.

[11] Holland, J.H., 1976. Adaptation in natural and artificial systems.

[12] Joy, K.I., 1999. Chaikin's algorithms for curves. Visualization and Graphics Re .

[13] Katoch, S., Chauhan, S.S., Kumar, V., 2021. A review on genetic algorithm: past, present, and future. Multimedia tools and applications 80, 8091–8126.

[14] Kobbelt, L., 2000. Square root 3 subdivision, in: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 103–112. doi:`10.1145/344779.344835`.

[15] Komar, A., Augsdörfer, U., 2023. Swarmcurves: Evolutionary curve reconstruction, in: International Symposium on Visual Computing, Springer. pp. 343–354.

[16] Kumar, M., Husain, D.M., Upreti, N., Gupta, D., 2010. Genetic algorithm: Review and application. Available at SSRN 3529843 .

[17] Lanquetin, S., Neveu, M., 2006. Reverse catmull-clark subdivision .

[18] Loop, C., 1987. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics .

[19] Ma, W., 2005. Subdivision surfaces for cad - an overview. Computer-Aided Design 37, 693–709. doi:`10.1016/j.cad.2004.08.008`.

[20] Moulaeifard, M., Bernard, S., Wellmann, F., 2023. Pysubdiv 1.0: open-source geological modeling and reconstruction by non-manifold subdivision surfaces. Geoscientific Model Development 16, 3565–3579.

[21] Rahamathulla, N.H., Misro, M.Y., 2023. Curve fitting using genetic algorithm and its application in craniofacial reconstruction. Pertanika Journal of Science & Technology 31.

[22] Sabsch, T., Braune, C., Dockhorn, A., Kruse, R., 2017. Using a multiobjective genetic algorithm for curve approximation, in: 2017 IEEE symposium series on computational intelligence (SSCI), IEEE. pp. 1–6.

[23] Sadeghi, J., Samavati, F.F., 2011. Smooth reverse loop and catmull-clark subdivision. Graphical Models 73, 202–217. doi:`10.1016/j.gmod.2011.03.004`.

[24] Samavati, F., Pakdel, H.R., Smith, C., Prusinkiewicz, P., 2003. Reverse loop subdivision .

[25] Sastry, K., Goldberg, D., Kendall, G., 2005. Genetic algorithms. Search methodologies: Introductory tutorials in optimization and decision support techniques , 97–125.

[26] Soon, G.K., Guan, T.T., On, C.K., Alfred, R., Anthony, P., 2013. A comparison on the performance of crossover techniques in video game, in: 2013 IEEE international conference on control system, computing and engineering, IEEE. pp. 493–498.

[27] Srinivas, M., Patnaik, L., 1994. Genetic algorithms: a survey. Computer 27, 17–26. doi:`10.1109/2.294849`.

[28] Takeuchi, S., Kanai, T., Suzuki, H., Shimada, K., Kimura, F., 2000. Subdivision surface fitting with qem-based mesh simplification and reconstruction of approximated b-spline surfaces, in: Proceedings the Eighth Pacific Conference on Computer Graphics and Applications, IEEE. pp. 202–212.

[29] Thengade, A., Dondal, R., 2012. Genetic algorithm–survey paper, in: MPGI national multi conference, Citeseer. pp. 7–8.

[30] Wang, X., Ang, K.D., Samavati, F.F., 2021. 4d surface mesh reconstruction from segmented cardiac images using subdivision surfaces, in: Proceedings of the 8th International Conference on Bioinformatics Research and Applications, pp. 56–62.

[31] Whitley, D., Rana, S., Dzubera, J., Mathias, K.E., 1996. Evaluating evolutionary algorithms. Artificial intelligence 85, 245–276. doi:`https://doi.org/10.1016/0004-3702(95)00124-7`.

[32] Xiyu, L., Mingxi, T., Hamilton Frazer, J., 2003. Shape reconstruction by genetic algorithms and artificial neural networks. Engineering Computations 20, 129–151.

LatEd: A Geometric Latent Vector Editor

Alexander Komar
a.komar@cgv.tugraz.at

Michael Kammerer
michael.kammerer@student.tugraz.at

Saeedeh Barzegar Khalilsaraei
s.barzegar@cgv.tugraz.at

Ursula Augsdörfer
u.augsdorfer@cgv.tugraz.at

Institute of Computer Graphics and Knowledge Visualisation, Graz University of Technology, Austria

ABSTRACT

Using a neural network approach, a shape may be compressed to a one-dimensional vector, the so-called latent dimension or latent vector. This latent shape dimension is examined in this paper.

This latent vector of a shape is used to identify the corresponding shape in a database. Two types of networks are evaluated in terms of lookup accuracy and reconstruction quality using a database of Lego pieces. Even with small training set a reasonable robustness to rotation and translation of the shapes was achieved.

While a human can interpret uncompressed data just fine, the compressed values of the network might be cryptic and thus offer no insight regarding the uncompressed input. Therefore, we introduce a latent dimension editor which allows the user to examine the geometry content of the latent vector and its influence on the decoded shape. The latent vector editor enables the visual exploration of the latent vector, by making changes to the latent vector visible in real-time via a 3D visualization of the reconstructed object.

Keywords
Neural Networks, GAN, Signed Distance Fields

1 INTRODUCTION

Extracting the defining features of a geometry allows us to represent a 3D object more efficiently, e.g. one could describe a Lego piece by precisely defining its geometrical shape, even though the same information can be condensed into its structural attributes, such as size and stud count.

In this paper we use neural networks to extract condensed information to a concise descriptor with a lower dimension than the original representation of the provided object.

Autoencoders [1], a subclass of neural networks (NN), are used to learn an efficient encoding of their input data in terms of latent dimensions in compact form. The autoencoder first encodes the input into a lower dimensional latent representation. The outcome of the compression is then decoded to reconstruct the original input.

The latent representation is a central part of this paper. The latent shape descriptor is useful in many applications, e.g. for storing large amounts of data efficiently due to its dimension and size.

To assess the applicability of using the latent vector in context for 3D object retrieval (3DOR) we derive the database lookup accuracy for a range of Lego pieces. Using a Lego database [12] as an example, we demonstrate that a fast and accurate retrieval of an arbitrary 3D shape can be achieved.

To visually examine the latent vector and its geometry encoding, we present a real-time latent vector editor. By visualizing the reconstructed 3D objects such that alteration to the shape latent dimension becomes visible in real-time, the user gets immediate feedback on how any changes applied to the latent vector affect the reconstruction and, thus, the encoding of the geometry. This feature opens up a world of exciting experiments in which the user gains insight into how the latent vector relates to the uncompressed input geometry. While in this paper we focus on two shape representations, namely a voxel or signed distance field (SDF) representation the approach may be extended to other shape representations.

2 RELATED WORK

Neural networks in combination with 3D voxel grids and SDF have been used frequently in context of machine learning geometry.

Voxel grids have been employed e.g. in surface reconstruction by Brock et al. [3]. Their generative convolu-

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.

https://www.doi.org/10.24132/CSRN.3401.35

tional autoencoders achieved a good reconstruction for 3D voxel grids, and discriminative convolutional neural networks achieved a robust classification of the latent dimension values. Wang et al. [16] proposed a network which uses voxel and SDF data of an object as its inputs. It processes the given data using a convolutional neural network, autoencoders, and an extreme learning machine. The primary purpose of said configuration is for 3D shape classification tasks. Maturana and Scherer [13] proposed a classification network, that uses a voxel occupancy grid as an input. They also showed some robustness to rotation. Wu et al. [18] proposed a method to recognize different shapes as voxel grids from a 2.5D input image. They applied their method to real world data.

Wang et al. [15] propose a new way of storing information in a Oct-tree and using it as an input for a Neural Network. They show their method on object classification, shape retrieval and shape segmentation.

Wohlhart et al. [17], trains a convolutional neural network (CNN) to perform object recognition and 3D pose estimation by computing the similarity of different descriptors through their Euclidean distance and employing nearest neighbor search to handle large amounts of objects.

The method proposed by Girdhar et al. [8] works by combining two encoder networks to produce one latent vector and one decoder network reconstructing a voxelized representations of the shape. In the testing phase the shape is reconstructed from the image alone. They further analyzed the latent vector for meaningful feature values.

DeepSDF [14] focuses on an auto-decoder neural network learning a continuous signed distance function representation of a class of shapes for shape representation, interpolation, and completion, even from partial and noisy 3D input data. It is a generative model that produces a continuous signed distance field given, for example, a depth map.

Chen et al. [5] proposed their implicit field decoder, called IM-NET to auto-encode 3D meshes by training it to a certain category of meshes. They also proposed a method to generate and interpolate between meshes using their Network setup. Further, the researchers proposed using 2D images as an input in order to generate and interpolate 2D shapes and reconstruct 3D shapes from a single image (Single-view 3D reconstruction).

The compressed latent vector has frequently been proposed as a tool to edit the geometry and often used in generative models such as Variational Autoencoders (VAEs) or Generative Adversarial Networks (GANs), see e.g. [4, 6, 10].

In this paper, we examine two autoencoder networks and explore their latent dimension of geometry for use in several tasks. The latent vectors are used for database queries and shape reconstruction. The look-up accuracy achieved demonstrates that the latent vector indeed is a well compressed representation of the shape. To visually explore the compressed latent shape representation further we introduce a latent editor, which enables the user to modify the latent dimension and visualize its effect on the decompressed geometry real-time, thus enabling the user to visually inspect the latent vector wrt geometry changes.

3 METHOD

We use deep neural networks (DNNs) [19] to encode and decode 3D geometry. Such networks consist of multiple layers of nodes (neurons), each having one or numerous in- and outputs. Each neuron generates an output by summing up its weighted inputs and bias and then evaluating an activation function. The activation function describes when the neuron "fires" and can be applied on a layer-per-layer basis.

We combine multiple such layers, each with its individual assigned activation functions, to build a network that can be split into two halves: The first half encodes a given input to a latent dimension. By forcing the network to trade information for a lower dimension, it must extract features representing the input well enough for the compression to be viable. The second half takes this representation and expands it into a reconstruction of the input. The error (loss) between input and output is used to adjust the weights and biases of the network such that the reconstruction quality improves with each epoch.

An implicit representation can depict arbitrary geometry at a fixed size, thus fulfilling the fixed input-output constraints of any network. In this paper, we use an implicit representation where each voxel of the input and output data contains a floating point value that indicates whether the voxel is outside ($+1.0$), inside (-1.0), or directly on the surface ($+0.0$) of the volume.

This representation may be expanded to contain the actual distance of the cell center to the closest object's surface, essentially becoming a signed distance field (SDF). In this paper we assess whether this additional information increases the accuracy of the networks prediction. For the networks, the aim was to obtain a 1D latent vector which could be further employed in a latent modelling interface.

We compare the results of two types of autoencoder networks, a densely connected NN and a large CNN:

The first network we trained and tested features various densely connected LeakyReLU and batch normalization layers and its structure is shown in Figure 1. LeakyReLU was chosen due to its fast computation time [9] and to enable negative float values to pass

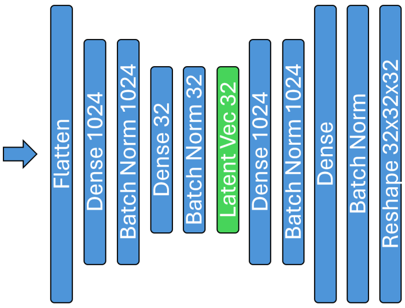


Figure 1: Structure of the first network.

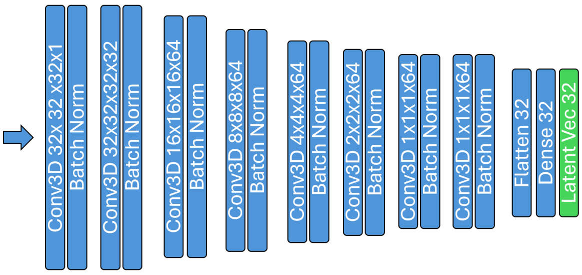


Figure 2: Structure of the second network. Decoder part is mirrowed from the encoder part and hidden due to limited space.

through the network. Batch normalization layers are employed to avoid vanishing gradients [7]. The encoder half of this network consists of four layers. First a densely connected layer reducing the size from 32^3 to 32^2, followed by a batch normalization layer, then a densely connected layer reducing the size to 32, followed by a batch normalization layer. The decoder half has the same structure as the encoder part, just mirrored.

The encoder of the second network is considerably deeper and consists of eight 3D convolution layers followed each by a batch normalization layer. The structure is visualized in Figure 2 The decoder consists of seven 3D convolution layers, again each followed by a batch normalization layer.

In our work, the input volumes are compressed into a 1D latent dimension vector containing 32 values. This value was found to be the lowest, where meaningful results could be achieved.

The network's input and output have the same dimensionality, size, and data type. We compute the reconstruction quality of the autoencoder network using a loss function based on the mean squared error between the implicit shape of the network's input and output.

Both networks are trained for 250 epochs using an 80/20 split for training and test data and use the Adam [11] as their optimizer. The networks are trained and tested with cuDNN-enabled Python using TensorFlow and the Keras API. cppflow was utilized to port the trained networks to C++ to visualize the output.

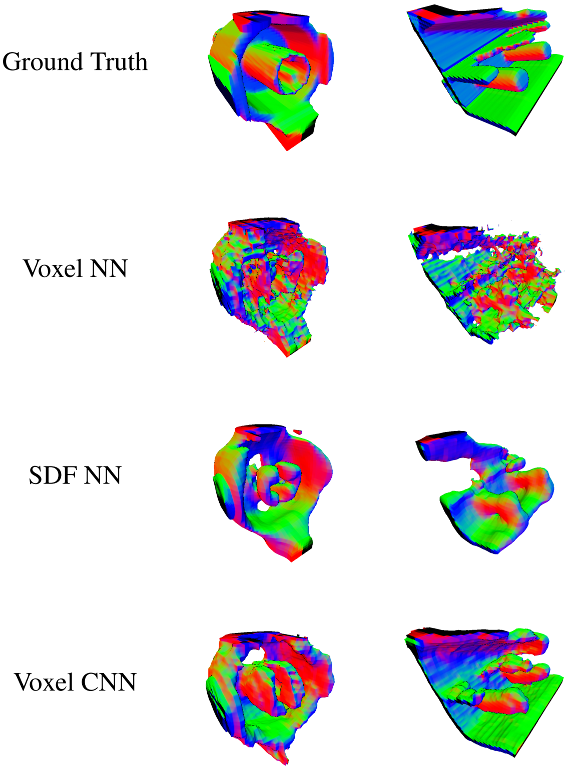


Figure 3: Examples of 3D shape reconstruction using two different shape representations (voxel or SDF) and two different NNs (a dense NN or a CNN).

To train and test the networks we gathered data from an online library of Lego meshes called LDraw [12]. The meshes of the Lego pieces are converted the CAD representation of the Lego pieces into their voxel and SDF representation. To gather sufficient data takes time, we therefore randomly rotate (max. ±180°), translate (max. 10%), and scale (max. 10%) the converted shapes to expand the data set from 40 to 4000 shapes, each consisting of 32^3 cells. Finally, the dataset is normalized, such that all values lie in the interval $[-1, 1]$.

4 RESULTS

We explored the results of the two autoencoder networks, by comparing the quality of their 3D reconstruction. We also test the applicability of the latent code in various tasks, like 3D object retrieval (3DOR), realtime latent editing, and morphing shapes.

4.1 3D Reconstruction

The decoder half of the network and the latent dimension vectors are loaded into the visualization framework, which calculates and displays the resulting volumes.

The reconstruction results in Figure 3 depict the reference and reconstructed volumes. The colors represent

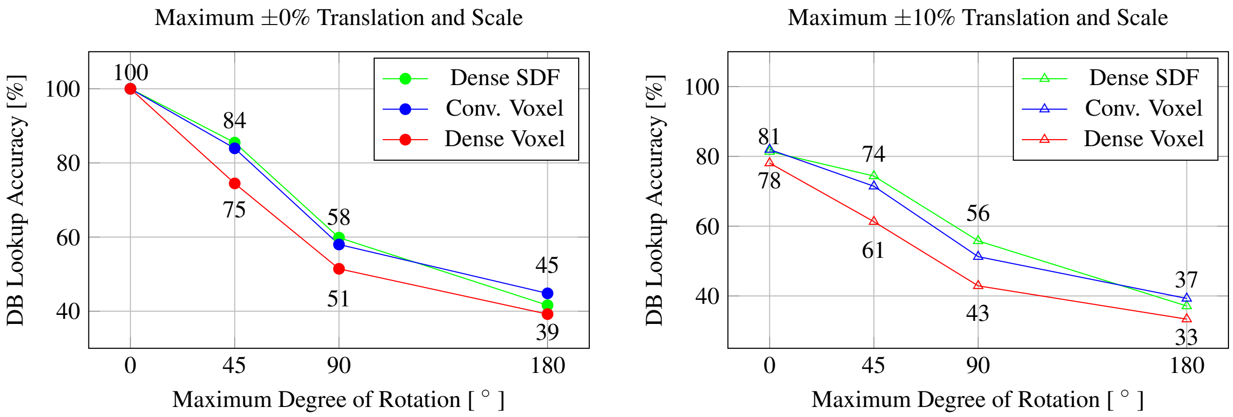


Figure 4: Database lookup accuracy of the dense and convolutional network with voxel and SDF input. The lookup accuracy decreases as the volumes are scaled, rotated, and translated.

the normal vectors of the object. The mean squared error of both depicted samples is below 0.1 for the SDF input. Since we can set negative values to -1.0 and positive values above zero to $+1.0$ as a post-processing step of the voxel output, we can simplify our measure of reconstruction error by simply counting the cells which have an incorrect sign. The percentage of cells with an incorrect sign of the two depicted samples equals 17% (left) and 14% (right) for the dense network with voxel input and 15% and 12% for SDF input.

In general the convolutional network produces smoother surfaces with less noise. It also achieves higher reconstruction accuracy. Using voxels the CNN achieves an incorrect cell sign percentage of 14% and 8% for both samples, respectively. The improvement is even more significant when using SDF as shape representation.

4.2 Latent based Object Retrieval

We prepared the data base by compressing all shapes to their 1D latent vector. After also deriving the latent shape for the query shape, the corresponding shape was retrieved from a database of shapes by finding the closest match between the query latent vector and those in the data base. The shape was retrieved with high accuracy from the shape data base.

The database lookup accuracy is given by the percent of correctly identified input samples. The database lookup accuracy depicted in Figure 4 shows that the retrieval accuracy is reasonably high for quiery shapes which are rotated up to 45° rotation from the target shape but drops significantly afterward.

The dense autoencoder network, combined with SDF input data, yields the best results, with the convolutional network with voxel input data performing similarly. The dense network with voxel data performs significantly worse. Increasing the maximum possible translation and scale to ±10% reduces the accuracy by twenty percent on average when no rotation is applied and reduces with increasing rotation.

4.3 Real-time Latent Editor

To examine the latent dimension further we implemented an interactive latent modelling editor which enables direct modification of the latent vector by the user. The user can alter the latent dimension of an input geometry and see the affected shape changes in the decoded result in real-time. The user can choose from a collection of 3D volumes, choose from two input representations, analyze the reconstruction error, rotate and scale the 3D volumes, and alter or set individual latent dimension values.

Some interesting results obtained by modifying the latent dimension values can be seen in Figure 5. In this example, the latent dimension vector of a 2x1 Lego piece represented as voxels is edited and decoded by the trained CNN decoder network which was loaded into the visualization framework. The editor is flexible such that it allows to upload any trained en- or decoder in order to experiment with the latent shape code.

Using more expressive latent vectors with this method, shapes that do not exist in the database could be created. This has a lot of potential in a creative workflow and designers could benefit from this ability. Another potential use case could be creating a design process based on this. A basic model can be encoded in a meaningful latent vector and then edited to add more features or quickly change the look of the created object.

4.4 Latent Editor: Morphing Shapes

We extended the Latent Editor to enable the user to select a source and target volume. The morphing tool of the editor then defines new shapes in between both by slowly linearly interpolating between the latent dimension vector of the source volume and the latent dimension vector of the target volume.

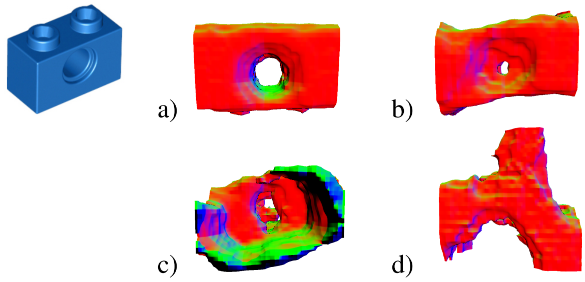


Figure 5: Examples of increasingly altering the latent dimension values of a Lego piece sample. Left: original Lego piece [2].

The result of this operation is the source volume slowly morphing into the shape of the target volume, as visible in Figure 6. This may be interesting in an entertainment context. But geometry which emerges during the morphing may also be interesting in a process, for example it could be used as inspiration for new shape designs.

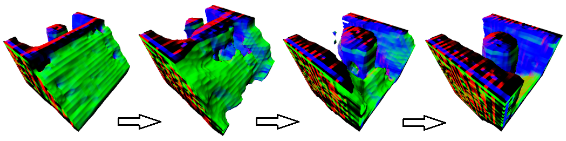


Figure 6: Morphing a Lego piece into another one via linear interpolation of the latent dimension values.

5 SUMMARY AND FUTURE WORK

In this paper we evaluate different shape representations and networks and their corresponding latent vectors in various scenarios. We evaluated the networks based on the quality of their 3D reconstruction and how accurate their corresponding latent vector succeed to retrieve 3D objects from a data base. We also evaluated two implicit shape representations, namely a simple voxel occupancy grid and a SDF. SDFs have performed significantly better than simple voxels when reconstructing shapes, as more information is contained in the input from which the network can learn. We explored the latent vector for two different types of autoencoder networks, namely a dense NN and a CNN.

While the latent vector derived using a CNN leads to a more accurate and smoother reconstruction of the original shape, a simple NN set may already suffice in using the latent code to retrieve a corresponding shape from a shape data base.

The shown results indicate that executing a database lookup favors a specialized network trained to do classification instead of using an autoencoder, as it is, in essence, a classification task. Future research will include developing and testing two separate networks, one for reconstruction and one for classification, and comparing the obtained results to the results described in this paper.

We also present an interactive tool that enables the user to interactively alter the latent vector and immediately explore the changes in the reconstruction. This tool can be used to explore the geometric meaning captured in a latent vector. It also enables the user to look into a potential meaning for the latent vector values and analyze each component on its own in real time. With interpolation between the database entries of different shapes a user can examine what different shapes lie between values and travel along the latent space.

Future research questions will focus on extending the capability of the editor and analyzing the latent dimensions with the proposed tool.

REFERENCES

[1] Bank, D., Koenigstein, N., Giryes, R., 2020. Autoencoders. URL: `https://arxiv.org/abs/2003.05991`, doi:`10.48550/ARXIV.2003.05991`.

[2] BrickLink, accessed 2024-01-02. `https://www.bricklink.com`.

[3] Brock, A., Lim, T., Ritchie, J.M., Weston, N., 2016. Generative and discriminative voxel modeling with convolutional neural networks. arXiv preprint arXiv:1608.04236 .

[4] Chen, M., Xie, J., Laina, I., 2023. Shap-editor: Instruction-guided latent 3D editing in seconds. arXiv preprint arXiv:2312.10825 .

[5] Chen, Z., Zhang, H., 2019. Learning implicit fields for generative shape modeling, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5939–5948.

[6] Elsner, T., Ibing, M., Czech, V., Nehring-Wirxel, J., Kobbelt, L., 2022. Intuitive shape editing in latent space. arXiv preprint arXiv:2111.12488 .

[7] Foster, D., 2019. Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play. First ed., O'Reilly Media, Inc.

[8] Girdhar, R., Fouhey, D.F., Rodriguez, M., Gupta, A., 2016. Learning a predictable and generative vector representation for objects, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14, Springer. pp. 484–499.

[9] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. `http://www.deeplearningbook.org`.

[10] Hu, V.T., Zhang, D.W., Mettes, P., Tang, M., Zhao, D., Snoek, C.G.M., 2023. Latent space editing in transformer-based flow matching. arXiv preprint arXiv:2312.10825 .

[11] Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. URL: `https://arxiv.org/abs/1412.6980`, doi:`10.48550/ARXIV.1412.6980`.

[12] LDraw, accessed 2024-01-02. `https://www.ldraw.org/`.

[13] Maturana, D., Scherer, S., 2015. Voxnet: A 3d convolutional neural network for real-time object recognition, in: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE. pp. 922–928.

[14] Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S., 2019. Deepsdf: Learning continuous signed distance functions for shape representation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[15] Wang, P., Liu, Y., Guo, Y., Sun, C., O-cnn, X.T., . Octree-based convolutional neural networks for 3d shape analysis., 2017, 36. DOI: https://doi.org/10.1145/3072959.3073608 , 1–11.

[16] Wang, Y., Xie, Z., Xu, K., Dou, Y., Lei, Y., 2016. An efficient and effective convolutional auto-encoder extreme learning machine network for 3d feature learning. Neurocomputing 174, 988–998. URL: `https://www.sciencedirect.com/science/article/pii/S0925231215014940`, doi:`https://doi.org/10.1016/j.neucom.2015.10.035`.

[17] Wohlhart, P., Lepetit, V., 2015. Learning descriptors for object recognition and 3d pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[18] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3d shapenets: A deep representation for volumetric shapes, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1912–1920.

[19] Yun, K., Huyen, A., Lu, T., 2018. Deep neural networks for pattern recognition. URL: `https://arxiv.org/abs/1809.09645`, doi:`10.48550/ARXIV.1809.09645`.

Brain Computer Interfacing with a Virtual Environment

Florian Gamillscheg
florian.gamillscheg@student.tugraz.at
Institute of Computer Graphics and Knowledge Visualisation,
Graz University of Technology, Austria

Irena Ruprecht
irena.ruprecht@fraunhofer.at
Fraunhofer Austria, Graz, Austria

Volker Settgast
volker.settgast@fraunhofer.at
Fraunhofer Austria, Graz, Austria

Krzysztof Pietroszek
pietrosz@american.edu
American University, Washington D.C, USA

Ursula Augsdörfer
u.augsdorfer@cgv.tugraz.at
Graz University of Technology, Austria

ABSTRACT

Virtual Reality (VR) applications constantly strive for more realism, immersion and intuitive user experiences. Traditional VR controllers can hinder full immersion, since they form an additional barrier between the user's thoughts or intentions and the virtual world. Brain computer interfaces (BCIs) have the potential to close this gap by enabling an immediate translation of human thoughts to commands that can be processed by a computer. This paper investigates the feasibility of employing an affordable commercial BCI device for VR interaction. In a preliminary study conducted in a Cave Automatic Virtual Environment (CAVE), we evaluate both the effectiveness and limitations of the popular BCI device *Emotiv Insight*.

Keywords

Brain Computer Interface, Virtual Reality, CAVE

1 INTRODUCTION

In recent years interest and popularity in virtual reality (VR) devices has increased drastically and has attracted the attention of the research world and consumers alike [1]. Today, VR experiences are more realistic and engaging than ever. Different input devices allow users to engage with their virtual environment and play an important role in increasing immersion [2]. Unlike traditional computers, where the interaction usually involves mouse and keyboard, VR applications use alternative input devices and aim for a more intuitive control. Besides input methods like gesture [2], or head-movement [32], brain computer interfaces (BCI) offer a promising alternative as a communication tool. These systems work by reading the user's brain signals, applying machine-learning algorithms to classify the brain state, and then based on the signal, trigger an action on the computer [11].

BCIs have the advantage that no muscular movements are needed, instead systems are only controlled by thoughts or emotion, which is why BCIs are often discussed as an alternative input device to help users with motion impairments to interact with a computer [26]. More recently BCIs have been explored as an additional input channel in video games and VR [22].

For noninvasive acquisition of brain activity, electroencephalography (EEG) [29] has been the most widely used method, due to its relatively low cost, high resolution and portability [21]. EEGs measure electrical activity resulting from current flows, which are produced when neurons are activated. Signals are read by electrodes that are placed on different regions of the skull [31]. The recorded brain activity can be seen as waves that can be categorized based on their frequency [31]. The dominance of each wave type is heavily influenced by the user's current emotional state and where the electrodes are placed [31].

In recent years, consumer-grade EEGs have found their way from research into real-world environments [22]. Despite limitations compared to medical-grade devices including lower sensor counts, accuracy and transfer rates, commercial EEG devices are still a popular choice for immersive experiences due to their affordable price and ease of use [22].

This paper presents an exploratory study on the applicability of BCIs in conjunction with virtual environments, aiming to test the technical feasibility of utilizing consumer-level BCIs for this purpose in a practical setting. We conducted experiments with the commercial and affordable EEG headset *Emotiv Insight 2.0* (shown in Figure 1) in order to evaluate, whether cost-efficient BCIs can be used as an intuitive input device for VR applications. We chose to utilize Emotiv Insight for our experiments due to the minimal setup time of 1

https://www.doi.org/10.24132/CSRN.3401.36

Figure 1: *Emotiv Insight*. Image taken from [4].

- 2 minutes, the seamless integration within the Unity game engine through Emotiv's API, and it's popularity in research. Emotiv offers a Pro license for 149$ per month that offers high-resolution performance metrics with a higher sampling rate than the free license. However, the Pro license would not have made an impact on the overall quality of the device, which is why we opted for the free license. Choosing the free license over the Pro license also ensured alignment with our goal of a low-cost interaction method [5]. To assess its applicability as an input device for VR we used the Emotiv headset to trigger different animations in a Cave Automatic Virtual Environment (CAVE).

All experiments were performed in the CAVE, a four sided, projection-based VR setup with lightweight LCD shutter glasses which offer improved wearing comfort compared to head-mounted displays, and minimise obstructions and interference between VR and BCI headsets.

2 RELATED WORK

Brain Computer Interfaces (BCI) have enabled many different possibilities over the years. Traditionally, BCI focused on medical applications, developing applications to allow patients to control and communicate with things in all aspects of life [22]. Research in this field has undergone major advances over the years, with BCI being used as an assistive technology to enable patients with a paralyzed body to communicate [15], to controllable wheelchairs [9] and many other applications.

Recently, BCIs have been introduced in the consumer market and new applications have been developed in particular for the entertainment/gaming area [22]. In this area, combining BCI devices and VR has led to promising results [22]. Friedman *et al.* [8] showed that BCIs can be suitable to interact with virtual environments. In their work, they used the Graz-BCI [25] to carry out different experiments with three subjects in different VR setups. The authors came to the conclusion that BCI has potential as an input device in VR. They also found that accuracy was the highest in a CAVE environment, which was also the preferred environment of all subjects. Leeb *et al.* [19] aimed to overcome the use of cue-based BCI and get closer to real world conditions. Therefore, the VR application must allow the users to freely decide, what they want to do. For their study, they built a freely explorable apartment and asked 10 test subjects to walk to a defined target room using EEG signals from only three channels. They concluded that also with a simple EEG setup of only three channels, the users could successfully navigate through the apartment. The study also showed that motivated users performed better than unmotivated ones, which indicates that motivation is an important factor during BCI training.

VR setups are also an ideal test environment for BCI research and for scenarios that would be too costly or dangerous under normal circumstances [18, 17]. Leeb *et al.* [18] showed that combining VR and BCIs can be a useful tool to control a wheelchair in VR. Guger *et al.* [10] demonstrated in a VR setup that BCI devices can be successfully used to control smart home devices and applications, such as opening doors.

In recent years, multiple studies have indicated that Virtual Reality can convey feedback to BCI users better than simpler 2D approaches [19]. Ron-Angevin and Díaz-Estrella [27] found in their research that receiving feedback through an immersive experience can have a positive impact on BCI accuracy. In this study, users, who performed BCI tasks in VR, demonstrated reduced error rates and also reported a motivational effect compared to a traditional 2D screens. This effect may arise from the increased immersion and realism in virtual scenes provided by VR that can lead to more distinct brain patterns, which can be better recognised by BCI devices [23].

Due to advancements in technology, EEG devices have recently entered the consumer market [22] and several commercial EEG devices have emerged. Examples include Emotiv Epoc, Emotiv Insight, Neurosky Mindwave, & the OpenBCI headsets, which vary in price & the amount of sensors. In their review on the use of consumer-level BCIs in research, Sabio et al. showed, that Emotiv devices were the most extensively utilized, followed by Neurosky Mindwave [28]. Neurosky Mindwave offers only one electrode and one reference electrode next to the ear, which might be the reason for it's low accuracy, as indicated by multiple works [22]. Other BCIs such as OpenBCI have received limited attention in research [28]. Zabcikova [33] investigated the quality of *Insight*'s EEG signals with the help of subjects that were exposed to visual and auditory stimuli. Other application where *Emotiv Insight* has been used include drone control [20] and controllable smart wheelchairs [6]. In combination with a head-mounted display, Fayed *et al.* [7] used *Emotiv*

Insight to create a cognitive training program to train focus. A cognitive approach was also taken in [12], where Hu and Roberts used an *Emotiv Insight* device with a VR setup to study the correlation between the emotional state and the built environment characters.

In this paper we employ the *Emotiv Insight* device to control VR scenes which surround the user in a CAVE. We chose this specific device due to it's widespread adoption in research and the acceptable count of five sensors. Aiming for affordability, Emotiv Insight additionally emerges as a cost-effective alternative compared to higher-priced products like Emotiv Epoc.

3 METHOD

This section provides insight into the input device, the test environment, as well as the animations that can be triggered by the EEG signals. Our proposed BCI-VR setup allows users to experience a changing environment that responds to the user's thoughts. A schematic model of our system is illustrated in Figure 2. The user's thoughts or mental commands, are picked up by the *Emotiv Insight* device and are used as an input for the VR Scene depicted in the CAVE.

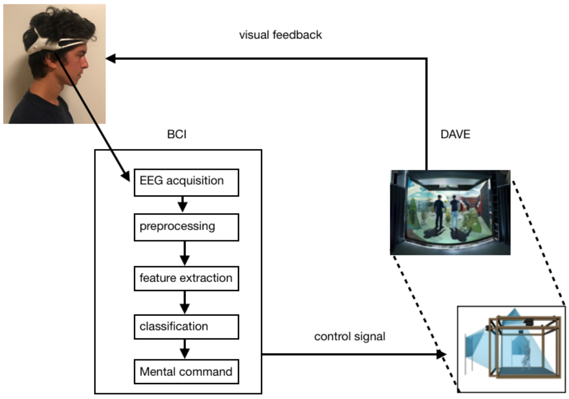


Figure 2: Schematic model of our *Emotiv Insight*-VR setup. EEG data is acquired by the Insight device, and processed to a *mental command*, which is subsequently used as an input for our VR system, which responds with visual feedback to the user.

3.1 The Virtual Environment

A CAVE offers an immersive experience and unlike head-mounted displays (HMD), also allows users to feel their presence in the virtual world [30]. In our case, the CAVE is a projection-based VR setup, which consists of four projection walls, four stereoscopic projectors and two standard hardware computers for each projection wall rendering the image for the left and the right eye respectively [16]. Additionally, users wear lightweight LCD shutter glasses, which can be comfortably worn together with the BCI. A schematic of our CAVE is shown in Figure 3.

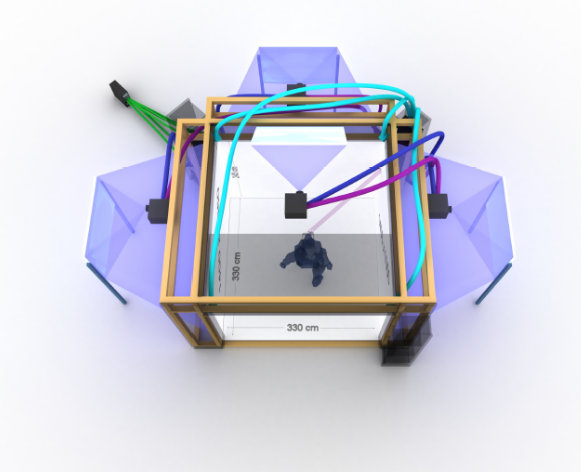


Figure 3: A schematic of the CAVE setup. Images are projected on the side walls from the back and on the floor from above. Mirrors are used to reduce spatial requirements.

3.2 The Brain Computer Interface

Emotiv Insight is a portable, consumer-grade EEG headset and is a cost efficient alternative to other EEG devices. The headset is designed for every day use and features 2 reference sensors and 5 channels which are located on the scalp according to the 10-20 electrode system [14] on positions AF3, AF4, T7, T8 and Pz. The connection between the computer that runs the CAVE application and the headset is established via Bluetooth. Communication with the headset is done with *Emotiv's Cortex API*, which is based on JSON and Websockets [3]. Additionally, the *Emotiv App* is also needed to authorize the application. The API can be used to obtain different data from the headset, namely *facial expressions, mental commands* and so called *performance metrics* that represent values of the user's current stress, relaxation, focus, excitement, interest and engagement levels. The data rate of the device is 128 Hz, which corresponds to 128 JSON messages per second for each opened data stream. Nevertheless, the actual data sampling rate heavily depends on the type of license. In this case, we used the free license, which allowed us to receive new *performance metrics* every 10 seconds. Since the use of such a device requires focus on the task, the time was used by applicants to really focus on the thought.

To trigger animations in VR we used *mental commands*. This functionality allows users to trigger events based on their thoughts and allowed us to receive data in real-time.Therefore, a profile must be trained with Emotiv's BCI software, which associates thoughts to specific events. Besides the *neutral* command, which represents a relaxed and calm state, this EEG headset is able to differentiate up to 4 different commands at a time, which are named as *push, pull, left, right*.

After sufficient training, we ensured that a good connection is established before we could use it in our CAVE application. For this, the Emotiv App indicates the EEG quality for each sensor with a color. Before usage, it is crucial that the headset fits firmly on the user's scalp and that the sensors are moisturized with contact lens fluid. Only if all sensors report a good EEG quality, high-quality *mental commands* can be received. In the API, *mental commands* come as string tuples that contain the *mental command's* name, the power, and the time. The current *mental command* and its power were then used, to trigger an animation. To sort out faulty and inaccurate *mental commands* that can be caused by bad EEG quality, we introduced a threshold on the power. The *mental command* was only used, if it exceeded the defined threshold. This allowed us to receive consistent data without the need of computationally intensive smoothing.

3.3 VR Animations

To assess the applicability of the commercial BCI input device for VR control, we used three animations which the user may control via a *mental command*. The experiments were carried out by a 22-year-old male participant. All animations where developed using Unity 3D.

3.3.1 Accelerating Sphere

In this animation, the user can accelerate a sphere and move it through hoops, Figure 4.

If the *mental command* is neutral, the sphere decelerates until it eventually comes to a standstill. If the user focuses on pushing the sphere through the hoops, the sphere accelerates and rolls through the hoops, much like a VR game of croquet.

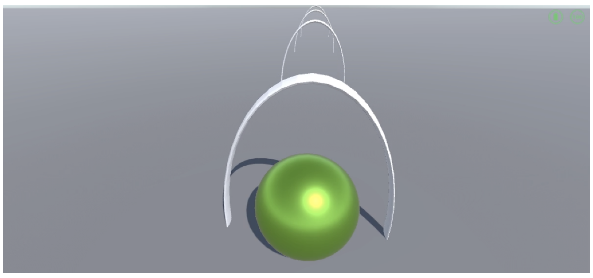


Figure 4: The sphere's acceleration is controlled by the power of the triggered *mental command*.

3.3.2 Terrain

A planar terrain, Figure 5, is modified by generating Perlin noise [24] using a users thoughts.

The hills increase in depth when the user starts to trigger a *mental command*. The longer the user holds on to the thought and the stronger the current command is picked up, the deeper the valleys and hills get. When users relaxes from the thought, the surface starts to get smoother, until it reaches a flat and even surface.

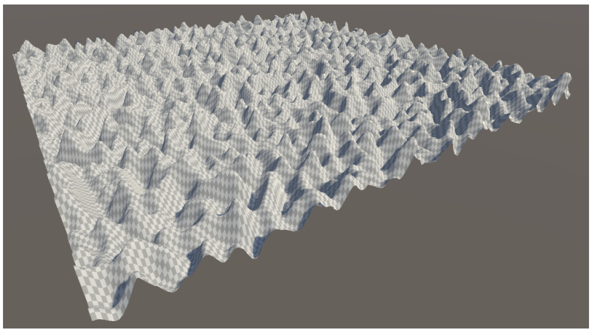


Figure 5: Generated terrain using *perlin noise*. The magnitude of the terrain is controlled by the duration and strength of the *mental command*.

3.3.3 Forrest Environment

Our third animation is a natural environment with rocks, trees, and grass that are waving in the wind, as illustrated in Figure 6. The user can transform this environment into a flower meadow with mushrooms, when triggering a *mental command*. As soon as the thought is relaxed, the flowers retreat into the ground.

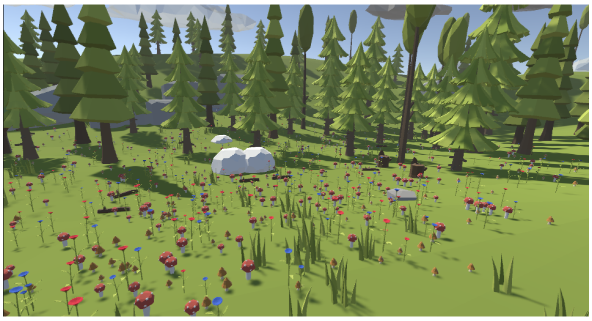


Figure 6: Forrest environment with mushrooms and flowers.

4 RESULTS

Our results using this headset showed several issues with the device. Originally, we planned to use the *relaxation value* to control animations, but due to the low sampling rate and inconsistencies within the metrics this plan was discarded. To determine whether the *performance metrics* were a reliable input for our CAVE application, we tested its accuracy by recording our mental state over a time span of 28 minutes. During the test, a relaxation video ran in the background and we closed our eyes to enter a state of relaxation. As shown in Figure 7, the relaxation value increases over time. However, we also found a correlation between the *stress* and *relaxation* value in our test (see Figure 8). This finding is unexpected, since relaxation and stress are emotions that usually do not correlate with each other.

This correlation and the low sampling rate of 0.1 Hz were the reason, why we utilized *mental commands* instead. The *mental commands* capability has a higher sampling rate, which allowed us to receive roughly one data sample per second. Using a single command, we were able to control the animations successfully. However, using more than one command severely affected the accuracy and made the use infeasible. Additionally, the result depended heavily on the contact quality of the headset. All electrodes must at least indicate some contact with the scalp to receive any *mental commands*. In our testing, we experienced that even subtle movements could disrupt the signal quality, which affected the experience in the CAVE significantly.

While training a command, the users were asked to hold on a thought for eight seconds. After each training, the users receive feedback on how representative the measurement was and can then decide, whether the training session should be erased or not. In the beginning, a profile was only trained with a single command (*push*) with a training time of 3-4 hours.

After we had sufficient control over one command, we tried to train a profile with higher command counts, with limited results. Even after extensive prior training the majority of *mental commands* was detected incorrectly. This is on par with other works, like Khan & Laique [13], who experienced similar results. In their work, a profile was trained on an *Emotiv Epoc* headset for 11 hours with 4 different *mental commands*, which resulted in an accuracy of only 25% [13]. This led us to only use one single command in our animations.

Users also experienced physical discomfort after already 15 minutes of use, because *Emotiv Insight's* sensors press tightly against the skull.

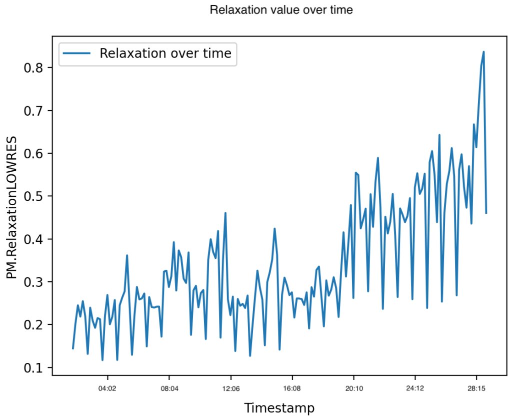


Figure 7: Relaxation value over a time span of 28 minutes.

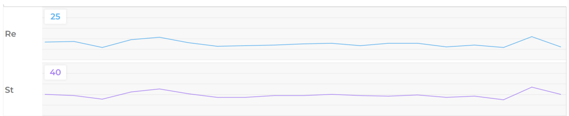


Figure 8: Line graph of the relaxation (Re, top) and stress (St, bottom) *performance metrics*. The figure indicates a correlation between the relaxation and stress value.

5 CONCLUSION AND DISCUSSION

In this preliminary study, we have investigated the use of the cost-efficient EEG device *Emotiv Insight* for VR control and tested the technical and practical feasibility of such a setup. We developed three different animations that react to the user's thoughts: A sphere, which can be accelerated, a procedurally generated terrain that forms valleys and hills and a forest environment, in which mushrooms and flowers are generated based on the user's brain activity.

Our experiments revealed a number of issues with the *Emotiv Insight* device. Severe reliability issues with the sensor data, the low sampling rate, as well as the physical discomfort of the device, led to the conclusion that *Emotiv Insight* is unsuitable as an input device for real-time applications. Our results further indicate a poor user experience, particularly due to reported headaches after only a brief period of wearing the BCI headset. Due to overwhelmingly negative feedback, a full evaluation of the setup involving this device was abandoned.

We also found that the initial training phase and the time-consuming task of adjusting the sensors make *Emotiv Insight* not suitable for applications in VR. Additionally, some important features of Emotiv's SDK, such as high-resolution *performance metrics* are restricted and require a costly monthly subscription.

We conclude that multiple improvements have to be made such that the *Emotiv Insight* becomes a viable alternative for controlling VR applications. Our suggestions include a more adjustable design, more sensors to improve accuracy, and overall a stronger focus on reliability. In the future, we will examine other BCI devices and conduct comparison experiments evaluating both usability and reliability. BCI devices are a promising technology, but need to overcome some shortcomings to be a practical input method for real-time interaction in VR environments.

REFERENCES

[1] Cipresso, P., Giglioli, I.A.C., Raya, M.A., Riva, G., 2018. The Past, Present, and Future of Virtual and Augmented Reality Research: A Network and Cluster Analysis of the Literature. Frontiers in Psychology 9.

[2] De Paolis, L., De Luca, V., 2020. The impact of the input interface in a virtual environment: the Vive controller and the Myo armband. Virtual Reality 24, 483–502.

[3] Emotiv, 2022. Cortex API Documentation. `https://emotiv.gitbook.io/cortex-api`. Accessed: 2022-05-19.

[4] Emotiv, 2024a. Emotiv Insight Product Page. `https://www.emotiv.com/insight//`. Accessed: 2022-05-19.

[5] Emotiv, 2024b. Emotiv Pro Licenses. `https://www.emotiv.com/products/emotivpro`. Accessed: 2024-05-03.

[6] Espiritu, N.M.D., Chen, S.A.C., Blasa, T.A.C., Munsayac, F.E.T., Arenos, R.P., Baldovino, R.G., Bugtai, N.T., Co, H.S., 2019. BCI-controlled Smart Wheelchair for Amyotrophic Lateral Sclerosis Patients, in: Proceedings of the International Conference on Robot Intelligence Technology and Applications (RiTA), pp. 258–263.

[7] Fayed, A.D., Rodriguez, J., Eisenschmidt, F., Li, Y., Yang, X., 2018. Virtual Reality Cognitive Training Program, in: Proceedings of the International Conference on Applied Cognitive Computing.

[8] Friedman, D., Leeb, R., Guger, C., Steed, A., Pfurtscheller, G., Slater, M., 2007. Navigating Virtual Reality by Thought: What Is It Like? Presence 16, 100–110.

[9] Galán, F., Nuttin, M., Lew, E., Ferrez, P., Vanacker, G., Philips, J., del R. Millán, J., 2008. A brain-actuated wheelchair: Asynchronous and non-invasive Brain-computer interfaces for continuous control of robots. Clinical Neurophysiology 119, 2159–2169.

[10] Guger, C., Holzner, C., Groenegress, C., Edlinger, G., Slater, M., 2009. Brain Computer Interface for Virtual Reality Control, in: Proceedings of the European Symposium on Artificial Neural Networks, pp. 443–448.

[11] Hammon, P.S., de Sa, V.R., 2007. Preprocessing and Meta-Classification for Brain-Computer Interfaces. IEEE Transactions on Biomedical Engineering 54, 518–525. doi:`10.1109/TBME.2006.888833`.

[12] Hu, M., Roberts, J., 2020. Built Environment Evaluation in Virtual Reality Environments-A Cognitive Neuroscience Approach. Urban Science 4. doi:`10.3390/urbansci4040048`.

[13] Khan, M.H., Laique, T., 2011. An Evaluation of Gaze and EEG-Based Control of a Mobile Robot. Master thesis. Blekinge Institute of Technology.

[14] Klem, G., Lüders, H., Jasper, H., Elger, C., 1999. The ten-twenty electrode system of the International Federation of Clinical Neurophysiology. Electroencephalography and clinical neurophysiology. Supplement 52, 3–6.

[15] Kübler, A., Kotchoubey, B., Kaiser, J., Wolpaw, J., Birbaumer, N., 2001. Brain-computer communication: Unlocking the Locked In. Psychological bulletin 127, 358–375.

[16] Lancelle, M., Settgast, V., Fellner, W.D., 2008. Definitely Affordable Virtual Environment, in: Proceedings of the Virtual Reality Conference, pp. 1–1. Video.

[17] Lécuyer, A., Lotte, F., Reilly, R.B., Leeb, R., Hirose, M., Slater, M., 2008. Brain-Computer Interfaces, Virtual Reality, and Videogames. Computer 41, 66–72. doi:`10.1109/MC.2008.410`.

[18] Leeb, R., Friedman, D., Müller-Putz, G., Scherer, R., Slater, M., Pfurtscheller, G., 2007a. Self-paced (asynchronous) BCI control of a wheelchair in virtual environments: A case study with a tetraplegic. Computational intelligence and neuroscience , 1–8.

[19] Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., Pfurtscheller, G., 2007b. Brain-Computer Communication: Motivation, Aim, and Impact of Exploring a Virtual Apartment. IEEE Transactions on Neural Systems and Rehabilitation Engineering 15, 473–482.

[20] Marin, I., Al-BattBootti, M.J.H., Goga, N., 2020. Drone Control based on Mental Commands and Facial Expressions, in: Proceedings of the International Conference on Electronics, Computers and Artificial Intelligence, pp. 1–4.

[21] Marshall, D., Coyle, D., Wilson, S., Callaghan, M., 2013. Games, Gameplay, and BCI: The State of the Art. IEEE Transactions on Computational Intelligence and AI in Games 5, 82–99.

[22] Maskeliunas, R., Damasevicius, R., Martisius, I., Vasiljevas, M., 2016. Consumer grade EEG devices: Are they usable for control tasks? PeerJ 4. doi:`10.7717/peerj.1746`.

[23] Millan, J.d.R., Rupp, R., Müller-Putz, G., Murray-Smith, R., Giugliemma, C., Tangermann, M., Vidaurre, C., Cincotti, F., Kübler, A., Leeb,

R., Neuper, C., Müller, K.R., Mattia, D., 2010. Combining Brain-Computer Interfaces and Assistive Technologies: State-of-the-Art and Challenges. Frontiers in neuroscience 4, paper 161. doi:`10.3389/fnins.2010.00161`.

[24] Perlin, K., 1985. An Image Synthesizer. SIGGRAPH Computer Graphics 19, 287–296. URL: `https://doi.org/10.1145/325165.325247`, doi:`10.1145/325165.325247`.

[25] Pfurtscheller, G., Neuper, C., Muller, G., Obermaier, B., Krausz, G., Schlögl, A., Scherer, R., Graimann, B., Keinrath, C., Skliris, D., Wortz, M., Supp, G., Schrank, C., 2003. Graz-BCI: state of the art and clinical applications. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11, 1–4. doi:`10.1109/TNSRE.2003.814454`.

[26] Prashant, P., Joshi, A., Gandhi, V., 2015. Brain computer interface: A review, in: Nirma University International Conference on Engineering (NUiCONE), pp. 1–6. doi:`10.1109/NUICONE.2015.7449615`.

[27] Ron-Angevin, R., Díaz-Estrella, A., 2009. Brain-computer interface: Changes in performance using virtual reality techniques. Neuroscience Letters 449, 123–127.

[28] Sabio, J., Williams, N.S., McArthur, G.M., Badcock, N.A., 2024. A scoping review on the use of consumer-grade eeg devices for research. PLOS ONE 19.

[29] Schomer, D.L., da Silva, F.H.L., 2017. Niedermeyer's Electroencephalography: Basic Principles, Clinical Applications, and Related Fields. Seventh ed., Oxford University Press.

[30] Settgast, V., Pirker, J., Lontschar, S., Maggale, S., Guetl, C., 2016. Evaluating Experiences in Different Virtual Reality Setups, in: Proceedings of the International Conference on Entertainment Computing, pp. 115–125.

[31] Teplan, M., 2002. Fundamental of EEG Measurement. IEEE Measurement Science Review 2, 1–11.

[32] Tsumoru, O., 1999. Computer input device for a physically disabled person using head movement. International Journal of Systems Science 30, 131–134.

[33] Zabcikova, Martina, 2019. Visual and auditory Stimuli Response, measured by Emotiv Insight headset. MATEC Web of Conferences 292, 01024. doi:`10.1051/matecconf/201929201024`.

A Comparative Study of Convex Combination and Inner Ordinate Methods For Scattered Data Interpolation Using Quartic Triangular Patch

Owen Tamin
Faculty of Computing and Informatics
Universiti Malaysia Sabah
Jalan UMS
Kota Kinabalu, 88400
Sabah, Malaysia
owentamin1996@gmail.com

Samsul Ariffin Abdul Karim
Data Technologies and Applications (DaTA) Research Lab
Faculty of Computing and Informatics
Jalan UMS
Kota Kinabalu, 88400
Sabah, Malaysia
samsulariffin.karim@ums.edu.my

Mohammad Khatim Hasan
Centre for Artificial Intelligence Technology,
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
Bangi, 43600 UKM
Bangi, Selangor, Malaysia
mkh@ukm.edu.my

ABSTRACT

In this study, we perform a comparative evaluation and assessment for the scattered data interpolation using a quartic polynomial triangular patch with ten control points on a triangular domain. The comparison is made using two different convex combinations and inner ordinates methods, i.e., cross derivative and cubic precision. Statistical Goodness-fit measurements used are maximum error, coefficient of determination (R^2), CPU time (in seconds), and contour plot. From the result, the cubic precision method with linear convex combination methods gave better results with smaller CPU times and higher R^2 value. All numerical and graphical results are presented using MATLAB programming.

Keywords

Quartic polynomial, convex combination, inner ordinates, scattered data, interpolation, triangular, patch

1 INTRODUCTION

Scattered data interpolation is a pivotal technique in modern computational science, offering a means to estimate values at arbitrary locations within a dataset characterized by irregularly distributed data points. Unlike structured datasets arranged on grids or meshes, scattered data often emerges from real-world observations or simulations, where data points are irregularly positioned in multidimensional space. Interpolating scattered data entails the construction of a continuous function that approximates the underlying behaviour of the dataset, enabling the inference of values at unobserved locations within the dataset's domain [Fran91].

There are two common categories for scattered data interpolation. The first type involves triangulating the data sets, while the second type, mesh-free interpolation, does not require the triangulation. Standard basis functions utilized in interpolation include Bézier, B-spline, and radial basis functions (RBFs). Previous studies [Cav19, Dell20, Dell18, Dell16, Ska23] have explored the application of the Shepard triangular method, a part of the meshless approach for surface reconstruction. However, these methods often demand significant computational time to generate interpolated surfaces. Alternatively, Bézier or spline triangular methods can construct piecewise smooth surfaces with desired degrees of smoothness (e.g., C^1 or C^2) while requiring less computational time, provided that certain continuity conditions are met at the adjacent boundary.

Goodman and Said [Good91] developed a C^1 triangular interpolant appropriate for interpolating scattered data. They achieved this by employing a convex combination scheme that combines three local schemes. Their work differs from that of Foley and Opitz [Foley92]. Furthermore, both works established a C^1 cubic triangular convex combination scheme. However, limited research has been conducted to compare the relative

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.

https://www.doi.org/10.24132/CSRN.3401.37

performance of these two methods. Foley and Opitz [Foley92] asserted that employing the cubic precision method resulted in enhanced visual smoothness of the surface for small scattered data sets, as well as a smaller root mean square (RMS) error compared to the method proposed by Goodman and Said [Good91]. Furthermore, two versions of convex combinations exist, differing only in the degree of the rational function. However, less research is conducted on determining which version is better when dealing with large scattered data sets.

Several studies have investigated the practical application of cubic Bézier curves in handling real-life data, primarily due to their simpler complexity and lower computational demands. For instance, Karim et al. [Karim18a] delved into spatial interpolation methods for rainfall data by employing cubic Bézier triangular patches to interpolate scattered data points. They also introduced a novel cubic Bézier-like triangular patches designed explicitly for interpolating scattered data [Karim20a]. Additionally, Karim and Saaban [Karim18b] utilized cubic Ball triangular patches to generate terrain data. However, the inherent simplicity of cubic Bézier patches is suitable only for achieving moderate smoothness in large datasets. A higher-degree polynomial is necessary to achieve a smoother surface. Quartic Bézier triangular patches have received less attention from researchers in the past due to their requirement of 15 control points, which need optimization techniques to ensure C^1 continuity is met on each adjacent triangles. This optimization process consumes more computational time, and the quartic scheme is global and not local [Piah06, Aziz05, Huss14a, Huss14b].

A recent study by Karim et al. [Karim20b] has successfully developed a C^1 scattered data interpolation scheme without requiring any optimization, building upon the extension work by Zhu and Han [Zhu13]. This newly proposed scheme is characterized as local, and the central processing unit (CPU) time needed for constructing the surface is notably faster compared to the quartic Bézier triangular patches in previous studies [Piah06, Aziz05, Huss14a, Huss14b]. Furthermore, the proposed scheme ensures positivity preservation in scattered data interpolation, leading to improved interpolated surfaces on the real-life data based on coronavirus disease 2019 (COVID-19) cases at Selangor State and Klang Valley in Malaysia. This advancement overcomes the limitations of previous works by [Ali20, Dra20, Karim20a], which did not apply positivity preservation in their interpolations. However, it is worth noting that this novel scheme has not yet been tested on other real-life datasets, such as geological data.

Motivated by these developments, this study aims to compare two version of convex combination and methods for forming local schemes based on the inner ordinates, focusing on the quartic Bézier surface interpolation scheme developed by Karim et al. [Karim20b]. The analysis will include error assessment using metrics such as R^2, maximum error, and CPU time. Additionally, graphical representations showcasing the surfaces and their corresponding contour plots will be presented. Finally, the proposed scheme will be applied to construct real-life geologic data, specifically Seamount data, to demonstrate its applicability in practical scenarios.

This paper is organized into five main sections. The first section introduces the research topic and outlines its objectives. The second section provides a review of related literature and methodologies from related studies. The third section elaborates on the methods utilized in the research. Subsequently, the fourth section presents the findings and engages in a comprehensive discussion. Finally, the fifth section concludes the paper and suggests areas for future work.

2 RELATED WORK

This section discusses the related work on the methods for forming local schemes for inner ordinates and convex combinations.

Goodman and Said's method

This method, also known as the cross derivative method, achieves a C^1 surface by prescribing first-order derivatives at the vertices. This method involves blending three cubic polynomials to create a rational function interpolant within each triangle of the domain. One of the significant advantages of this method is its locality; the surface at any point depends only on the data "close" to that point. This local nature makes the method efficient and well-suited for surface interpolation applications. The method begins by triangulating the domain using the data points as vertices. Within each triangle, a local interpolant is defined. To ensure a smooth surface, specific derivative values are typically employed on the boundaries of the triangles. The inner Bézier ordinates, denoted as b_{111}, are determined by the local scheme to satisfy the C^1 requirement along all edges ε_i, $i = 1,2,3$ of the triangular respectively as shown in Fig. 1.

Foley and Opitz's method

This method is also known as the cubic precision method. This method is best represented in Fig. 2 in which the two adjacent triangles with the vertex of V_i and $\tilde{V}_i$ where $i = 1,2,3$. The hybrid patch on the left triangle with boundary control point b_{ijk} is identical to the right triangle with boundary control point b_{ijk},

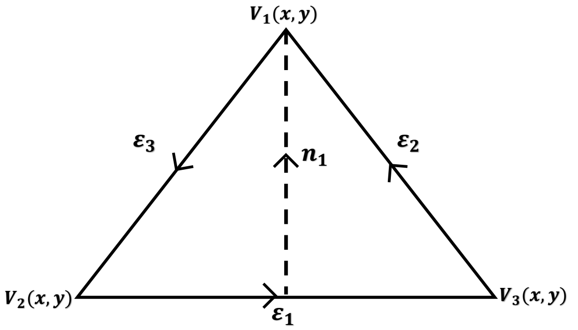


Figure 1: Triangle with side-vertex blending

where $i+j+k=3$. The C^1 continuity is achieved by making sure the sets of four control points along the edge from both sides of the triangle are co-planar. Complete derivations can be found in Karim et al. [Karim20b].

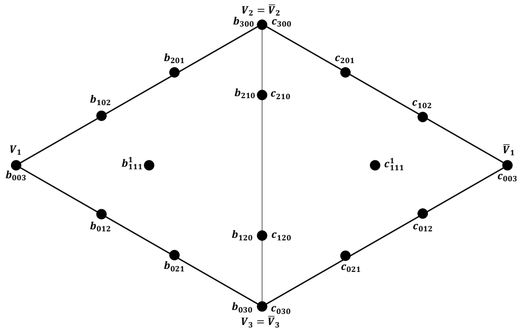


Figure 2: Two adjacent hybrid cubic patches

Convex Combination

In the literature, there are two common convex combination schemes that can be used in scattered data interpolation for calculating the inner ordinates $b^i_{111}, i=1,2,3$ where the barycentric coordinate (u,v,w) on the triangle is defined by $u+v+w=1$, where $u,v,w\geq 0$. Eqn. (1) represents a linear convex combination, whereas Eqn. (2) represents a square linear convex combination which can be shown as below:

$$b_{111}=\rho_1 b^1_{111}+\rho_2 b^2_{111}+\rho_3 b^3_{111}$$

where

$$\begin{aligned}\rho_1&=\frac{vw}{uv+vw+wu},\quad \rho_2=\frac{wu}{uv+vw+wu},\\ \rho_3&=\frac{uv}{uv+vw+wu}\end{aligned}\tag{1}$$

or

$$\begin{aligned}\rho_1&=\frac{v^2w^2}{u^2v^2+v^2w^2+w^2u^2},\quad \rho_2=\frac{w^2u^2}{u^2v^2+v^2w^2+w^2u^2},\\ \rho_3&=\frac{u^2v^2}{u^2v^2+v^2w^2+w^2u^2}\end{aligned}\tag{2}$$

3 METHOD

This section describes the construction of quartic triangular patches on a triangular domain.

3.1 Quartic Triangular Patches

Since we are dealing with triangulation, the barycentric coordinates (u,v,w) on the triangle T with vertices V_1, V_2 and V_3 is defined by $u+v+w=1$, where $u,v,w\geq 0$. Set the point inside the triangle as $V(u,v,w)\in R^2$ (as shown in Fig. 3), which can be expressed as:

$$V=uV_1+uV_2+uV_3$$

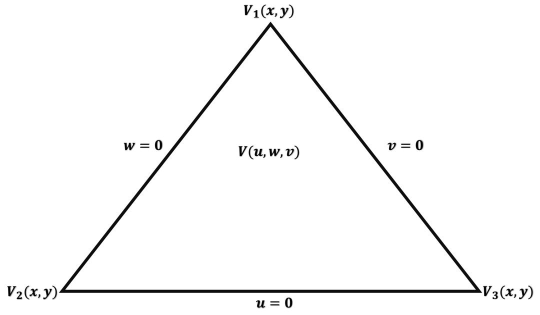


Figure 3: Triangle T

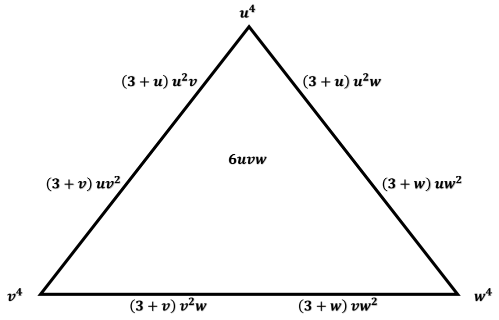


Figure 4: Basis functions for quartic triangular patch

Fig. 4 shows the basis functions for quartic triangular patch. The basis functions satisfy the properties of non-negativity, partition of unity and symmetry (full details of the properties can be referred to Zhu and Han [Zhu13]). Thus, the quartic triangular patch from Zhu and Han is further defined by

$$\begin{aligned}R(u,v,w)=&u^4b_{300}+v^4b_{030}+w^4b_{003}+u^2v(3+u)b_{210}+\\ &(3+u)u^2wb_{201}+(3+v)v^2ub_{120}+\\ &(3+v)v^2wb_{021}+(3+w)w^2ub_{102}+\\ &(3+w)w^2vb_{012}+6uvwb_{111}\end{aligned}\tag{3}$$

3.2 Scattered Data Interpolation Using Quartic Triangular Patches

In this section, the following scattered data interpolation scheme is constructed based on Karim et al. [Karim20b].

3.2.1 Boundary Ordinates

The boundary ordinates are calculated based on Goodman and Said [Good91] method for each triangle, as represented in Fig. 5.

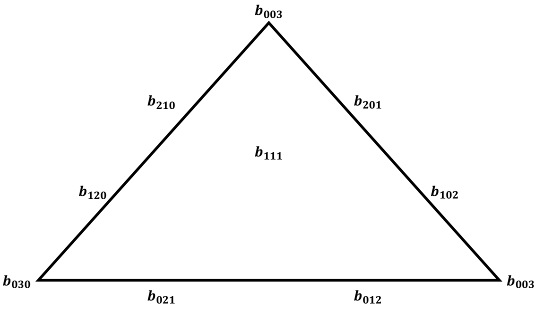


Figure 5: Control points for quartic triangular patch

Based on the literature review from Sec. 2, the vertex is given as $F(V_1) = b_{300}$, $F(V_2) = b_{030}$, and $F(V_3) = b_{003}$. The other six boundary ordinates are expressed as:

$$\begin{aligned}
b_{210} &= b_{300} + \frac{1}{4}\left((x_2 - x_1)F_x(V_1) + (y_2 - y_1)F_y(V_1)\right)\\
b_{201} &= b_{300} - \frac{1}{4}\left((x_1 - x_3)F_x(V_1) + (y_1 - y_3)F_y(V_1)\right)\\
b_{021} &= b_{030} + \frac{1}{4}\left((x_3 - x_2)F_x(V_2) + (y_3 - y_2)F_y(V_2)\right)\\
b_{120} &= b_{030} - \frac{1}{4}\left((x_2 - x_1)F_x(V_2) + (y_2 - y_1)F_y(V_2)\right)\\
b_{102} &= b_{003} + \frac{1}{4}\left((x_1 - x_3)F_x(V_3) + (y_1 - y_3)F_y(V_3)\right)\\
b_{012} &= b_{003} - \frac{1}{4}\left((x_3 - x_2)F_x(V_3) + (y_3 - y_2)F_y(V_3)\right)
\end{aligned}$$

3.2.2 Inner Ordinates

The remaining inner ordinates, $b^i_{111}, i = 1,2,3,$ are obtained by first using the cross derivative method by Goodman and Said [Good91]. Then, after the inner ordinates are calculated, the second inner ordinates are then calculated using cubic precision by Foley and Opitz [Foley92].

3.2.3 Final Scheme

The final interpolating scheme can be written as

$$\begin{aligned}
R(u,v,w) = &\sum_{\substack{i+j+k=3\\ i.j.k\neq 1}} b_{ijk}B^3_{i,j,k}(u,v,w)\\
&+ 6uvw\left(\rho_1 b^1_{111} + \rho_2 b^2_{111}\right.\\
&\left.+\rho_3 b^3_{111}\right)
\end{aligned} \tag{4}$$

where $B^3_{i,j,k}(u,v,w)$ represents the Bernstein polynomial of degree 3, and ρ_1, ρ_2, and ρ_3 are determined by Eqn. (1) or (2).

3.3 Algorithm

This section shows the overall algorithm used in this study, as represented in Algorithm 1.

Algorithm 1 Quartic Triangular Patches for Scattered Data Interpolation

```
Input: scattered data points
Estimate the partial derivative at the data points by using [Karim20a];
Triangulate the data points by using Delaunay triangulation;
Calculate the boundary control points;
Calculate inner control points for the local scheme, $b^i_{111}, i = 1,2,3$ by using the cubic precision method as in Foley and Opitz [Foley92];
Construct the interpolated surface using the convex combination method of three local schemes defined by Eqn. (4)
```

---

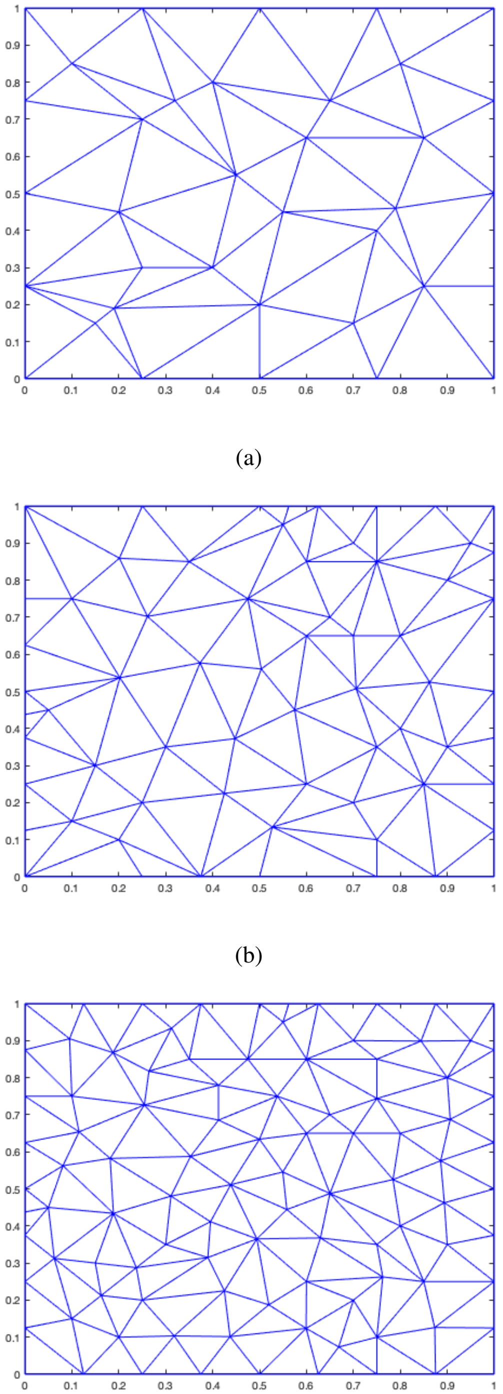
4 RESULTS AND DISCUSSION

In this section, we will compare the performance between the different convex combination and methods for forming the local scheme. The proposed scheme is tested with one well-known function, Franke's exponential function:

$$\begin{aligned}
F_1(x,y) =& 0.75e^{-\frac{(9x-2)^2+(9y-2)^2}{4}} + 0.75e^{-\frac{(9x+1)^2}{49} - \frac{(9y+1)^2}{10}}\\
&+ 0.5e^{-\frac{(9x-7)^2+(9y-3)^2}{4}} - 0.2e^{-(9x-4)^2-(9y-7)^2}
\end{aligned}$$

The error norms are computed using a 33×33 uniform rectangular grid of evaluation points in the unit square for 36, 65 and 100 data points. Fig. 6 shows the Delaunay triangulation for sample of 36, 65 and 100 data points. The error measurements used are (a) Coefficient of determination, R^2, (b) maximum errors and (c) Central Processing Unit (CPU) time in seconds. The simulation is conducted using MATLAB 2023a version.

Fig. 7 shows an example of scattered data interpolation using quartic triangular patches based on $F_1(x,y)$ function based on 65 data points. The scattered data interpolation is performed within each mesh triangle, which is then evaluated at points on a 33×33 uniform rectangular grid within the unit square domain. This evaluation involves applying the interpolant function to each grid point to estimate the function value based on the test function F_1. The contour plot of the respective surface interpolation is shown in Fig. 8.



(a)

(b)

(c)

Figure 6: Delaunay triangulation for (a) 36 data points; (b) 65 data points and (c) 100 data points

Based on Fig. 7, there are no significant differences between the surface of the F_1 function and the four interpolated surfaces. However, when it comes to the contour plot, which can be referred to in Fig. 8, there are no exact similarities between the four contour plots for the interpolated surface. These differences indicate that different methods with different convex combinations give different appearances, and it's worth further investigating which one of them is actually the best. A better comparison between the methods is represented with error analysis, which is shown in Table 1.

Based on Table 1, the cubic precision method with linear convex combination shows the best performance in terms of higher R^2 value and smaller CPU time (in seconds) as compared to the three methods. Interestingly, the linear convex combination also favors the cross derivative method, where its performance is better than the one with the square convex combination. This observation tells us that selecting the convex combination depends on which method is acquired for scattered data interpolation. This observation will help researchers to be able to select the best choice for selecting which convex combination to choose when it comes to different methods for calculating the inner ordinates.

Another interesting observation is that the performance of both proposed methods significantly improved as the number of data points increased from 33 to 100, as evidenced by higher R^2 values, reduced maximum errors, and more efficient CPU utilization. This improvement can be attributed to the increase in triangulation density from coarse to fine, enabling the capture of subtle variations and features of the underlying surface of the test function F_1. This pattern is also reflected in the contour plot shown in Fig. 8, where visual smoothness is observed to increase with the higher density of data points. Finer triangulation are adept at accurately representing intricate details of the surface but may lead to increased computational complexity and runtime. On the other hand, coarser triangulation simplify the surface representation but can potentially result in a loss of detail and accuracy.

Nevertheless, the cubic precision method with linear convex combination still performs the best based on the four-method comparison. By having this result, the best performing method will be used to reconstruct real-life geologic data, which is the Seamount data. Seamount data sets obtained from MATLAB represent the surface of the Mount under the sea located on Louisville Ridge in the South Pacific in 1984. Fig. 9 shows the Delaunay triangulation of Seamount with 294 data points as well as the 3D interpolation of those data points. Fig. 10 shows the surface of the Seamount data points as well as the interpolated surface using cross derivative and cubic precision with linear combination. The interpolated surface consists of 566 triangles and is evaluated by subdivision of the triangles. Table 2 shows the CPU time comparison between the two methods. The CPU time is taken by averaging three runs of the experiment to prevent bias. Based on the table, the cubic precision with linear convex combination takes a lower time to reconstruct the Seamount data points at 3.3901 seconds faster than cross derivative with the linear combination. This observation shows that the cubic precision method developed by Foley and Opitz [Foley92] is more accurate and faster than the cross derivative by Goodman and Said [Good91].

A claim by Foley and Opitz [Foley92] mentioning that their method works better than cross derivative is valid in this case even though the differences are not significant. These small differences can lead to big changes in performance when it comes to interpolating more big data points. Furthermore, the CPU time required to reconstruct seamount data by using a cubic Timmer triangular patch developed by Ali et al. [Ali20] scheme is 123.9761 seconds. Meanwhile, the proposed quartic triangular patch has achieved an average CPU time of 83.4646 (cross derivative) and 80.0745 (cubic precision) for linear convex combination. Based on this result, the proposed scattered data interpolation scheme by using quartic polynomial is at least 1.5 time faster than scattered data interpolation scheme based on cubic triangular Timmer developed by Ali et al. [Ali20]. This result is significant because the proposed scheme has quartic degree, meanwhile the existing schemes are cubic degree.

5 CONCLUSION

In conclusion, this study conducted a comparative evaluation of scattered data interpolation using quartic polynomial triangular patches with ten control points on a triangular domain. The comparison focused on two different methods: convex combination with inner ordinates using cross derivatives and cubic precision. The results of our evaluation indicate that the cubic precision method, coupled with linear convex combination schemes, outperformed the cross derivative method. Specifically, the cubic precision method exhibited shorter CPU times and higher R^2 values, demonstrating its effectiveness in interpolating scattered data using quartic polynomial triangular patches. Furthermore, our study highlights the efficacy of the linear convex combination scheme in conjunction with the cubic precision method for scattered data interpolation. This combination yielded superior results, particularly in reconstructing geologic real-life data, such as Seamount, and outperformed the linear convex combination scheme with the cross derivative method in terms of computational efficiency. Future studies will focus on implementing these successful schemes on GPU platforms using machine learning techniques. This strategic approach aims to enhance computation time for surface reconstruction while simultaneously improving accuracy. Overcoming challenges in GPU implementation and leveraging machine learning algorithms will contribute significantly to advancing the field of scattered data interpolation.

6 ACKNOWLEDGMENTS

This research was fully supported by Ministry of Higher Education (MOHE) of Malaysia through Fundamental Research Grant Scheme **[FRGS/1/2023/ICT06/UMS/02/1] (New Scattered Data Interpolation Scheme Using Quasi Cubic Triangular Patches for RGB Image Interpolation)** and Universiti Malaysia Sabah. Special thanks to the Faculty of Computing and Informatics, Universiti Malaysia Sabah for the tremendous computing facilities support.

7 REFERENCES

[Fran91] Franke, R., & Nielson, G. M. (1991). Scattered data interpolation and applications: A tutorial and survey. Geometric Modeling: Methods and Applications, 131-160.

[Cav19] Cavoretto, R., De Rossi, A., DellAccio, F., Di Tommaso, F. (2019). Fast computation of triangular Shepard interpolants. Journal of Computational and Applied Mathematics, 354, 457-470.

[Dell20] Dell'Accio, F., Di Tommaso, F. (2020). On the hexagonal Shepard method. Applied Numerical Mathematics, 150, 51-64.

[Dell18] Dell'Accio, F., Di Tommaso, F., Nouisser, O., Zerroudi, B. (2018). Increasing the approximation order of the triangular Shepard method. Applied Numerical Mathematics, 126, 78-91.

[Dell16] DellâAccio, F., Di Tommaso, F., Hormann, K. (2016). On the approximation order of triangular Shepard interpolation. IMA Journal of Numerical Analysis, 36(1), 359-379.

[Good91] Goodman, T. N. T., Said, H. B. (1991). A C^1 triangular interpolant suitable for scattered data interpolation. Communications in Applied Numerical Methods, 7(6), 479-485.

[Foley92] Foley, T. A., Opitz, K. (1992). Hybrid cubic Bézier triangle patches. In Mathematical methods in computer aided geometric design II (pp. 275-286). Academic Press.

[Karim18a] Karim, S. A., Saaban, A., Hasan, M. K., Sulaiman, J., Hashim, I. (2018). Interpolation using cubic Bézier triangular patches. Int. J. Adv. Sci. Eng. Inf. Technol, 8(4-2), 1746-1752.

[Karim20a] Karim, S. A. A., Saaban, A., Skala, V., Ghaffar, A., Nisar, K. S., Baleanu, D. (2020).

Convex combination	Data Points (F_1)	R^2		Max Error		CPU Time (s)	
		CD[1]	CP[2]	CD[1]	CP[2]	CD[1]	CP[2]
Linear	36	0.9917	0.9918	0.1146	0.1110	0.2948	0.2910
	65	0.9960	0.9962	0.0789	0.0786	0.3238	0.3382
	100	0.9990	0.9991	0.0632	0.0643	0.4246	0.3752
Square	36	0.9916	0.9916	0.1151	0.1111	0.4644	0.3492
	65	0.9960	0.9961	0.0785	0.0787	0.4828	0.3331
	100	0.9989	0.9990	0.0629	0.0655	0.5160	0.4962

[1] Cross Derivative
[2] Cubic Precision

Table 1: Overall error measurements for 36, 65 and 100 data points for Quartic Bézier Triangular Surface with different method and convex combination

Method	Convex combination	CPU Time (s)			
		First run	Second run	Third run	Average
Cross Derivative	Linear	81.1796	89.1468	80.0674	83.4646
Cubic Precision	Linear	79.9700	80.2119	80.0416	80.0745

Table 2: CPU time comparison between cross derivative and cubic precision method using linear convex combination

Construction of new cubic Bézier-like triangular patches with application in scattered data interpolation. Advances in Difference Equations, 2020(1), 151.

[Karim18b] Karim, S. A. B. A., Saaban, A. (2018). Visualization terrain data using cubic Ball triangular patches. In MATEC Web of Conferences (Vol. 225, p. 06023). EDP Sciences.

[Piah06] Mt Piah, A. R., Saaban, A., Abd Majid, A. (2006). Range restricted positivity-preserving scattered data interpolation. Journal of Fundamental Sciences, 2(1-2), 63-75.

[Aziz05] Azizan, S., Piah, A. R. M., Ahmad, A. M. (2005). G^1 scattered data interpolation with minimized sum of squares of principal curvatures.

[Huss14a] Hussain, M., Abd Majid, A., Hussain, M. Z. (2014). Convexity-preserving Bernstein-Bézier quartic scheme. Egyptian Informatics Journal, 15(2), 89-95.

[Huss14b] Hussain, M., Hussain, M. Z., Buttar, M. (2014). C^1 positive Bernstein-Bézier rational quartic interpolation triangle, 1(2), 3.

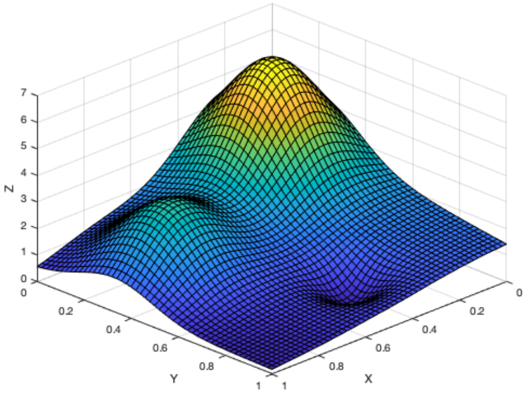
[Karim20b] Karim, S. A. A., Saaban, A., Nguyen, V. T. (2020). Scattered data interpolation using quartic triangular patch for shape-preserving interpolation and comparison with mesh-free methods. Symmetry, 12(7), 1071.

[Zhu13] Zhu, Y., Han, X. (2013). A class of $\alpha\beta\gamma$-BernsteinâBézier basis functions over triangular domain. Applied Mathematics and Computation, 220, 446-454.

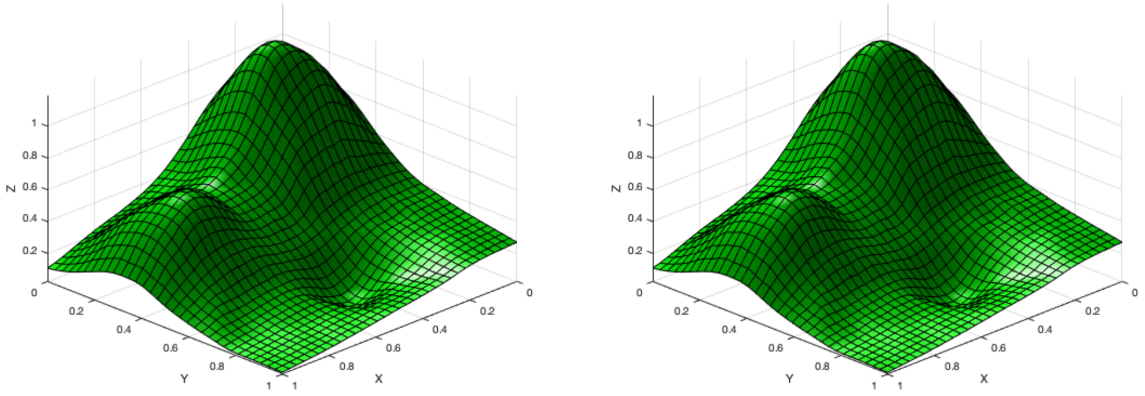
[Ali20] Ali, F. A. M., Abdul Karim, S. A., Saaban, A., Hasan, M. K., Ghaffar, A., Nisar, K. S., Baleanu, D. (2020). Construction of cubic Timmer triangular patches and its application in scattered data interpolation. Mathematics, 8(2), 159.

[Dra20] Draman, N. N. C., Karim, S. A. A., Hashim, I. (2020). Scattered data interpolation using rational quartic triangular patches with three parameters. IEEE Access, 8, 44239-44262.

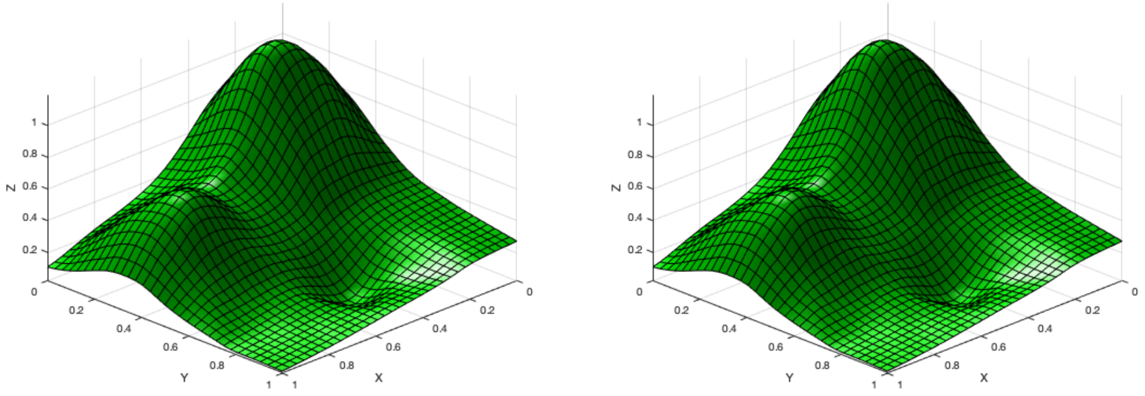
[Ska23] Skala, V.; Mourycova, E. Meshfree Interpolation of Multidimensional Time-Varying Scattered Data. Computers 2023, 12, 243. https://doi.org/10.3390/computers12120243



(a) F_1 test function

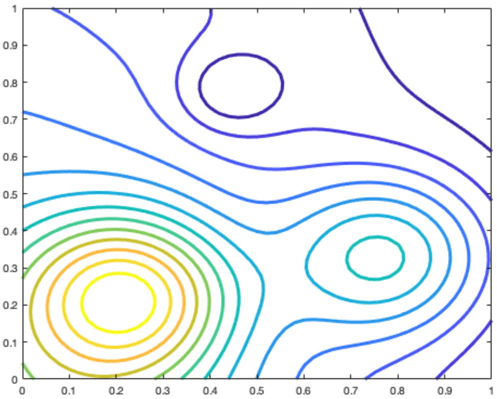


(b) Cross derivative method with linear convex combination (c) Cross derivative method with square convex combination

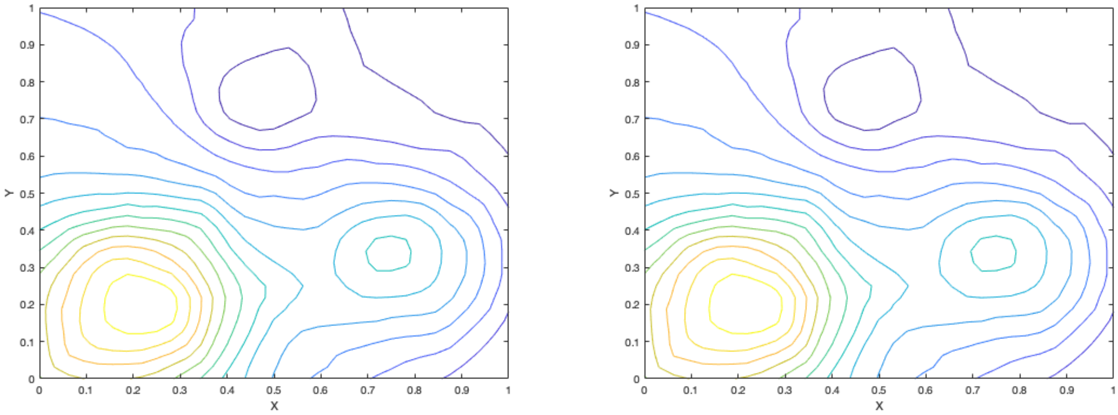


(d) Cubic precision method with linear convex combination (e) Cubic precision method with square convex combination

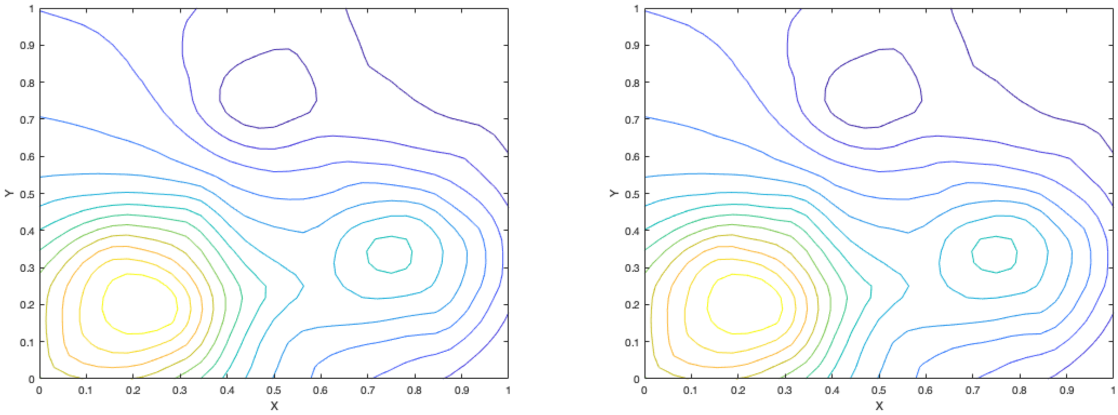
Figure 7: Example of surface interpolation based on F_1 test function



(a) F_1 test function

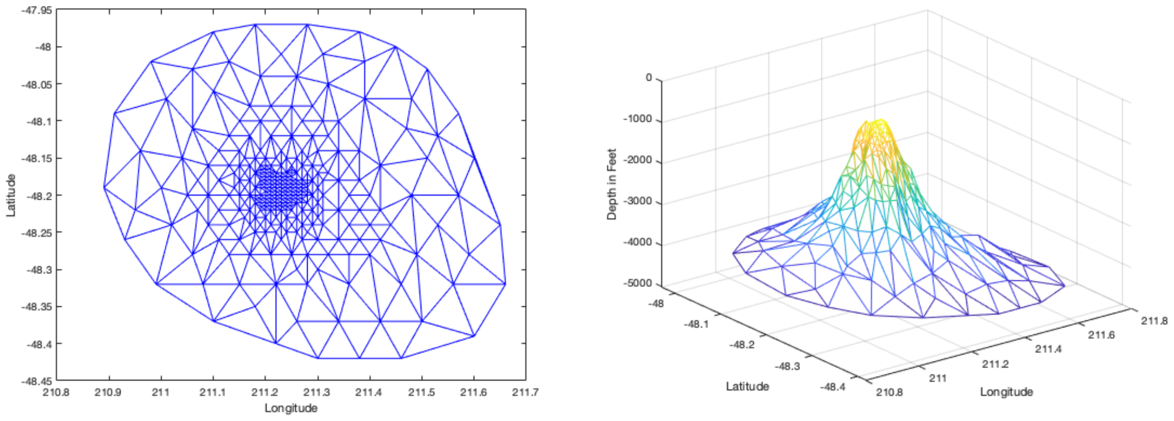


(b) Cross derivative method with linear convex combination (c) Cross derivative method with square convex combination



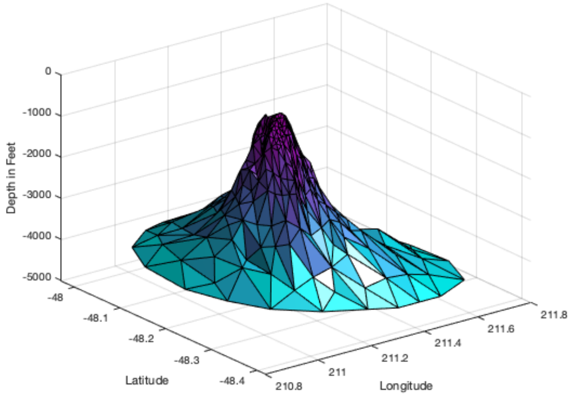
(d) Cubic precision method with linear convex combination (e) Cubic precision method with square convex combination

Figure 8: Contour plots of surfaces presented in Fig. 5

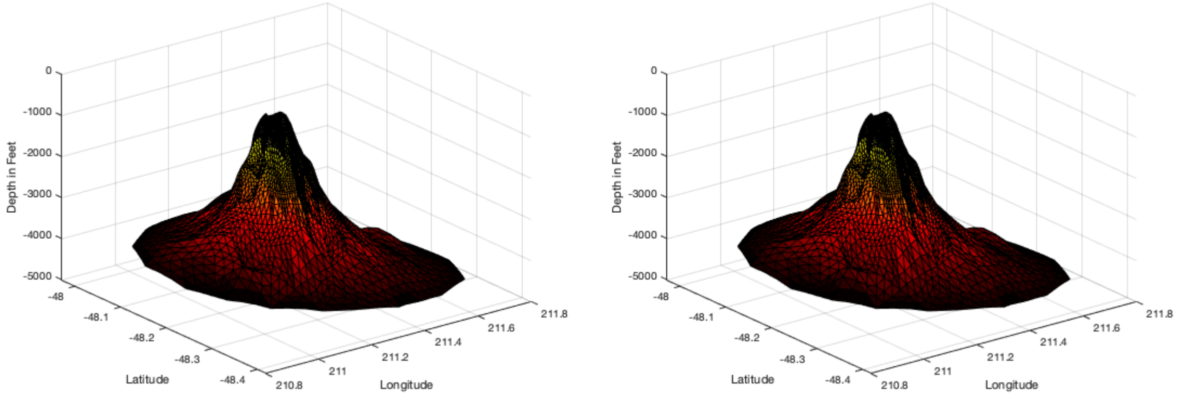


(a) (b)

Figure 9: Example of (a) Delaunay Triangulation of Seamount with 294 data points and (b) 3D interpolation of Seamount data points



(a) Seamount surface generated in Matlab



(b) Cross derivative method with linear convex combination (c) Cubic precision method with linear convex combination

Figure 10: Example of surface generated based on Seamount real data points

Lightweight single image dehazing utilizing relative depth information

Panagiotis Frasiolas
CERTH
6th km
Charilaou-Thermi
road
Greece 57001,
Thessaloniki
frasiolas@iti.gr

Asterios Reppas
CERTH
6th km
Charilaou-Thermi
road
Greece 57001,
Thessaloniki
asterisreppas@iti.gr

Konstantinos
Konstantoudakis
CERTH
6th km
Charilaou-Thermi
road
Greece 57001,
Thessaloniki
k.konstantoudakis@iti.gr

Dimitrios Zarpalas
CERTH
6th km
Charilaou-Thermi
road
Greece 57001,
Thessaloniki
zarpalas@iti.gr

ABSTRACT

Considering the need for lightweight and fast implementations, this paper presents an architecture based on a MobileVit encoder for efficiency and speed, introducing a fully convolutional lightweight decoder with skip connections for feature extraction. The main purpose of this network is to address the problem of single image dehazing. Recognizing the critical role of depth information in assisting the above task, the merging of these two tasks into a single network was performed in a supervised manner. Taking into account that there is a shortage of datasets that provide both dehazing and relative depth estimation ground truths, Depth Anything was utilized to extract the relative depth values of the images, which is the SOTA network in this task.

Keywords

Lightweight, Vision Transformers, relative depth, dehazing

1 INTRODUCTION

Despite the recent advancements in computer vision research, scene understanding remains a fundamental problem. Monocular depth estimation provides a deeper insight to the scene, capturing depth information and transforming perception from a two-dimensional representation to a richer three-dimensional understanding. It has a potential to revolutionize applications such as autonomous navigation [4], augmented reality, and scene understanding.

In the presence of haze, because it has a strong effect on visual clarity and detail, comprehending a scene becomes really challenging. Single image dehazing aims to mitigate the adverse effects of atmospheric scattering, enhancing the visibility and fidelity of images captured in hazy or foggy conditions and aims to restore the true radiance of objects obscured by haze or fog. In essence, both depth estimation and image dehazing share a common goal: the recovery of a more faithful representation of the scene.

Recent studies of both monocular depth estimation and single image dehazing methods, have introduced Vision Transformers [25, 1, 8] as a fundamental component for a global understanding of the scene, unlike traditional methods which rely on convolutional neural networks [17], [7], [5] with limited receptive field.

Tasks like self-driving cars require real-time processing, because it directly impacts user experience and safety. Single image dehazing is essential in scenarios where visibility is compromised due to adverse weather conditions. Real-time dehazing can enhance image clarity and enable immediate responses. Real-time processing ensures that the information provided is current, allowing systems to react swiftly and effectively to changing environments and unforeseen obstacles. That is the reason why a lightweight model has been developed in this paper.

The proposed model is based on an encoder-decoder architecture. MobileVit [22], a Vision Transformer with a low complexity, is employed in the encoder, aiming to minimize the model's parameter count. The decoder is a fully-convolutional neural network. In the earlier stages of the decoder both dehazing and depth estimation are learned simultaneously and in later stages these tasks are separated into 2 branches. Skip-connections from the encoder to the decoder result in an efficient and

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.

effective feature processing pipeline and enable the network to leverage both fine and coarse-grained details. The combination of these components allows the decoder to generate accurate pixel-wise predictions. This approach contributes to the ability of the network to efficiently and accurately process visual data.

The main contributions in this paper are summarized below:

- Proposition of a lightweight network for single image dehazing utilizing MobileVit [22] in the encoder
- Utilization of the relative depth values in order to help the image dehazing tasks and also extract a relative depth image from the input hazy image.
- Utilization of a fully-convolutional, fast and accurate decoder with skip connections that requires few parameters.

2 RELATED WORK

2.1 Monocular depth estimation

An early work on monocular depth estimation is that of Eigen et al. [7], which uses two CNNs. The first is used to predict a coarse global depth, and the second to refine the prediction locally. Jung et al. [13] proposed a solution for poor boundary localization and spurious regions by using a two-stage convolutional network as a generator. Their approach employs a deep adversarial learning framework, with an adversarial discriminator training criterion aiming to effectively tell real and synthetically generated depth images apart. A lightweight model was proposed by Wofk et al. [31], utilizing MobileNet [12] as the decoder in order to have a low count of parameters. They also incorporated skip-connections between the encoder and the decoder. Rudolph et al. [28] used a Guided Upsampling Block (GUB) for building the decoder. GUB relies on the image to guide the decoder in upsampling the feature representation and the depth map reconstruction, achieving high resolution results with fine-grained details. Lee et al. [16] proposed a token sharing transformer that utilizes global token sharing, which enables the model to obtain an accurate depth prediction with high throughput in embedded devices. The model used for the ground truth relative depth images is Depth Anything [33], a practical solution for robust monocular depth estimation, focusing on simplicity and effectiveness. By scaling up the dataset to approximately 62 million unlabeled images and employing data augmentation techniques and auxiliary supervision, the method achieves impressive generalization across various datasets.

2.2 Single image dehazing

To describe the formation of a hazy image, the atmospheric scattering model was first proposed by McCartney [21]. The equation of this model can be written as

$$I(x) = J(x) \cdot t(x) + a \cdot (1 - t(x)) \qquad (1)$$

where I(x) is a hazy image, J(x) is the real scene to be recovered, t(x) is the medium transmission, a is the global atmospheric light.

DCP Net [9] is a simple but effective image prior - dark channel prior to remove haze from a single input image. DehazeNet [2], which is one of the earliest deep learning works, uses a CNN with specialized Maxout layers for haze-related feature extraction and introduces the Bilateral Rectified Linear Unit (BReLU) activation function to enhance haze-free image quality. Ren et al. [27] used an encoder-decoder architecture and adopted a novel fusion-based strategy which derives three inputs from an original hazy image by applying white balance, contrast enhancement, and gamma correction. Dong et al. [5] presented a Multi-Scale Boosted Dehazing Network using the U-Net framework, which is designed based on two principles: boosting and error feedback. The model incorporates the Strengthen-Operate-Subtract boosting strategy in the decoder, gradually enhancing the haze-free image. They introduced a dense feature fusion module with back-projection feedback in the U-Net architecture to maintain spatial information. Hong et al. [11] introduced a knowledge distillation-based dehazing network that employs process-oriented learning with the student network mimicking image reconstruction. Wu et al. [32] introduced a novel regularization technique that utilizes contrastive learning. CR leverages hazy images as negatives and clear images as positives, guiding the restored image closer to clear images and away from hazy ones in the representation space. Cui et al. [3] was inspired by the consistent degradation of various regions in corrupted images, and suggested a shift towards prioritizing essential areas for reconstruction. In the latter approach, they introduced a dual-domain selection mechanism to accentuate critical information for restoration, including elements like edge signals and challenging regions. FFA-NET [24] is an end-to-end feature fusion attention network, consisting of three key components.1) Channel Attention with Pixel Attention mechanism, 2) Local Residual Learning, and 3) An Attention-based different levels Feature Fusion (FFA) structure, that performs especially outstanding in regions with thick haze and rich texture details. AOD-Net [18] is designed based on a re-formulated atmospheric scattering model and directly generates clean images through a lightweight CNN, making it easily embeddable into other deep models. MSCNN [26] is a multi-scale CNN consisting of a coarse-scale net that predicts a holistic transmission map based on the entire image, and a fine-scale net that refines results locally. LightDehazeNet [29] jointly estimates both the transmission map and atmospheric light using a transformed atmospheric scattering model. There are not many lightweight models to perform image dehazing delivering good results. The proposed method targets these two goals concurrently.

2.3 Vision Transformers

Vision Transformer (ViT) [6] adapts the transformer architecture used in natural language processing to extract multiscale information from images by breaking them down into smaller patches. The most important part is the self-attention mechanism which helps to encode relationships between the patches. ViT-based models have achieved remarkable results in tasks like image classification and segmentation, depth estimation, and single-image dehazing. Depth estimation works like Ada-Bins [1] proposed a transformer-based architecture block that divides the depth range into bins whose center value is estimated adaptively per image. The final depth values are estimated as linear combinations of the bin centers. Ranft et al. [25] gathered tokens from different stages in the vision transformer to create representations that resemble images at various resolutions. These representations are gradually fused to produce full-resolution predictions using a convolutional decoder. In another encoder-decoder architecture, Kim et al. [14] deployed a hierarchical transformer-based encoder to capture the global information in an image, and a lightweight decoder to generate an estimated depth-map, while also considering local connectivity. Vision transformers are also used in single-image dehazing. Guo et al. [8] proposed a novel transmission-aware 3D position embedding to involve haze density-related prior information into the vision transformer. Lu et al. [20] created two modules, one for handling both fine textures and large hazy areas, and another for addressing uneven haze distribution in image dehazing. The first module uses parallel dilated convolutions with large receptive fields, while the second efficiently extracts global and local information in parallel to improve dehazing results. Zhao et al. [34] combined intrinsic image decomposition and image dehazing, enhancing the generation of high-quality haze-free images. The Complementary Feature Selection Module (CFSM) was used to effectively fuse complementary features, thereby boosting feature aggregation. In the scope of this research paper, vision transformers are adopted as the encoder of the network. The reason for this selection relies on the proven capacity of vision transformers to produce rapid and precise results. Vision transformers are favored for their adeptness in managing visual data, detecting patterns, and comprehending the content of images.

3 METHOD

An image dehazing network is trained Fig.1 which aims to predict the dehazed image $Y \in \Re^{H\times W\times 3}$ and the relative depth map $D \in \Re^{H\times W\times 1}$ from an RGB hazy image $X \in \Re^{H\times W\times 3}$. The primary focus lies on the dehazing output, with the relative depth output serving a supplementary role. The relative depth information primarily aids the image dehazing task, given its inherent inclusion within the haze-scattering model. Eq.1.

To accomplish that, a model based on an encoder-decoder architecture was implemented. In the encoder, a pre-trained MobileVit [22] was used, and in the decoder a fully convolutional network. MobileViT is designed to bring together the strengths of CNNs and vision transformers to create a lightweight and fast-to-evaluate network for mobile vision tasks. It offers a new perspective on how to process visual information efficiently using transformer-based approaches in the context of mobile devices. Most of the standard encoders like ResNet [10] are fully convolutional and do not utilize the benefits of Vision Transformers. The resulting feature map is further upscaled and integrated with the MobileVit layer outputs. To the best of our knowledge, this is the first paper that uses MobileVit as the encoder for the single image dehazing problem. The model has a total of 2.29 million parameters, positioning it as a lightweight solution without compromising its performance.

3.1 Encoder

The encoder extracts the feature map from the input image. For this extraction to be possible the classification layer of MobileVit was deleted. MobileViT block combines CNN (local information) and transformers (global information). It uses convolutions for local details, then transforms patches to capture relationships between image parts. Four intermediate blocks and the output of the Encoder are used for feature extraction, each capturing different aspects of the input image. As shown in Fig.1 each of these blocks (light blue color), has a different width, height and channel values.

- Block 1: $\frac{1}{2} \times \frac{1}{2} \times C1$
- Block 2: $\frac{1}{4} \times \frac{1}{4} \times C2$
- Block 3: $\frac{1}{8} \times \frac{1}{8} \times C3$
- Block 4: $\frac{1}{16} \times \frac{1}{16} \times C4$
- Out: $\frac{1}{32} \times \frac{1}{32} \times C5$

As these stages progress, the feature maps become smaller but contain richer information. This helps in the understanding of the relationships between different parts of the image.

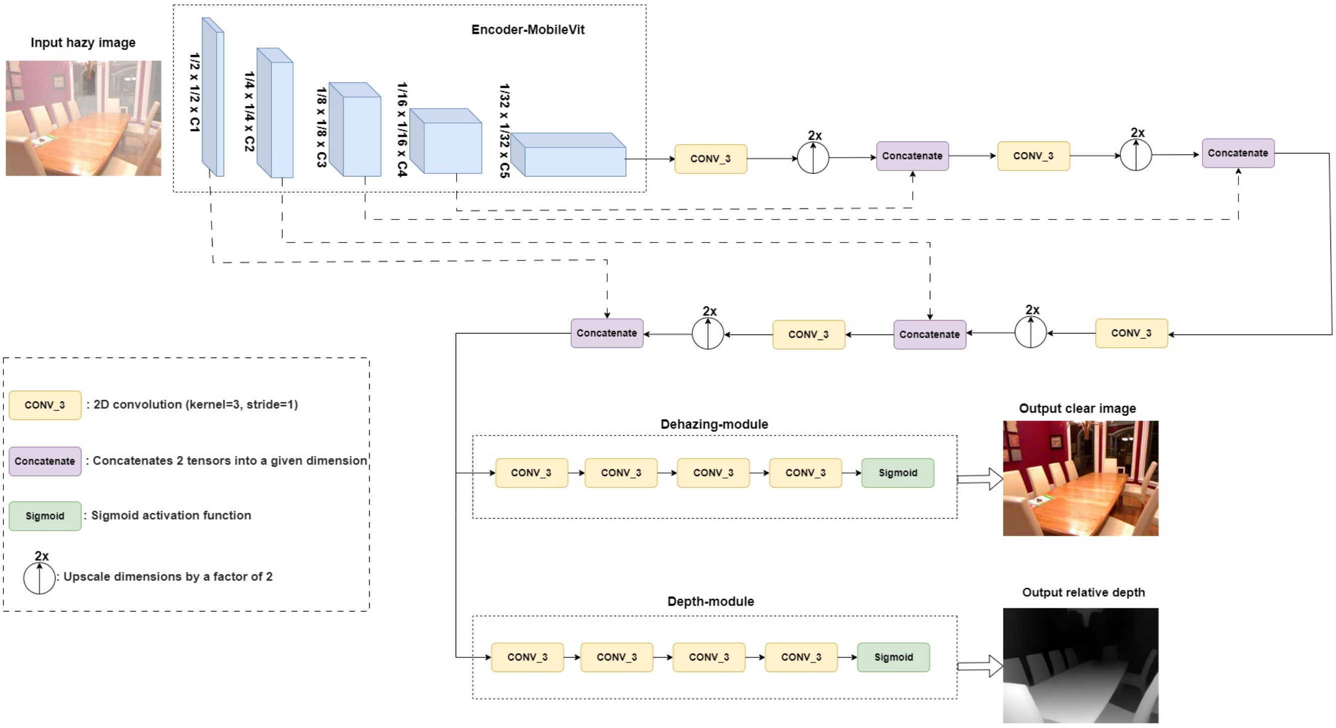


Figure 1: Model used for image dehazing utilizing relative depth. Each feature extraction block of MobileVit is concatenated with the output at different stages of the decoder. All the convolutions have kernel size=3 and stride=1.

3.2 Decoder

The decoder is the part of the network where the information provided from the features extracted from the encoder is used to reconstruct both the clear image and the relative depth map. The features from the MobileVit blocks are passed through a series of convolution, upsampling, and concatenation. As depicted in Fig.1 the first input of the decoder is the output of MobileVit with dimensions: $\frac{1}{32} \cdot H \times \frac{1}{32} \cdot W \times C5$. After undergoing a 3×3 convolution, the output is upsampled using Bilinear Interpolation. This enables the concatenation of features with those extracted from Block 4, as they share the same dimensions, specifically $\frac{1}{16} \cdot H \times \frac{1}{16} \cdot W$ of the original input. This process iterates through all blocks until the output matches the dimensions of the input hazy image ($H \times W$). The rationale behind this approach lies in the simultaneous presence of dehazing and depth feature information, which mutually reinforce each other, thereby enhancing overall performance.

After that, the decoder is split into 2 modules, a dehazing-module and a depth-module. The dehazing module utilizes this mixed information to refine its image reconstruction process, ensuring that the final output is visually coherent and faithful to the input. Similarly, the depth module benefits from this combined information to achieve more discernible depth cues.

3.3 Loss Function

Let y_{gt} (ground truth) be the clear image, y_p the predicted dehazed image, d_{gt} (ground truth) relative depth values from Depth Anything and d_{pred} the predicted depth values from the model. For single image dehazing $L1$ loss was chosen and for depth estimation a combination of 2 losses was implemented Structural Similarity ($SSIM$) [30] and $L1$ loss.

The dehazing and depth loss were combined to get the final loss function.

$$\mathscr{L}_{\text{Haze}}(y_p, y_{gt}) = |y_p - y_{gt}| \tag{2}$$

$$\mathscr{L}_{\text{Depth}}(d_p, d_{gt}) = \alpha \cdot \mathscr{L}_{\text{SSIM}}(d_p, d_{gt}) + \beta \cdot |d_p - d_{gt}| \tag{3}$$

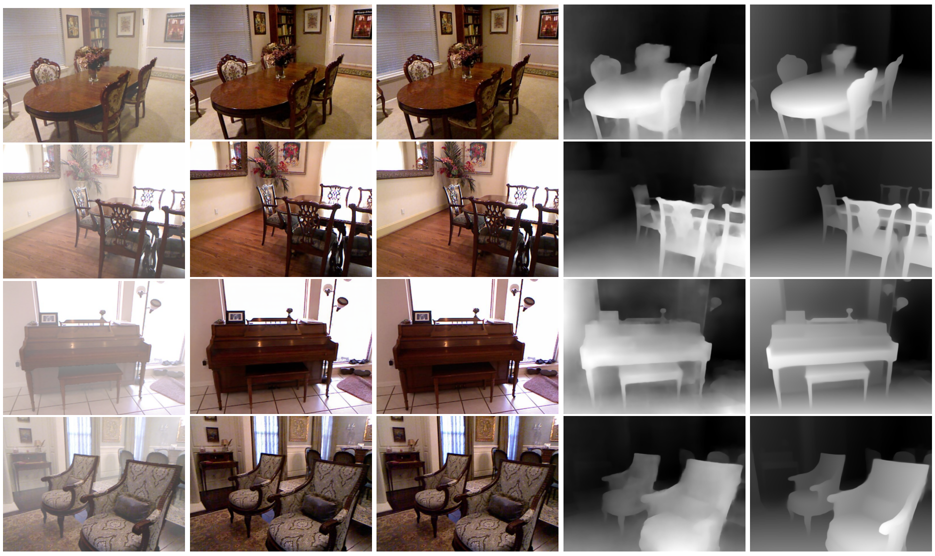
$$\begin{aligned}\mathscr{L}_{\text{Combined}}(y_p, d_p, y_{gt}, d_{gt}) = \gamma \cdot \mathscr{L}_{\text{Haze}}(y_p, y_{gt}) \\ + \delta \cdot \mathscr{L}_{\text{Depth}}(d_p, d_{gt})\end{aligned} \tag{4}$$

4 EXPERIMENTS

In this section, the evaluation of the proposed model is conducted using standard benchmarks for image dehazing. The presented implementation is compared with existing lightweight and heavyweight models.

Method	RESIDE-IN		RESIDE-OUT		RESIDE(IN+OUT)		Inference (ms)
	PSNR(M)	SSIM	PSNR(M)	SSIM	PSNR(M)	SSIM	
DCP [9] (2010)	16.627	0.818	19.13	0.815	17.875	0.816	-
MSCNN [26] (2016)	-	-	-	-	17.57	0.8125	-
AOD-Net [18] (2017)	20.51	0.816	24.14	0.920	22.325	0.868	-
FFA-NET [24] (2021)	36.39	0.9886	33.57	0.9840	34.98	0.8963	310.15
Light-DehazeNet [29] [29] (2021)	-	-	-	-	28.39	0.9487	9.28
MixDehazeNet-S [20] (2023)	39.47	0.995	35.09	0.985	37.28	0.99	131.62
Proposed Model	30.339	0.965	-	-	-	-	43.01

Table 1: Performance evaluation of the models on RESIDE-SOTS dataset [19].



(a) Hazy Image (b) Dehazed (c) Clear (d) Depth (e) GT depth [33]

Figure 2: Qualitative analysis on RESIDE-SOTS indoors dataset for both dehazing and depth [19]. The proposed model takes four hazy images as input. Ground truth (GT) depth is determined using DepthAnything outputs.

4.1 Implementation details

PyTorch [23] was the framework used for the implementation. A training regimen comprising 40 epochs was adopted, with an initial learning rate of 3×10^{-4} for the first 20 epochs, followed by a reduction to 3×10^{-5} for the next 20 epochs. Adam optimizer with default settings [15] and a batch size of 4 was utilized for both models.

4.2 Datasets

RESIDE [19] provides a comprehensive collection of hazy images, encompassing both real-world and synthetic scenes. Three subsets of the dataset were utilized: RESIDE-IN(ITS), comprising 13,990 hazy images along with their corresponding clear counterparts from indoor environments, RESIDE-OUT(OTS), comprising 50,874 hazy images and their corresponding clear images captured in outdoor settings, and Synthetic Objective Testing task (SOTS), consisting of 1000 clear images from indoor and outdoor scenes, each paired with its hazy counterpart.

For every clear image Depth anything [33] was employed to extract the relative depth map, serving as the ground truth value.

Two separate models were trained, one using the images from the indoor scenes and the other with the images from the outdoor scenes. For each model there are two ground truth values, the clear image and the relative depth map and one input value which is the hazy image.

Evaluation metrics: The standard evaluation metrics employed for single image dehazing are utilized.

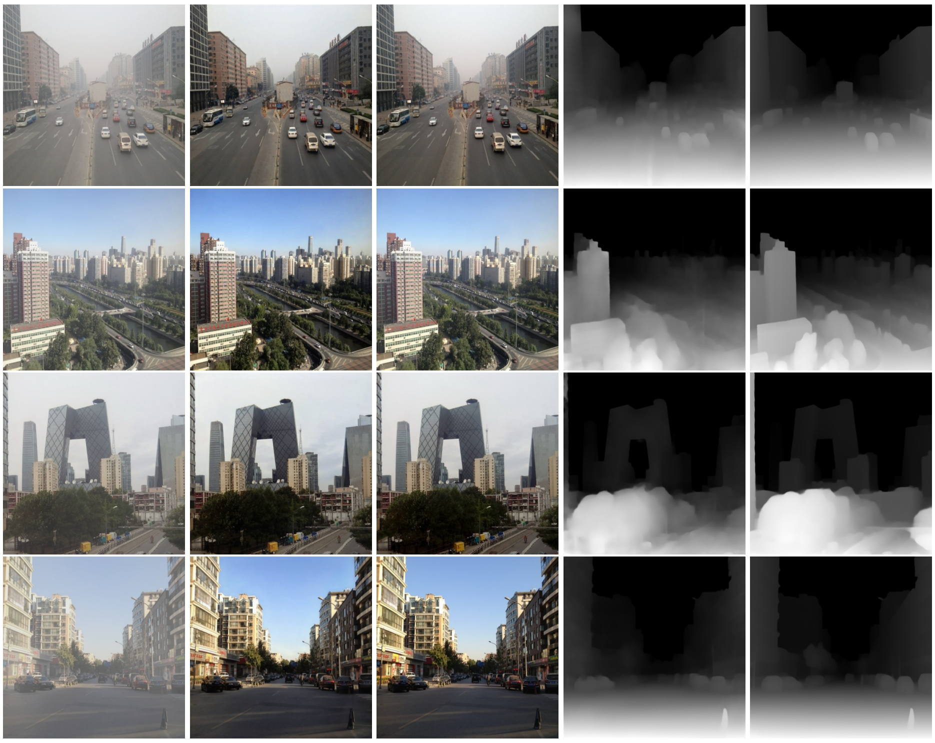
- Structural Similarity Index(SSIM):

$$\frac{(2\mu_{\hat{y}}\mu_y + C_1) + (2\sigma_{\hat{y}y} + C_2)}{(\mu_{\hat{y}}^2 + \mu_y^2 + C_1)(\sigma_{\hat{y}^2} + \sigma_y^2 + C_2)} \tag{5}$$

- Peak Signal-to-Noise Ratio (PSNR):

$$20 \cdot \log_{10} \frac{1}{RMSE} \tag{6}$$

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



(a) Hazy Image (b) Dehazed (c) Clear (d) Depth (e) GT depth [33]

Figure 3: Qualitative analysis on RESIDE-SOTS outdoors dataset for both dehazing and depth [19]. The proposed model takes four hazy images as input. Ground truth (GT) depth is determined using DepthAnything outputs.

4.3 Evaluation protocol

The model was tested on the indoor set which consists of 500 image pairs of RESIDE-SOTS (indoors) [19] at full resolution. The GPU used for the calculation of inference time is NVIDIA GeForce RTX 3060. The proposed model produces images at a fixed size, whereas the samples provided by RESIDE-SOTS (outdoors) exhibit varying resolutions. Consequently, direct comparison of metrics on the RESIDE-SOTS (outdoors) with other methods that output images at full resolution may lead to unfair assessments of performance.

4.4 Results

A selection of dehazed images and their corresponding relative depth values will be presented for qualitative analysis for both RESIDE-SOTS indoors Fig.2 and RESIDE-SOTS outdoors Fig.3. The implementations are compared to other models Table.1 and a quantitative comparison is provided Fig.4. While the proposed method is lightweight, the results exhibit remarkable clarity in both the dehazed and depth images, showing the efficiency and robustness of the approach. The qualitative evaluation with the heavyweight models reveals minimal disparities, whereas with lightweight models, the distinctions are prominently noticeable. In comparison to the implementations detailed in Table 1, the proposed model demonstrates better performance in terms of inference time, with the exception of LightDehazeNet. Notably, the proposed architecture stands out as the sole model providing the relative depth map, a factor that influences inference time.

4.5 Ablation study

To showcase the efficacy of the proposed architecture, an ablation study was conducted to analyze the number of need blocks of MobileVit and the loss function.

The model underwent testing where each of the four intermediate blocks of the Encoder was systematically omitted, allowing for an assessment of their individual impact on performance. Additionally, an evaluation without any of the intermediate blocks is provided. As shown in Table 2, the removal of any one of these four intermediate blocks yields negative effects on performance. The first block exhibits the most pronounced impact, while the fourth block shows the least. When

Blocks	PSNR	SSIM
1,2,3	29.958	0.963
1,2,4	30.195	0.963
1,3,4	30.039	0.962
2,3,4	28.373	0.931
No Blocks	22.179	0.691

Table 2: Performance Impact of removing intermediate encoder blocks

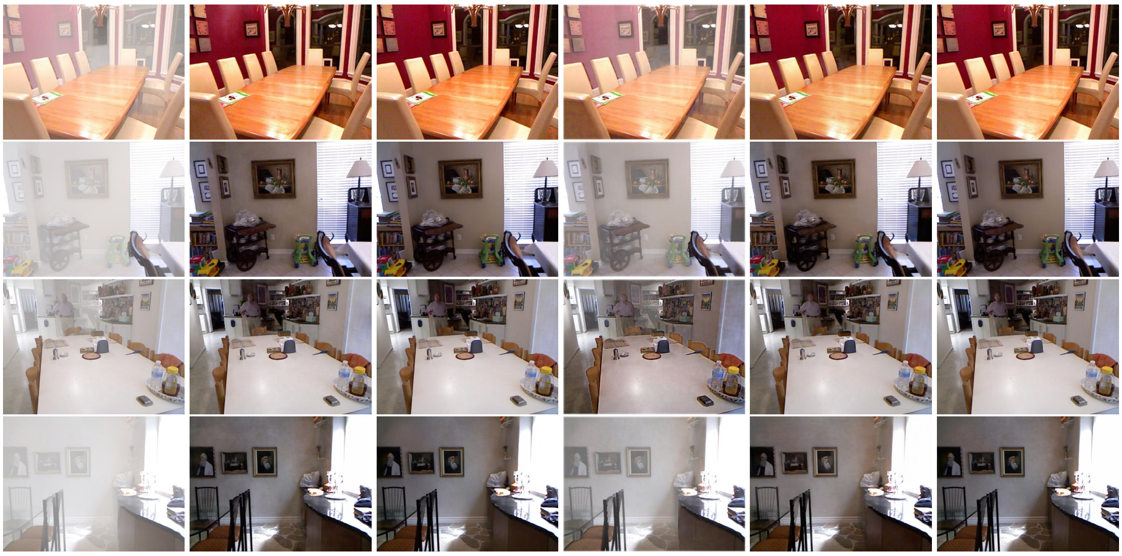

(a) Hazy Image (b) FFA-NET (c) MixDehazeNet (d) LightDehazeNet (e) Proposed model (f) Clear

Figure 4: Qualitative analysis on RESIDE-SOTS indoors [19].

all four blocks are discarded, the performance drops significantly, underscoring the value they provide.

We also conducted tests on the constants within the loss function (described in Table 3), which combines the haze loss with the depth loss.

γ	δ	PSNR	SSIM
0.5	0.5	29.02	0.958
0.6	0.4	29.572	0.960
0.8	0.2	30.339	0.965
0.4	0.6	28.56	0.947
0.2	0.8	28.05	0.941

Table 3: Impact of constants in the loss function

5 CONCLUSION

In conclusion, this paper presents a novel lightweight architecture tailored for single image dehazing, leveraging a MobileVit encoder for efficiency and speed, alongside a fully convolutional lightweight decoder featuring skip connections for enhanced feature extraction. By integrating depth estimation into the dehazing task within a single network in a supervised manner, depth information aids in scene understanding. Moreover, the shortage of datasets providing both dehazing and relative depth ground truths is overcome by employing state-of-the-art networks like Depth Anything for relative depth extraction. The main contribution lies in proposing a lightweight solution for image dehazing, utilizing MobileVit in the encoder, incorporating relative depth values to empower dehazing, and employing a fully convolutional decoder with skip connections for efficient and accurate processing. The relative depth output can be leveraged in various other computer vision tasks to enhance their performance and robustness. By providing depth information alongside dehazed images, the proposed model not only improves visual clarity but also enriches the data available for downstream tasks, thus contributing to more accurate and comprehensive computer vision solutions. Overall, this framework offers a promising implementation for enhancing scene understanding in challenging environmental condition

6 ACKNOWLEDGMENTS

This research has been supported by the European Commission funded program RESCUER, under H2020 Grant Agreement 101021836.

7 REFERENCES

[1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.

[2] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE transactions on image processing*, 25(11):5187–5198, 2016.

[3] Yuning Cui, Wenqi Ren, Xiaochun Cao, and Alois Knoll. Focal network for image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13001–13011, 2023.

[4] Raul de Queiroz Mendes, Eduardo Godinho Ribeiro, Nicolas dos Santos Rosa, and Valdir Grassi Jr. On deep learning techniques to boost monocular depth estimation for autonomous navigation. *Robotics and Autonomous Systems*, 136:103701, 2021.

[5] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang. Multi-scale

boosted dehazing network with dense feature fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2157–2167, 2020.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

[8] Chun-Le Guo, Qixin Yan, Saeed Anwar, Runmin Cong, Wenqi Ren, and Chongyi Li. Image dehazing transformer with transmission-aware 3d position embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5812–5820, 2022.

[9] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2010.

[10] 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*, pages 770–778, 2016.

[11] Ming Hong, Yuan Xie, Cuihua Li, and Yanyun Qu. Distilling image dehazing with heterogeneous task imitation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3462–3471, 2020.

[12] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[13] Hyungjoo Jung, Youngjung Kim, Dongbo Min, Changjae Oh, and Kwanghoon Sohn. Depth prediction from a single image with conditional adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1717–1721. IEEE, 2017.

[14] Doyeon Kim, Woonghyun Ka, Pyungwhan Ahn, Donggyu Joo, Sehwan Chun, and Junmo Kim. Global-local path networks for monocular depth estimation with vertical cutdepth. *arXiv preprint arXiv:2201.07436*, 2022.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Dong-Jae Lee, Jae Young Lee, Hyunguk Shon, Eojindl Yi, Yeong-Hun Park, Sung-Sik Cho, and Junmo Kim. Lightweight monocular depth estimation via token-sharing transformer. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4895–4901. IEEE, 2023.

[17] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[18] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE international conference on computer vision*, pages 4770–4778, 2017.

[19] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2018.

[20] LiPing Lu, Qian Xiong, DuanFeng Chu, and BingRong Xu. Mixdehazenet: Mix structure block for image dehazing network. *arXiv preprint arXiv:2305.17654*, 2023.

[21] Earl J McCartney. Optics of the atmosphere: scattering by molecules and particles. *New York*, 1976.

[22] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[24] Xu Qin, Zhilin Wang, Yuanchao Bai, Xiaodong Xie, and Huizhu Jia. Ffa-net: Feature fusion attention network for single image dehazing. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11908–11915, 2020.

[25] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[26] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via multi-scale convolutional neural networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 154–169. Springer, 2016.

[27] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3253–3261, 2018.

[28] Michael Rudolph, Youssef Dawoud, Ronja Güldenring, Lazaros Nalpantidis, and Vasileios Belagiannis. Lightweight monocular depth estimation through guided decoding. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2344–2350. IEEE, 2022.

[29] Hayat Ullah, Khan Muhammad, Muhammad Irfan, Saeed Anwar, Muhammad Sajjad, Ali Shariq Imran, and Victor Hugo C de Albuquerque. Light-dehazenet: a novel lightweight cnn architecture for single image dehazing. *IEEE transactions on image processing*, 30:8968–8982, 2021.

[30] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[31] Diana Wofk, Fangchang Ma, Tien-Ju Yang, Sertac Karaman, and Vivienne Sze. Fastdepth: Fast monocular depth estimation on embedded systems. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6101–6108. IEEE, 2019.

[32] Haiyan Wu, Yanyun Qu, Shaohui Lin, Jian Zhou, Ruizhi Qiao, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. Contrastive learning for compact single image dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10551–10560, 2021.

[33] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024.

[34] Dong Zhao, Jia Li, Hongyu Li, and Long Xu. Complementary feature enhanced network with vision transformer for image dehazing. *arXiv preprint arXiv:2109.07100*, 2021.

Correlational Data Visualizations with Colored Bar Charts

Bdour Alwuqaysi
King's College London
bdour.alwuqaysi@kcl.ac.uk

Alfie Abdul-Rahman
King's College London
alfie.abdulrahman@kcl.ac.uk

Rita Borgo
King's College London
rita.borgo@kcl.ac.uk

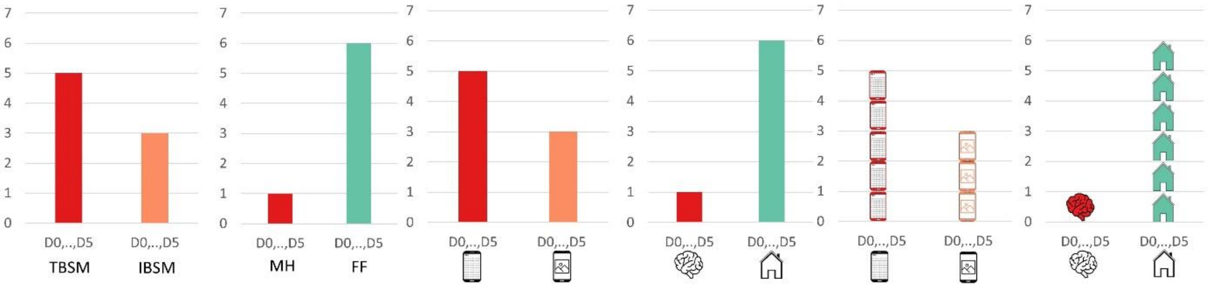


Figure 1. Examples of study stimuli that show variations of colored vertical bar charts.

ABSTRACT

This study evaluates the efficacy of three varied designs of colored bar charts—featuring text annotations, icon shapes, and stacked icons—in illustrating the correlation between social media use, mental health, and family functioning. It seeks to bridge a research gap by exploring this specific aspect of visualization techniques. The study uses an online survey with 500 participants to evaluate these designs in terms of accuracy, response time, and user preferences. The research findings suggest that colored bar charts with text can reduce response time in medium complexity tasks. Meanwhile, colored bar charts with text and stacked icons can enhance accuracy in medium and hard complexity tasks. Colored bar charts with icons can engage users more in medium and hard tasks, and consistently demonstrate high interaction. The study revealed that gender could influence response time and interaction, with colored bar charts with icons generally preferred. However, visualization preferences may vary across age groups, highlighting the importance of personalized visualizations for diverse users. These insights are crucial for individuals aiming to utilize visualizations effectively for correlational data, prompting further investigation into enhancing data visualization through subsequent research based on these results.

Keywords

Correlational data visualizations, Colored bar charts, Text, Icons, Stacked icons.

1. INTRODUCTION AND BACKGROUND

In the domain of data visualization, the design of visual representations significantly influences how users interpret and engage with data. Previous research has underscored the importance of intuitive and visually appealing visualizations in facilitating comprehension and decision-making [1, 2, 3]. However, the relationship between visualization complexity and user behavior remains complex and multifaceted. Correlational data representation studies offer a unique opportunity to explore this relationship by examining how variations in visualization design correlate with user responses. Understanding these correlations is essential for creating visualizations that effectively convey information while minimizing cognitive load and enhancing user engagement. Moreover, demographic factors such as gender, and age can further influence user preferences and interactions with visualizations. Studies have shown that individuals from different demographic backgrounds may have varying cognitive styles and preferences when interacting with visual information [4, 5]. Therefore, exploring how demographic factors intersect with visualization design is crucial for ensuring the inclusivity and effectiveness of visualizations across diverse user groups. By focusing

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.

on correlational data, this study seeks to uncover relationships between visualization complexity, user behavior, and demographic characteristics. In recent years, visualization techniques has become increasingly important in improving user comprehension and engagement with data. Among these techniques, colored bar charts with text, colored bar charts with icons, and colored stacked icons with icons have emerged as methods for conveying information. The effectiveness of some of these visualization techniques in facilitating user understanding and engagement has been demonstrated in various studies [6, 7, 8]. However, the impact of these techniques on user preferences and interactions, particularly in correlation with demographic factors such as gender, and age remains an area of exploration. Thus, this study aims to explore the relationship between correlational data visualization designs and user responses regarding demographic characteristics.

Contributions:

The study revealed several key findings. Firstly, a colored bar chart with text was more effective in reducing response time than a colored bar chart with icons, especially in medium complexity designs. Secondly, a colored bar chart with text and colored stacked icons showed greater accuracy compared to a colored bar chart with icons in both medium and hard complexities. Thirdly, the colored bar chart with icons was the most engaging and efficient design for completing tasks. Particularly in medium and hard complexities, with increasing repetitions, response time decreased across all design complexities. The colored bar chart with icons consistently shows higher interaction levels, highlighting the potential to enhance user engagement and learning efficiency. Furthermore, the study found gender variations in the responses. Males exhibited lower response times in certain designs. Preferences also varied among participants. The colored bar chart with icons was the most preferred due to its appeal and understandability. In contrast, the colored stacked icons with icons were favored for their memorability and potential for triggering behavioral changes. The majority favored the colored bar chart with icons. Lastly, younger participants tended to prefer the colored bar chart with icons, while older individuals favored the colored bar chart with text.

2. METHODOLOGY

An online study was conducted to investigate the topic. Participants have started with detailed training. This was followed by the main segment of the study, consisting of stimuli and questions prompting individuals to conclude from the presented visualizations. Inspired by the current methodology in vision science [9], we deliberately selected methods employing a familiar chart type (bar chart) as the foundation for stimuli design and tasks to enhance relevance, simulating common data visualization interactions. The process began with creating various visualizations to represent our topic-specific correlational data design, emphasizing having these visualizations evaluated by the participants. Efficient data visualization (i.e., high accuracy and short response time) requires careful comparison of encodings and layouts, accurate value estimation, magnitude sense, and subdivision within charts [10]. Our core stimuli tasks were visual search, and comparison. We also combined data from two charts with inverted color scales.[1]

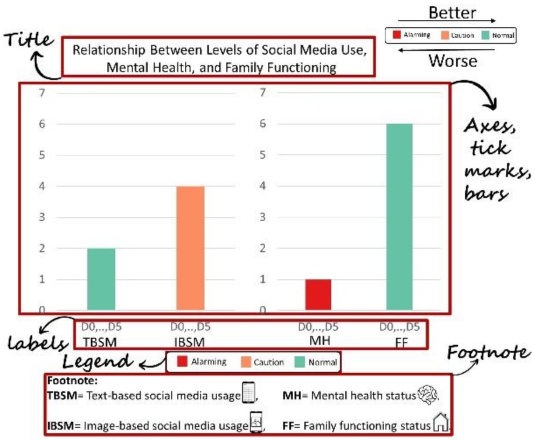


Figure 2. Training was provided to participants to help them understand each visual encoding.

2.1 Study Structure

The online study was conducted using the Qualtrics platform [11], Participants received a comprehensive information sheet and consent form explicitly stating their right to withdraw from the study at any point. A computer screen was a prerequisite for participation, and individuals were required to have normal or corrected-to-normal vision without any color-vision deficiencies. At the outset, two questions utilizing the Ishihara test [12], a widely recognized test in visualization studies involving colors, were employed to identify red-green color deficiencies among participants. To ensure the accuracy of the results, we excluded responses to the two color-deficiency screening questions from subsequent data analysis. An additional attention check question was used to verify the survey results, but it was not included in the data analysis. No personally identifiable information was collected, and all data were securely stored on a

---

[1] Refer to the supplementary material for additional details: https://anonymous.4open.science/r/Vis-5C01/Qualtrics%20survey%20A.pdf

private server at the author's institution. The study comprised three main sections. Participants were asked seven demographic questions followed by three questions about their social media usage after signing the consent form. In the study's second section, introduced. To ensure participants understood the concepts, definitions of key terms were provided customized training and visualizations were along with a sample question they needed to answer correctly before moving on. The training helped participants understand each visual encoding by presenting charts, annotations, and brief textual explanations, i.e., The participants were provided with to clarify the meaning of 'Better' and 'Worse' in addition to the definition of these concepts in the context of the study (see Fig.2). Although the questions from the training phase were not included in the data analysis, they were used as a quality assurance measure of the results. A clear separation was added between the training and visualization sections to prepare and prompt the participants. The visualization section included three design variations presented at three complexity levels (easy, medium, and hard), with two repetitions to validate results. Participants confirmed their readiness to proceed to the next block, allowing breaks as needed. Response time (TSubmit), interaction-user engagement (Click Count), and accuracy (Percentage of Wrong Answers) were assessed for each visualization. The participants were asked to choose the visualizations based on their understandability, appeal, memorability, and identification of the most useful visual design feature when answering questions. Two levels of randomization were used to increase result validity: one within each section's complexity and another within each question's answer choices.

Screening: Participants were started with seven demographic questions to gather information about gender, age, ethnicity, education level, employment status, field of expertise, and any prior experience in visualization. Only participants with normal or corrected-to-normal vision without any color-vision deficiencies were included to ensure a consistent and valid sample. We do this by asking participant to correctly answer the two Ishihara test questions to be included in the analysis. Participation in the study was restricted to individuals using a laptop or desktop computer with an internet connection, facilitating access to the web browser required for the study. Data from 39 participants were excluded due to failure to pass attention checks, missing responses, or experiencing a server interruption during the study. The data collection process continued until a high-quality dataset was achieved (i.e., characterized by a balanced sample across genders).

2.1.1. Stimuli Design: Visual stimuli for this online study were created using draw.io, a cross-platform graph drawing software developed in HTML5 and JavaScript. Stimuli were saved as static images and presented to participants using the Qualtrics platform. Visual task performance was tested over the four manipulation elements in the stimuli: data density, chart type, color scale, and task difficulty.

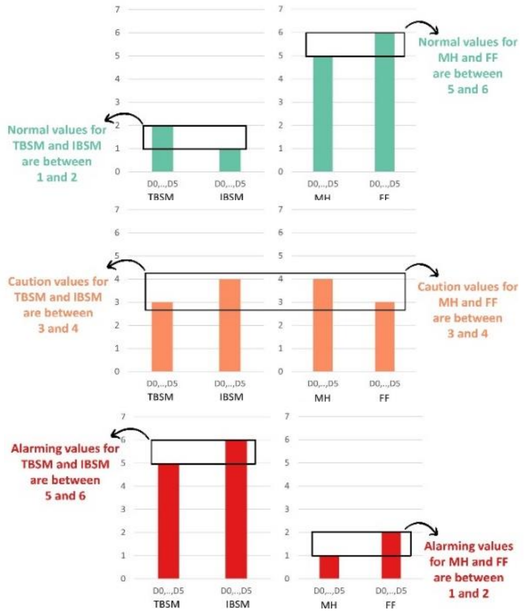


Figure 3. The color scale was used in this study.

Data: The data used in the visualizations of this study was real-world data collected from an Android application that the researcher had previously developed in 2023. This application was used to collect self-reported data on mental health and family functioning, and track social media usage for a month. The data has already been analyzed and proven correlational [13], making it the wel-suited dataset for this study. This study uses a bar chart approach to display two sets of data side-by-side with different color scales. The left bar chart (LC) depicts the correlational data of individuals' text-based social media use (TBSM) and image-based social media use (IBSM) for six days, while the right bar chart (RC) displays the anonymous records of mental health status (MH) and family functioning status (FF) for the same six days (see Fig.1). The data in our study are represented by integer numbers floating points are rounded to the nearest integer value. The data is presented at three different levels of density, with two, four, and six points on each chart, making a total of 4, 8, and 12 points across the two bar charts in each correlational data representation (see Fig.4). Each presented stimulus was unique, representing a new data record without repetition or manipulation. The human mind only holds around 7 points in immediate memory and absolute judgment. Going over this limit increases cognitive load [14]. Therefore, we chose set sizes of 8 and 12 for our subsequent data points. To align with the left-to-right reading sequence of English readers, we placed social media use charts on the left

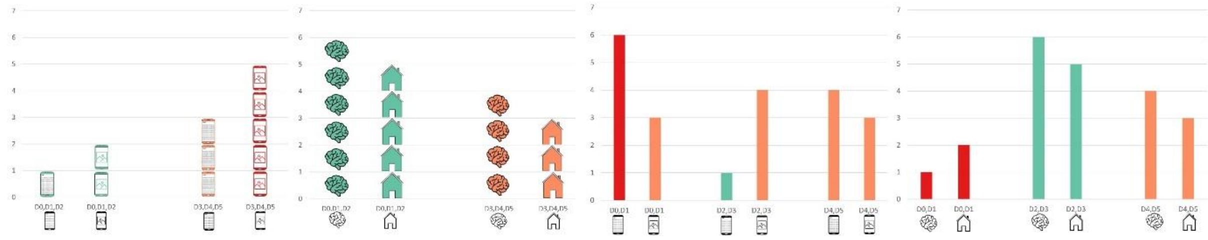


Figure 4. Stimulus density level 8 & 12 examples.

to create a logical flow. This arrangement establishes a clear narrative progression by introducing social media variables before revealing their impact on mental health and family functioning. This approach helps readers understand the correlational data represented. The LC begins with TBSM and progresses to IBSM, following a logical and chronological sequence. This mirrors the provided information and emphasizes the foundational nature of TBSM. On the RC, MH precedes FF, emphasizing that personal mental health serves as a precursor to the health of the family unit. Arranging similar variables together facilitates easy comparison, helping readers evaluate the impact of social media on mental health and family function. Aligning charts with corresponding social media types underscores potential causal relationships to enhance comprehension. We varied set sizes across charts while maintaining consistent density for side-by-side comparison. As detailed below, we also gathered visualization assessments, including selected answers, response times, and participant preferences of each chart type.

Chart Components: The charts in this study have the same title, axes, tick marks, legend, and footnote. Although they have similarities, the three designs differ in their use of bars and labels. Design I features colored bars with text, Design II uses colored bars with icons, and Design III displays colored stacked icons with icons, as shown in Fig.1 and Fig.2.

Chart Types: Design I colored bar charts with text offer a simple way to present data, combining visuals with annotations for context [15]. Design II colored bar charts with icons allow for quick identification of categories [16, 17]. Design III colored stacked icons with icons are used for complex datasets, allowing for easy interpretation of multidimensional data [18].

Color Scale: The color scales employed in our correlational data bars representation differ between the LC, representing TBSM and IBSM, and the RC, representing MH and FF for individuals (see Fig.3). The green bars on LC denote normal values between 1 and 2, where lower social media consumption is considered better, as proven in the literature [19]. In contrast, normal values on RC fall between 5 and 6, where higher scores of MH and FF represent the good state of individuals based on the literature [20, 21]. The orange color on both LC and RC indicates caution values between 3 and 4, displaying intermediate TBSM, IBSM, MH, and FF values. The red bars on LC signify alarming values between 5 and 6, where increased social media consumption is considered worse, as proven by the literature [19]. Conversely, alarming values on RC are between 1 and 2, where lower scores of MH and FF represent the worst state of individuals, according to the literature [20, 21]. Inspired by the color theory principles [22], we designed the bars in different color variations to create a clear and effective visual display. Red is utilized to signify alarm and danger, while green represents normalcy and calmness. The orange is used to indicate caution and alertness. These color choices aim to elicit specific emotional responses and enhance the understanding of the information presented in the visualizations [23]. The shades of these colors have been selected from the Colors Brewer tool [24] to ensure friendliness to individuals with color blindness. Redundant color encoding used in this study can help differentiate data points more effectively. Combined with other visual variables, it creates unique visual patterns that aid pattern recognition and data segmentation. This improves visualization clarity, especially in dense contexts [22].

2.1.2. Task Design

The study used a primary task of visual search process and comparison, an essential element in interpreting visualizations. The search process task involved searching over two channels (i.e., two side-by-side charts with the same density but different color scales). The difficulty of this search process increased with the increase in object density. The study aimed to assess participants' performance in three different types of charts. There were three levels of density for each chart type.

Additionally, there were three multiple-choice questions with an alternative forced-choice format. The questions were designed to increase difficulty and compare the performance of the participants across varying levels of difficulty. These questions were categorized into Easy, Medium, and Hard, and the response options were randomized across all questions. In most questions, two answers are partially incorrect, while the third is entirely correct. This approach ensures the results are valid and accurate (refer to the supplementary material for more details).

Easy Question: The easy question involves a search and comparison task. It displays two side-by-side charts with values from the same set of data (i.e., (LC: TBSM, IBSM), (LR: MH, FF)) across 6 days. This question is classified as easy because it involves searching for specific target data. Once located, participants compare the target data across the charts. For instance, participants searched for IBSM in the chart and compared its value to the second set of data TBSM to determine if cases were Worse, Same, or Better over the same 6-day period. Example of an Easy Question: According to the above chart, is IBSM Worse, Same, or Better than TBSM? Easy Responses: IBSM is Worse than TBSM, IBSM is Same as TBSM, IBSM is Better than TBSM.

Medium Question: The medium question is a search and comparison task across an increased number of targets. It displays two side-by-side charts with values from two sets of data (i.e., (LC: TBSM, IBSM), (LR: MH, FF)) across 6 days. This question is classified as a medium as it increases the data set for search and comparison to two targets across both charts; participants search for the Worse, Same, or Better values and extract the category name and day for the Worse, Same, or Better value. Medium question example: Based on the above chart, which category and what days have the Better values? The options are provided to you in the format of (Category-days). Medium responses are (IBSM-D0,D1,D2) and (MH-D3,D4,D5), (TBSM-D3,D4,D5) and (FF-D0,D1,D2), (IBSM-D3,D4,D5) and (FF-D3,D4,D5).

Hard Question: The hard question is a search and comparison task that involves mathematical estimation. It displayed two side-by-side charts with values from two sets of data (i.e., (LC: TBSM, IBSM), (LR: MH, FF)) across 6 days. This question is classified as hard as after searching for the four target data sets across charts, participants had to perform compare them with mathematical symbols to estimate the answers. Participants must estimate the numerical Normal, Caution, and Alarming values in this question. Hard question example: In the chart above, what are the numerical Normal values for each individual column in different categories, where '&' represents the logical 'and' operation? Hard responses: TBSM & IBSM >6 and MH& FF >4, TBSM & IBSM <3 and MH& FF >4, TBSM & IBSM <3 and MH& FF >2.

2.1.3 Measures

We designed 27 multiple-choice questions by combining colored vertical bar charts on either right (RC) or left (LC), three levels of data density, and three question types/difficulty levels. To evaluate the performance, we recorded the chosen answer (Accuracy), response time (RT), and interaction (Click Count) for each question in the task parts of the study. We presented the stimuli randomly to participants to reduce any learning effects (e.g., bias in selection).

2.1.4 Preference

We used a two-phase question approach to determine the preferences of the participants. The first question asked the participants to evaluate various aspects of the designs presented, such as their favorite design, the one that was easiest to understand, the most visually appealing, the most memorable, and the potential for triggering behavioral change towards social media consumption. The participants had the option to choose from the provided designs or select "none." The second question aimed to identify the specific feature within the designs the participants relied on the most during the study. The options included colored bars with text, colored bars with icons, icons bar with icons, or an alternative option where participants could specify their choice. This approach provided a detailed understanding participant's self-reported preferences and the factors that influencing their decision-making process.

2.2 Recruitment

Participants were recruited through Prolific [25].To ensure the quality of our sample, participants underwent a pre-screening process, confirming normal vision and the absence of any deficiencies. Additionally, the Ishihara Color Test was employed as an extra measure to identify and exclude individuals with red-green color blindness, a common consideration in visualization studies [26]. The eligibility criteria included being 18 years and older, and fluency in English. We successfully achieved a balanced sample across genders, totaling 500 participants. Compensation adhered to Prolific's fair pay policy, with participants receiving £7.85 per hour. The average response time was 28 minutes and 39 seconds. A computer screen was explicitly required for participants to participate in the study. Data collection spanned six days, covering various time slots to enhance the diversity of potential participants. The study began on January 31, 2024, and concluded on February 5, 2024.

2.3 Participants

500 responses were collected, with approximately equal gender distribution (256 males, 252 females, and less than 1% others). The group of Gender “Others” has very low frequency (only 4); thus it is excluded from further statistical analysis. The age distribution reveals a significant representation in the 18-24 age group (23.05%) and a dominant presence in the 25-34 age group (49.02%). In this sample, individuals of Black ethnicity make up nearly a quarter (24.61%), while those of White ethnicity constitute about two-thirds (62.5%). Individuals with Mixed and Asian ethnicity make up around (5%). Arab ethnicity is less than (1%), and other ethnic groups comprise less than

(3%). Education-wise, the majority hold either a Bachelor's degree (42.77%) or a Master's (21.48%). Employment status indicates two-thirds are employed (50.2% full-time, 12.3% part-time, 8.79% self-employed), (15%) are students, and about (10%) are not employed. Participants span 27 fields, with the most prevalent being Business, Management, Marketing, and Related Support Services (15.23%), Computer and Information Sciences and Support Services (15.23%), Engineering Technologies and Engineering-Related Fields (10.74%), and Education (8.01%). Additionally, around (25%) of the population is distributed across various other fields.

2.4 Statistical Analysis

Python was used for most statistical analysis, except for Fisher’s exact test and ranked two-way ANOVA, which were performed using R. Categorical variables were described by frequency and percentage, while numerical variables were presented as mean ± standard deviation, and median [minimum-maximum]. Before analysis, variables with low-frequency categories were either re-categorized or excluded. ANOVA was used to compare numerical variables among more than two groups. Welch ANOVA was used if homogeneity assumptions were violated, and Kruskal-Wallis test was applied if normality assumptions were violated. Pairwise comparison tests were performed for significant results. Two-way ANOVA on rank-transformed data was used to compare numerical variables among two grouping levels. A p-value less than 0.1 was considered significant in all tests.

2.5 Pilot Testing

We conducted a pilot test to identify potential refinements related to the length and complexity of the study. Based on the pilot results, specific questions in the study are reformatted for increased clarity and simplicity. As part of our efforts to enhance the overall understandability of the study's questions, we have included definitions for a few key concepts. These definitions are provided not only when the concepts are initially introduced but also when they are referenced in the questions.

2.6 Ethics

The user study described in this paper and other study that provided the data, have been approved by the Research Ethics Committee of the researcher institution (anonymized for submission).

3. RESULTS AND DISCUSSION

The evaluation results of all designs are summarized in Table 1 and elaborated further below. Our analysis has revealed that gender significantly affect response time and interaction. Moreover, age was significant factors affecting preferences, as evidenced by the p-values below.

3.1 Response Time

Designs Complexity: Participants' response times vary based on the complexity of the design. In easy designs, response times remain consistent across different designs. However, in medium designs, participants took less time on Design I (42.55 ± 72.92) than Design II (49.85 ± 59.86), showing a significant difference in TSubmit. Design III falls between Design I and II but is not significantly different from either. In hard designs, Design II and III (44.55 ± 40.71 and 44.0 ± 69.3 respectively) have significantly higher TSubmit than Design I (36.85 ± 42.16). These findings offer insights into how the complexity of task design influences response times. The findings regarding response time shed light on how design complexity impacts participants' performance. Consistent with previous literature, easy designs yielded steady response times across different designs, indicating minimal cognitive load [27].

Conversely, medium and hard designs demonstrated significant variations in response time, aligning with studies emphasizing the influence of task complexity on cognitive processing [28]. The observed differences between Design I and Design II in medium and hard designs suggest that certain design elements may pose greater cognitive challenges, resonating with prior research on the impact of design features on user response [29]. These insights underscore the importance of considering design complexity in visualization to optimize user efficiency and task performance.

Table 1. Evaluation factors across design complexity performance.

Evaluation Factors	**Design Complexity Performance**		
Measures	**Best**	**Intermediate**	**Worst**
Response time	I	III	II
Interaction	II	I	III
Accuracy	I	III	II
Self-reported			
General Preference	II	I	III
Understandability	II	I	III
Appeal	II	III	I
Memorability	III	II	I
May Trigger Behavioral Changes	III	II	I

Repetition: There is a significant decrease in TSubmit across all designs as repetitions increase, indicating participants' learning and familiarity with the target task. In easy designs, the median TSubmit in the last repetition (R3) (11.32 [1.83:753.34]) is nearly half of that in the first repetition (R1) (20.36 [3.45:1464.75]). In medium designs, R3 median TSubmit (20.25 [1.47:1314.16]) is less than half of R1 (52.49

[2.91:1728.93]), and in hard designs, R3 median TSubmit (23.55 [1.72:892.73]) is nearly half of R1 (42.54 [2.17:1728.5]). This highlights the impact of repetition on response time, showing participants' increased efficiency with task familiarity. This is consistent with findings in skill acquisition literature [30]. The observed result supports the notion that repeated exposure leads to improved performance and reduced cognitive effort [31].

Gender: A significant difference in TSubmit was observed among different designs based on gender. In easy design, where the highest TSubmit median is for Design III in both genders (male: 22[5-223] & female: 23[4-301]) when compared to Design I (male: 19[3-1465] & female: 20[7-665]) and Design II (male: 17[5-636] & female: 20[3-468]). It is clear from the interaction plot (see Fig.5) that males and females spent nearly the same amount of time in Design I and III, but males spent much lower TSubmit in Design II. The difference in TSubmit among genders is significant, although the interaction is not. Also, in medium and hard designs, the gender as well as the designs are significant as in medium designs the TSubmit in Design II (male: 65[3-592]& female: 72[5-589]) higher than other designs and in hard designs, Design I has the lowest TSubmit (male: 34[2-436]& female: 40[2-489]) while Design II has the highest TSubmit (male: 52[2-207]& female: 52[2-527]). The gender-based differences in response time highlight the nuanced interaction between design complexity and user characteristics (i.e., gender). Consistent with previous research on gender disparities in cognitive processing [32], e.g., information processing styles [33]. These findings highlight the importance of adopting inclusive design practices that accommodate diverse user demographics and cognitive preferences, aligning with the universal design principles [34]. Further research exploring the underlying factors driving these differences could provide deeper insights into designing more inclusive and user-centric visualizations.

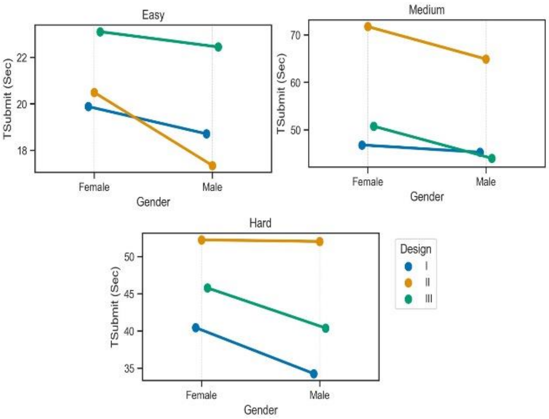


Figure 5. Interaction plot of TSubmit among different genders.

3.2 Interaction

Designs Complexity: In easy designs, there is no significant difference in Click Count among different designs. However, in medium designs, Click Count is significantly lower in Design I and Design III (2.79±2.19 and 2.82±2.34, respectively) than in Design II (2.97±2.86). Notably, Design I has the lowest maximum Click Count (2.0[2.0:23.0]), which is nearly half the maximum of Design II (2.0[2.0:55.0]). Furthermore, in hard designs, Design I has a significantly lower Click Count (2.87±2.34) when compared to Design II (3.01±2.64). In easy designs, users interact similarly. However, medium-level designs have notable differences. Design II has higher interaction and Click Count, making it more effective in engaging users. This highlights its potential to enhance user engagement and lerning efficiency. In easy designs, where the cognitive load is minimal, no significant disparity in Click Count among different designs is observed, aligning with prior studies emphasizing simplicity in user interactions [35]. However, significant differences emerge in medium and hard designs, with Design II exhibiting a higher Click Count than other designs. This finding resonates with literature highlighting the importance of interactive elements in engaging users and facilitating task learning [36].

Repetition: In easy designs, there is no significant difference in Click Count among different designs. However, in medium designs, Click Count is significantly lower in Design I and Design III (2.79±2.19 and 2.82±2.34, respectively) than in Design II (2.97±2.86). It is worth noting that Design I has the lowest maximum Click Count (2.0[2.0:23.0]), which is nearly half the maximum of Design II (2.0[2.0:55.0]). Furthermore, in hard designs, Design I has a significantly lower Click Count (2.87±2.34) when compared to Design II (3.01±2.64). Across different complexities, the impact of repetition on Click Count is observed, particularly in medium and hard designs. Consistent with theories of skill acquisition and automation [37], participants exhibit increased interaction efficiency with task familiarity, as shown by higher Click Count in Design II across all complexities. Thus, the findings underscore the importance of iterative design approaches that facilitate user learning and skill development over time.

Gender: It is clear that in all designs females have a significantly higher Click Count than males. The average Click Count in females in all designs 3±2 is significantly higher than in males 2±1. Moreover, there is a significant difference among designs where Design I has the lowest average Click Count (male: 2.6±2 & female: 3.2±3), whereas Design II has the highest average Click Count (male: 2.8±2 & female:

4.2±6). There is a significant difference between gender, where males have a significantly lower Click Count in all designs (Design I: 2.5±1, Design II: 2.8±2 and Design III: 2.7±2) than in females (Design I: 3.3±4, Design II: 3.4±4 and Design III: 3.3±3). The gender-based differences in Click Count reveal disparities in user interaction behavior. Females consistently exhibit higher Click Count across all designs, suggesting potential gender-specific differences in interaction patterns. These findings align with previous research highlighting variations in cognitive processing styles between genders [38]. Thus, the observed differences in Click Count among designs underscore the importance of considering gender-specific preferences and cognitive styles in visualization design [39].

3.3 Accuracy

Designs Complexity: The analysis of accuracy rates across different design complexities reveals significant differences in task performance. In easy designs, where the accuracy rates are consistent among all designs with around (6%) of wrong answers, no significant variation observed. However, substantial differences emerge in accuracy rates among different designs in both medium and hard complexity designs. In medium designs, Design I and Design III demonstrate significantly lower percentages of wrong answers (10.2% and 11.5%, respectively) than Design II (19.5%), indicating their effectiveness in facilitating accurate task learning.

Similarly, in hard designs, Design I exhibits the lowest percentage of wrong answers (15.4%) compared to Design II (17.3%) and Design III (21.1%), suggesting it outperforms task accuracy. Therefore, Design I is the preferred option across medium and hard complexities based on its consistently higher accuracy rates than the other designs. Examination of accuracy rates across varying design complexities sheds light on the nuanced relationship between design complexity and user performance. In easy designs, where the cognitive load is minimal, consistent accuracy rates are observed across all designs, aligning with prior research highlighting the simplicity of tasks in this category [40]. However, in both medium and hard complexity designs, significant differences in accuracy rates appeared, suggesting the impact of design on user performance in completing target tasks. These findings resonate with studies emphasizing the importance of design optimization in enhancing user performance and satisfaction [41]. This underscores the critical role of design considerations, particularly in complex tasks, where design choices significantly influence task accuracy and user experience.

Repetition: In easy designs, there is no significant difference in accuracy between all designs among all repetitions. In medium design, Design II has the lower accuracy in all repetitions as it has the highest percentage of wrong answers in R1 157(10.4%), which is more than twice that in Design I and Design III [68(4.5%) and 74(4.9%), respectively]. Concerning R2, Design I has the lowest percentage of wrong answers 34(2.3%), which is nearly half that in Design II and Design III [64(4.3%) and 58(3.9%), respectively]. Moreover, in R3, Design III has the lowest percentage of wrong answers 41(2.7%), when compared to Design I and Design II [51(3.4%) and 72(4.8%), respectively]. The difference in accuracy a cross medium designs is significant in all repetitions. In hard designs the accuracy of hard designs is nearly the same, and there is no significant difference in the percentage of wrong answers in R1 and R2 in all designs, around 7% and 5% respectively. However, in R3, the accuracy is significantly varied among different designs as Design III has the highest percentage of wrong answers 117(7.8%), which is one and half times Design II's 81(5.4%) and more than twice in Design I, 53(3.5%). The impact of repetition on accuracy rates reveals interesting patterns in user performance across varying design complexities. In easy designs, where tasks are straightforward, no significant difference in accuracy is observed across repetitions, consistent with prior studies emphasizing performance stability in low-complexity tasks [42]. However, in medium designs, significant differences appear in accuracy rates across different repetitions, highlighting the role of task familiarity in influencing user performance. Design II exhibits lower accuracy rates across all repetitions, suggesting potential challenges in task completion. Comparing both results, it's evident that while design complexity plays a significant role in influencing accuracy rates, the impact of repetition is more pronounced in medium and hard complexity designs. This suggests that while design optimization is crucial, iterative refinement based on user feedback and task repetition is equally essential in enhancing user performance and satisfaction across different complexities.

3.4 Preference

Designs Complexity: The study found that Design II had the highest percentage of general preference and understandable design, with (41.8%) and (39.26%), respectively. This is nearly twice the percentage found in Design III, which had (21.48%) and (23.05%), respectively. Additionally, the choice of appeal in Design II and Design III was more than one and a half times that of Design I. Specifically, Design II and Design III had (39.06%) and (34.38%) appeal, respectively, while Design I had (24.41%). Furthermore, the resulting percentage of the most memorable design was Design III 250, which is almost twice as high as in Design II, with (48.83%) and (27.34%), respectively, as well as in Design I, with (20.31%). Lastly, more than half of the participants believed that Design III could trigger behavioural changes (52.34%), which is almost four times that of Design II (15.43%) and Design I (9.38%).

Regarding the factors affecting preferences, about half of the participants chose the colored bar with icons (44.53%), while nearly one-third chose the colored bar with text (39.26%). The lowest percentage of participants (14.84%), chose the icons bar with icons. Only seven participants had other opinions about the factors affecting their preference. Analysing preference across different design complexities uncovers significant user preferences and perceptions disparities. Design II ranked as the preferred choice regarding general preference, understandability, and appeal with substantially higher percentages than Designs I and III. At the same time, Design III is preferred for memorability and potential for triggering behavioral changes. These findings align with previous research highlighting the importance of user-centered design principles, where intuitive and visually appealing charts are favored by users [43]. The higher preference for Design II suggests its effectiveness in engaging users and promoting positive responses, resonating with studies emphasizing the significance of aesthetics and usability in visualization design [44].

Age: More than one and a half times group 45-64, which has a percentage of memorable for Design III 17(30.91%). Moreover, in understandable, Design II has the highest percentage in the lowest age group 54(46.55%). which is more than three times prefer than Design III, 18(15.52%) under the same age group but in age 25-34, Design I 97(39.11%) is preferred nearly more than one and half times Design III 59(23.79%). Moreover, Design I 24(43.64%) is also preferred also in the largest age group, nearly four times more than Design III 11(20%). There is a significant difference between factors of preference and age groups where all age groups preferred colored bars with icons; nearly half of each group except the age group 25-34 less than 40% chose this preference 97(39.11%), while this age group preferred more colored bars with text 108(43.55%) this percentage is nearly one and half times the percentage of same choice in the age group 35-44 (see Fig.6).

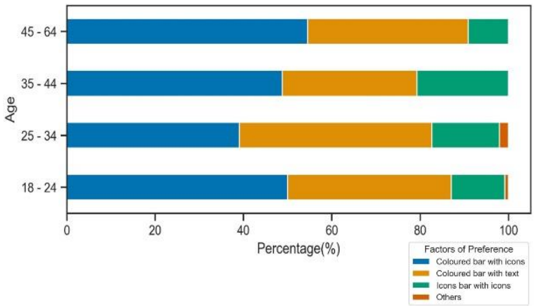


Figure 6. Percentage of factors of preference among different age groups.

Moreover, the percentage of those who have chosen icons bar with icons in the age group 35-44 is nearly twice 17(20.73%) that in the highest and lowest age group. The two intermediate age groups have a great percentage of choice icons bar with icons. The analysis of preferences across different age groups shows notable differences in design preferences and perceptions. Design II is favored among younger age groups, particularly regarding understanding ability. In comparison, Design I is preferred among older age groups, indicating potential age-related differences in cognitive processing and visualization preferences [45]. These findings align with theories of cognitive aging, where older adults may prefer simpler charts with clear navigation cues [46]. Additionally, the preference for colored bars with icons among all age groups underscores the importance of visual cues and intuitive design elements in enhancing user experience across diverse age demographics [47].

4. IMPLICATIONS

The study's findings underscore the importance of thoughtful design choices in visualizing correlational data. Tailored visualization designs can significantly impact task performance metrics such as accuracy and response time, suggesting potential improvements in data interpretation and analysis efficiency. Moreover, the influence of demographic factors like age on user preferences highlights the need for inclusive and user-centered visualization approaches. Specific design features, such as icons in colored bar charts, offer valuable insights for future research and practical applications in data visualization. Advancements in visualization techniques based on these findings can enhance decision-making processes in fields reliant on correlational data analysis. Ultimately, leveraging insights from this study can lead to improved data-driven decision-making and knowledge-sharing across various domains. By providing guidance on tailored visualization designs and considering demographic factors like age, our findings can enhance data interpretation and analysis.

5. CONCLUSION

This study examines the effectiveness of different colored bar chart designs in visualizing correlations between social media use, mental health, and family functioning. It emphasizes the importance of design choices in influencing response time, accuracy, user engagement, and preferences across various task complexities. The study highlights the influence of gender on user behavior and preferences, underscoring the need for personalized visualizations for diverse user demographics.

6. REFERENCES

[1] Alice, Plutino., L., Armellin., Andrea, Mazzoni., Roberta, Marcucci., Alessandro, Rizzi. (2023). Aging

variations in Ishihara test plates. Color Research and Application, doi: 10.1002/col.22877

[2] Alwuqaysi B, Abdul-Rahman A, Borgo R. The Impact of Social Media Use on Mental Health and Family Functioning Within Web-Based Communities in Saudi Arabia: Ethnographic Correlational Study. JMIR Form Res. 2024 Jan 16;8:e44923. doi: 10.2196/44923. PMID: 38227352; PMCID: PMC10828947.

[3] Andrew, S., McAvan., Andrew, S., McAvan., Yu, Karen, Du., Yu, Karen, Du., Alexis, Oyao., Alexis, Oyao., Stephanie, Doner., Stephanie, Doner.,Matthew, D., Grilli., Arne, D., Ekstrom., Arne, D., Ekstrom. (2021). Older Adults Show Reduced Spatial Precision but Preserved Strategy-Use During Spatial Navigation Involving Body-Based Cues.. Frontiers in Aging Neuroscience, doi: 10.3389/FNAGI.2021.640188

[4] Aniko, Simon., Mary, Guindy., Peter, A., Kara.,Tibor, Balogh., Laszlo, Szy. (2022). Through a different lens: the perceived quality of light field visualization assessed by test participants with imperfect visual acuity and color blindness. doi: 10.1117/12.2618638

[5] Ben, Steichen., Bo, Fu., Tho, Nguyen. (2020). Inferring Cognitive Style from Eye Gaze Behavior During Information Visualization Usage. doi: 10.1145/3340631.3394881

[6] Beverly Farrell (2022). The Development of an Instrument to Assess Strategical Information Processing Style. doi: 10.31390/gradschool_disstheses.338

[7] Cheng, Yang., Yiteng, Peng., Jing, Zeng. (2021). Research on cognition and application of icon complexity based on EEG. doi: 10.1007/S42486-021-00058-2

[8] Dragan, Rangelov., Hermann, J., Müller., Hermann, J., Müller., Michael, Zehetleitner. (2013). Visual search for feature singletons: multiple mechanisms produce sequence effects in visual search. Journal of Vision, doi: 10.1167/13.3.22

[9] Edy, Setiyo, Utomo. (2022). Analysis of the Mathematical Visualization Process of Female Students in Solving Contextual Problems Based on Cognitive Style. Phenomenon: Jurnal Pendidikan MIPA, doi: 10.21580/phen.2022.12.1.10850

[10] Eleanor, Eytam., Oded, Lowengart., Noam, Tractinsky. (2021). Effects of visual simplicity in product design and individual differences in preference of interactive products. Review of Managerial Science, doi: 10.1007/S11846-020-00391-3

[11] Epstein, N., Bishop, D., Levin, 1978. The McMaster model of family functioning. Publication Title:Journal of Marriage and Family Counseling Volume: 4

[12] G. A. Miller. The magical number seven plus or minus two: some limits on our capacity for processing information. Psychological Review, 63 2:81–97, 1956

[13] Ingrid, Maria. (2023). Universal Design and Inclusive Participation. doi: 10.4324/9781003266068-5

[14] Jianfeng, Wu., Dongfang, Jiao., Chunfu, Lu., Chengmin, Li., Xiaofang, Huang., Suzan, Weng. (2022). How Do Older Adults Process Icons in Visual Search Tasks? The Combined Effects of Icon Type and Cognitive Aging. International Journal of Environmental Research and Public Health, doi: 10.3390/ijerph19084525

[15] John, King., Kathy, Sonderer., Kevin, Lynch. (2016). Cognitive Benefits of a Simple Visual Metrics Architecture. doi: 10.1007/978-3-319-39396-4_29

[16] Julie, Baca., Daniel, W., Carruth., Alex, Calhoun., Michael, Stephens., Christopher, Lewis. (2021). Challenges in Evaluating Efficacy of Scientific Visualization for Usability and Aesthetics. doi: 10.1007/978-3-030-23535-2_29

[17] Kaifeng, Liu., Pengbo, Su., Hailiang, Wang., Da, Tao. (2022). Contextualizing Visualizations of Digital Health Information among Young and Older Adults Based on Eye-Tracking. Sustainability, doi: 10.3390/su142416506

[18] Karin, Eberhard. (2021). The effects of visualization on judgment and decision-making: a systematic literature review. doi: 10.1007/S11301-021-00235-8

[19] Leilani, Battle., Alvitta, Ottley. (2022). Testing theories of task in visual analytics. Interactions, doi: 10.1145/3529149

[20] Lichen, Zhang. (2023). Bar Charts, Butterfly Charts, Waterfall Charts, Dot Plots, Needle Plots, Area Bar Charts, Text Graphs, and Line Charts: Charts for Categorical Data. doi: 10.1007/978-1-4842-8609-8_4

[21] M. Elliott, C. Nothelfer, C. Xiong, and D. A. Szafir. A design space of vision science methods for visualization research. IEEE Trans. Vis. & Comp. Grap., 27:1117–1127, 2021

[22] Main, Uddin, Rony., Fan, Du., R., Rossi., Jane, Hoffswell., Niyati, Chhaya., Iftikhar, Ahamath, Burhanuddin., E., Koh. (2023). Augmenting Visualizations with Predictive and Investigative Insights to Facilitate Decision Making. doi: 10.1145/3543873.3587317

[23] Makoto Takeuchi; Kentaro Yumita; Rentaro Yoshioka (2023). UI Design Method for Visualizing Sensor Data to Enhance User Understanding. doi: 10.1109/iconat57137.2023.10080807

[24] Mariya, E., Manahova., Eelke, Spaak., Floris, P., de, Lange. (2020). Familiarity Increases Processing Speed in the Visual System. Journal of Cognitive Neuroscience, doi: 10.1162/JOCN_A_01507

[25] Mark, Andrew. (2022). Cognitive challenges in complex system design. doi: 10.21606/drs.2022.152

[26] Mark, Harrower., Cynthia, A., Brewer. (2003). ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. Cartographic Journal, doi: 10.1179/000870403235002042

[27] Mary, P., Czerwinski., Nancy, Lightfoot., Richard, M., Shiffrin. (1992). Automatization and training in visual search.. American Journal of Psychology, doi: 10.2307/1423030

[28] Mengting, Zhang., Almut, Hupbach. (2020). Repeated encoding fosters retention of perceptual detail in visual recognition memory. Learning & Memory, doi: 10.1101/LM.052209.120

[29] Nor Sheereen, Z., Rozumah, B., 2010. Using the 12-item general health questionnaire (GHQ-12) to assess the psychological health of malaysian college students. Global Journal of Health Science 2, 73–80. URL:http://www.ccsenet.org/journal/index.php/gjhs/article/view/3852, doi:10.5539/gjhs.v2n1p73.

[30] Nozipho, Matumbu. (2023). A User-Centred Approach to User Interface Languages and Icons: Co-evaluation and Co-creation of Accessible Digital Mobility Services. doi: 10.1007/978-3-031-26155-8_12

[31] Phil, Reed., Tegan, Fowkes., M., Khela. (2023). Reduction in Social Media Usage Produces Improvements in Physical Health and Wellbeing: An RCT. Journal of technology in behavioral science, doi: 10.1007/s41347-023-00304-7

[32] Pil, Brix, Purup., Pil, Brix, Purup., Steffen, E., Petersen. (2021). Characteristic traits of visualizations for decision-making in the early stages of building design. Journal of Building Performance Simulation, doi: 10.1080/19401493.2021.1961864

[33] Reshma Pise, Bharati Ainapure (2022). User Interface Design With Data Visualization Technique. 227-251. doi: 10.4018/978-1-7998-9121-5.ch011

[34] S. Palan and C. Schitter. Prolific.ac—a subject pool for online experiments. Journal of Behavioral and Experimental Finance, 17, 12 2017

[35] Saeed, Asgharzadeh. (2022). Visualization of Information and Its Application in Information Management. doi: 10.52547/ijamac.1.2.20

[36] Sara, Tandon., Alfie, Abdul-Rahman., Rita, Borgo. (2022). Measuring Effects of Spatial Visualization and Domain on Visualization Task Performance: A Comparative Study. IEEE Transactions on Visualization and Computer Graphics, doi: 10.1109/TVCG.2022.3209491

[37] Saratchandra, Chatterjee. (2022). Culture and perception. doi: 10.4324/9781003051497-8

[38] Shady, Afrin, Jeesan., Han-Seok, Seo. (2020). Color-Induced Aroma Illusion: Color Cues Can Modulate Consumer Perception, Acceptance, and Emotional Responses toward Cooked Rice.. Foods, doi: 10.3390/FOODS9121845

[39] So-Yeon, Yoon., Yun, Jung, Choi., Hyunjoo, Oh. (2015). User attributes in processing 3D VR-enabled showroom: Gender, visual cognitive styles, and the sense of presence. International Journal of Human-computer Studies \/ International Journal of Man-machine Studies, doi: 10.1016/J.IJHCS.2015.04.002

[40] Sukanya, Khanom., Anneli, Heimbürger., Tommi, Kärkkäinen. (2015). Can icons enhance requirements engineering work. Journal of Visual Languages and Computing, doi: 10.1016/J.JVLC.2014.12.011

[41] The data for this paper was generated using Qualtrics software, Version 2023 of Qualtrics. Copyright © 2023 Qualtrics. Qualtrics and all other Qualtrics product or service names are registered trademarks or trademarks of Qualtrics, Provo, UT, USA. https://www.qualtrics.com

[42] Tien-Lung, Sun. (2003). An icon-based data image construction method for production data visualization. Production Planning & Control, doi: 10.1080/0953728031000118787

[43] Ting Dai, Joshua M. Rosenberg, Michael Lawson (2022). Data Representations and Visualizations in Educational Research. doi: 10.4324/9781138609877-ree148-1

[44] Umair, Afzal., Arnaud, Prouzeau., Lee, Lawrence., Timothy, D., Dwyer., Saikiranrao, Bichinepally., Ariel, Liebman., Sarah, Goodwin. (2022). Investigating Cognitive Load in Energy Network Control Rooms: Recommendations for Future Designs. Frontiers in Psychology, doi: 10.3389/fpsyg.2022.812677

[45] V., Viqueira, Pérez., D., De, Fez, Saiz., F., M., Martínez, Verdú. (2010). Colour vision: theories and principles. doi: 10.1533/9780857090195.1.3

[46] Valentina, Bianco., Marika, Berchicci., Federico, Quinzi., Rinaldo, Livio, Perri., Donatella, Spinelli., F., Di, Russo. (2020). Females are more proactive, males are more reactive: neural basis of the gender-related speed/accuracy trade-off in visuo-motor tasks.. Brain Structure & Function, doi: 10.1007/S00429-019-01998-3

[47] Xuedi, Qin., Yuyu, Luo., Nan, Tang., Guoliang, Li. (2020). Making data visualization more efficient and effective: a survey. doi: 10.1007/S00778-019-005883

A Novel DWT-based Method for Image Sharpness Measure using Edge-related Frequency Components

Yuan-Kang Lee
Communication Engineering
National Taiwan University
Taipei, Taiwan
r12942062@ntu.edu.tw

Jian-Jiun Ding
Communication Engineering
National Taiwan University
Taipei, Taiwan
jjding@ntu.edu.tw

Abstract— **In this paper, a novel no-reference image quality metric of sharpness is proposed. Our image quality metric is evaluated on two key attributes discerned during the assessment of image sharpness by the human visual system (HVS): 1. Image sharpness is principally contingent upon the salience of edges within the image. 2. With an increase in the decomposition level of the Discrete Wavelet Transform (DWT), the high-frequency coefficients correspond to higher spatial frequency information in an image. Experimental results show that in comparison to other state-of-the-art metrics, our method not only accurately assesses image sharpness in both defocus and motion blur scenarios but also showcases superior precision and broader applicability.**

Keywords—Image sharpness, image blur, no-reference, discrete wavelet transform, edge detection, image quality assessment

I. INTRODUCTION

As the development of digital cameras, augmented reality (AR), virtual reality (VR), autonomous vehicles, and medical imaging systems advance rapidly, the precise evaluation of the image sharpness is essential in the fields of image processing and computer vision. A high-quality image characterized by high sharpness contains fine details and textures, enriching the overall viewing experience while also augmenting precision in content analysis. As subjective methods for assessing image sharpness are time-consuming and inconsistency, objective assessments have taken precedence as the primary focus of research. The evaluation of image sharpness aims to analyze the clarity of the acutance and the texture in an image, serving as an important metric in tasks such as image deblurring [1][2], auto-focus algorithms [3][4][5], and remote sensing image analysis [6]. Broadly, objective methods for evaluating image sharpness can be categorized into three types: full-reference image quality assessment (FR-IQA), reduced-reference image quality assessment (RR-IQA), and no-reference image quality assessment (NR-IQA). FR-IQA involves comparing an image to a high-quality reference image, while RR-IQA relies on partial information from a reference image. On the other hand, NR-IQA evaluates image quality without any reference image. In practical applications, obtaining an undistorted reference image is frequently unfeasible. Hence, NR-IQA has emerged as the predominant development focus in contemporary image quality assessment endeavors.

In current research on no-reference evaluation of image sharpness, spatial-domain methods and transform-domain are widely employed. Spatial-domain methods are based on the concept that when an image is affected by blur, its contrast decreases, and its edges weaken. Utilizing the spatial property, many methods use the image's gradients and edge strength to assess image sharpness [7][8]. Due to the maturity of current image edge detection methods [9][10][11], employing spatial-domain techniques for image sharpness assessment is intuitive and straightforward. Transform-domain methods primarily use transform techniques such as the Discrete Fourier Transform [12], the Discrete Cosine Transform [13], and the Discrete Wavelet Transform [14][15] to extract the image's frequency components. Generally, the high-frequency components within an image its level of detail. When the image suffers from blurring, the high-frequency elements significantly diminish. Consequently, a blurred image contains less high-frequency information than a sharp image. Yet, both spatial-domain and transform-domain approaches are susceptible to the influence of noise in accessing image sharpness. Moreover, we observed that the existing no-reference image sharpness metrics can evaluate defocus blur but struggle to accurately assess cased of motion blur. The limitation results in a current lack of a more widely applicable metric for measuring image sharpness.

In our work, a novel no-reference metric for measuring image sharpness is proposed. Our method is based on two primary properties of the HVS regarding image sharpness:

1. The high-frequency elements associated with edges in images are the most significant factors used by human eyes to assess the image sharpness.
2. The DWT at different decomposition levels captures various scales of image's high-frequency coefficients in the transform domain. As the level of decomposition increases, the obtained high-frequency coefficients correspond to the information of higher spatial frequency in an image.

The proposed image sharpness metric leverages the edge-related DWT coefficients and combine three different scales of high-frequency components, culminating in a final sharpness map. As blur predominantly affects high-frequency elements, the low-frequency components of an image remain relatively unchanged. Hence, our metric relies on the content of edge-related and weighted high-frequency coefficients in the final sharpness map relative to its low-frequency components. The rest of this paper is structured as follows. Section II present the analysis of our metric. Section III shows the performance of our method compared to other state-of-the-art metrics in different experiments. Section IV offers conclusions of our work.

II. PROPOSED IMAGE SHARPNESS MEASURE

Defocus blur and notion blur are two types of blurring that can significantly degrade image quality. Defocus blur occurs when the camera lens fails to focus properly on the target, resulting in a global blur across the entire image. On the other hand, motion blur arises due to the relative movement between the camera and the object during image capturing, resulting in a directional blurred smudges in the image. The characteristic of defocus blur causing a uniform blur across the image makes it relatively easier for blur evaluation. Compared to defocus blur, the image influenced by motion blur exhibits a distinct property: a noticeable decrease in the high-frequency energy along the direction of motion in the blurred image [16][17][18]. Due to the different effect of motion blur and defocus blur on images, solely employing spatial or transform domain methods is ineffective for accurately evaluating sharpness.

A. Combination of the Spatial and the Transform Information

The two-dimensional DWT decomposes an image into 4 divided layers, and a DWT decomposition result with multiple decomposition levels is shown in Fig 1.

1. LL layer: Represents low-frequency components of the image, which remain relatively unchanged during blurring.
2. LH layer and HL layer: Capture image's high-frequency components along horizontal and vertical directions, both effected noticeably by blurring.
3. HH layer: Contains high-frequency elements along the diagonal direction, also dramatically effected by blurring.

To align our proposed metric more closely with the human visual assessment of image sharpness, we employ Sobel edge detection method on the LL layer to extract high-frequency coefficients related to edges. Through this approach, we not only enhance the significant influence of edges on sharpness but also reduce the impact of noise on sharpness evaluation. Since the Haar transform holds higher practical value because of its computational efficiency and implementation simplicity compared to other wavelet transforms, we then apply the Haar transform to capture edge-related high-frequency coefficients. In our method, the edge map *EM* is expressed as in (1).

$$EM_i = \sqrt{LH_i^2 + HL_i^2 + HH_i^2} \cdot EdgeMask \qquad (1)$$

The sub-bands *LH*, *HL*, and *HH* are generated through the Haar transform's decomposition process, and the lowercase notation *i* denotes the decomposition level. *EdgeMask* denotes the outcome of edge detection achieved by applying the Sobel operator to the LL layer of the image. The improved edge map include all high-frequency elements corresponded to edges. It is worth noting that to encompass all edge-related coefficients, an edge dilation operation becomes necessary. We employ a dilation process on *EdgeMask*, which expands the size of the edges fourfold, facilitating the inclusion of various scales of high-frequency coefficients related to edges.

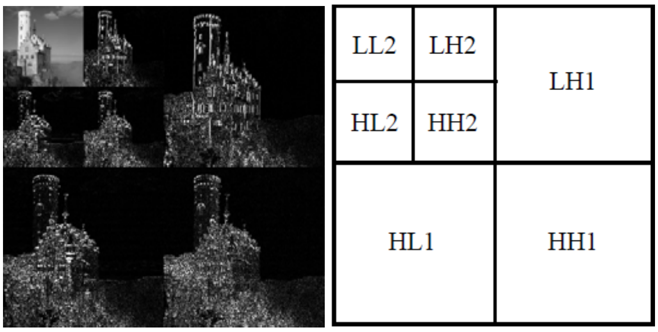


Fig. 1 A DWT decomposition result with multiple decomposition levels

B. Spatial Frequency and DWT Coefficients

Our human visual system assesses image sharpness based on the image's spatial frequency. An image with higher spatial frequency exhibits sharper edges, characterized by shorter rise distance. The rise distance of an edge can be expressed as the difference of pixel levels between 10% to 90% of its final value. Fig. 2(a) shows two bar pattern with different spatial frequencies. The upper half displays a high-quality bar pattern with high spatial frequency, while the lower half shows a quality-degraded bar pattern with low spatial frequency. Fig. 2(b) illustrates two edge with high and low spatial frequencies.

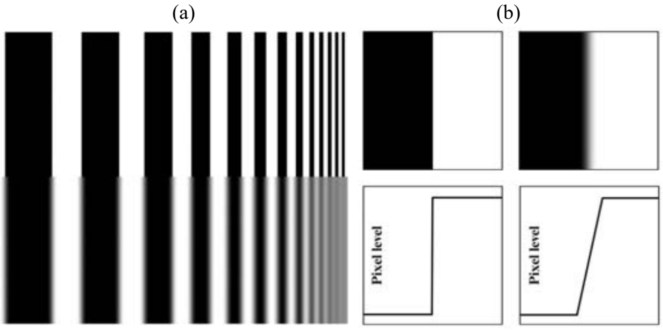


Fig. 2(a) The comparison of bar pattern images exhibiting different spatial frequencies. Fig. 2(b) The top left corner: the edge image with high spatial frequency. The bottom left corner: the pixel level variation curve of the edge with high spatial frequency. The top right corner: the edge image with low spatial frequency. The bottom right corner: the pixel level variation curve of the edge with low spatial frequency.

There are numerous well-established and widely applied methods in the industry for computing spatial frequencies in images [19][20][21]. Yet, these methods require specific test charts such as the slanted-edge chart [19], the Siemens star chart [20], and the dead leaves chart [21] to be captured by a camera for the spatial frequency measurement. This limitation restricts their applicability, rendering these methods unsuitable for evaluating sharpness across various scenes. To enhance the applicability of sharpness evaluation, the pivotal focus lies in extracting higher spatial frequency information from image's content. Our study revealed that the high-frequency elements extracted by higher decomposition levels contain information of higher spatial frequency. To verify this image characteristic, we utilize the Power Spectrum Density (PSD) to analyze the spatial frequency information of an image. PSD represents the distribution of energy across different spatial frequencies in the image. It is obtained through the Fourier transform of the image, providing insight into the frequency content and spatial characteristics. For an image f(m,n) with dimensions of M x N pixels, its Fourier Transform F(u,v) is represented as shown in (2). PSD can then be expressed as in (3).

$$F(u,v) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-j2\pi(um/M+vn/N)} \tag{2}$$

$$P(u,v) = 10 \cdot log(|F(u,v)|^2) \tag{3}$$

If an image contains higher spatial frequency information, its mean value of Power Spectral Density (PSD) tends to be higher. To analyze this property, we captured six Siemens star charts with varying levels of sharpness for experimentation. Siemens star charts encompass spatial frequency information ranging from low to high. The Modulation Transfer Function (MTF) for each of these six test charts was calculated using professional image quality software iQ-analyzer to evaluate the image spatial resolution. Fig. 3 displays the Siemens star chart images arranged from left to right in ascending order of sharpness. To quantify the sharpness of these six Siemens star chart images, we utilized MTF50 value, which represents the Modulation Transfer Function at 50% contrast. Table I shows the MTF50 values alongside the mean values of PSD for the edge maps EM1, EM2, and EM3 corresponding to the six Siemens star charts. Experimental data validates that the edge map EM with a higher wavelet decomposition level contains higher spatial frequency information. To enhance the precision of our proposed image sharpness measure, it is imperative to incorporate edge maps at higher decomposition levels. In other words, a comprehensive image sharpness metric necessitates the consideration of finer details in the evaluation.

TABLE I. MTF50 AND MEAN PSD VALUES OF EDGE MAP EM1, EM2, AND EM3 OF SIEMENS STAR CHART IMAGES SHOWN IN FIG .3 (A)-(F)

	Fig .3(A)	Fig .3(B)	Fig .3(C)	Fig .3(D)	Fig .3(E)	Fig .3(F)
MTF50 (LP/PH)	415	461	485	509	521	574
EM_1 mean PSD (dB)	50.4666	52.0948	53.4735	54.2395	55.4626	56.0850
EM_2 mean PSD (dB)	62.5617	64.1274	65.6600	66.5087	67.4256	68.7728
EM_3 mean PSD (dB)	72.5062	74.4403	75.5845	76.5072	77.3337	77.9862

The final edge map EM_f is expressed as in (4), where EM_1, EM_2, and EM_3 represent the edge maps corresponding to the wavelet decomposition level 1, 2, and 3, respectively. It is noteworthy that EM_1 is twice the size of EM_2, and EM_2 is twice the size of EM_3. Hence, for precise generation of the weighted edge map, both EM_1 and EM_2 should be resized to match the dimensions of EM_1.

$$EM_f = EM_1 + EM_2 + EM_3 \tag{4}$$

C. Image Sharpness Evaluation Model

Considering the limited impact of blurriness on the low-frequency components of an image, the assessment of image sharpness relies on the ratio of high-frequency content to low-frequency content. Based on this concept, our proposed image sharpness evaluation model is expressed as in (5), (6), and (7).

$$\overline{EM_f} = \left[\sum_{i=1}^{M/8}\sum_{j=1}^{N/8} EM_f(i,j)\right]/(MN/64) \tag{5}$$

$$\overline{LL_1} = \left[\sum_{i=1}^{M/2}\sum_{j=1}^{N/2} EM_f(i,j)\right]/(MN/4) \tag{6}$$

$$HFEM = \overline{EM_f}/\overline{LL_1} \tag{7}$$

HFEM, EM_f, and LL_1, are denoted as the proposed image sharpness measure using High-Frequency Edge Map, the final edge map, and the low-frequency sub-band decomposed by the level 1 Haar transform, respectively.

III. EXPERIMENTS

We conducted a comparative analysis of our proposed no-reference image quality metric for sharpness HFEM against existing state-of-the-art approaches: CPBD [22], PSI [23], and MGVG [8]. Four sharpness evaluation method were validated using Kodak24 database.

A. Gaussian Blur experiments

We experimented blurring simulations using Gaussian Blur on the images, varying the standard deviation of Gaussian blur from 0.5 to 10 in intervals of 0.5. Several blurred images with varying degrees are shown in Fig .4, using the Kodak8 image. With an increase in the standard deviation, a corresponding decrease in the image sharpness measure is anticipated.

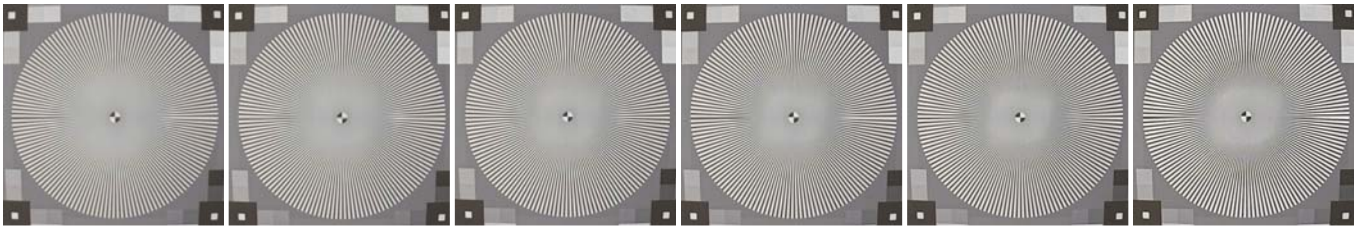
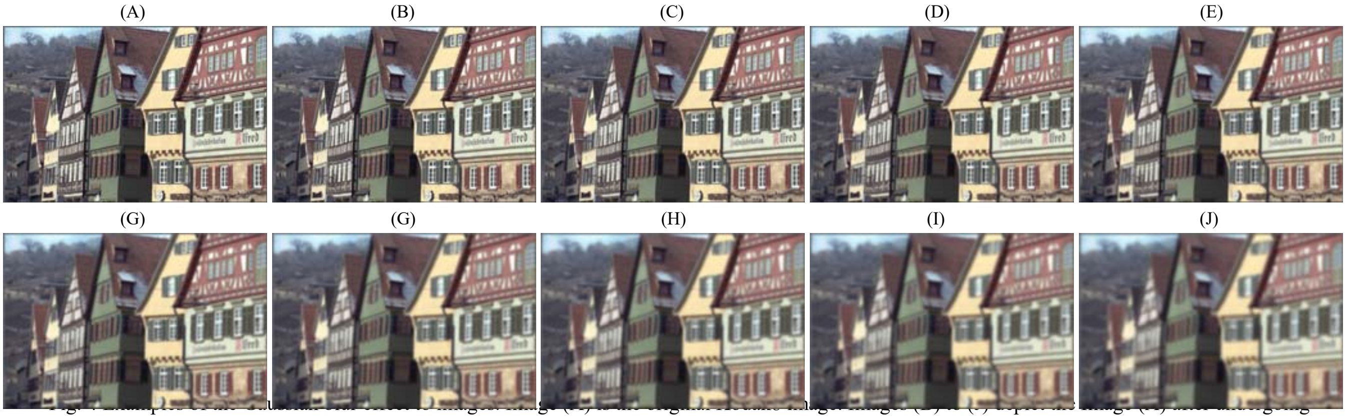
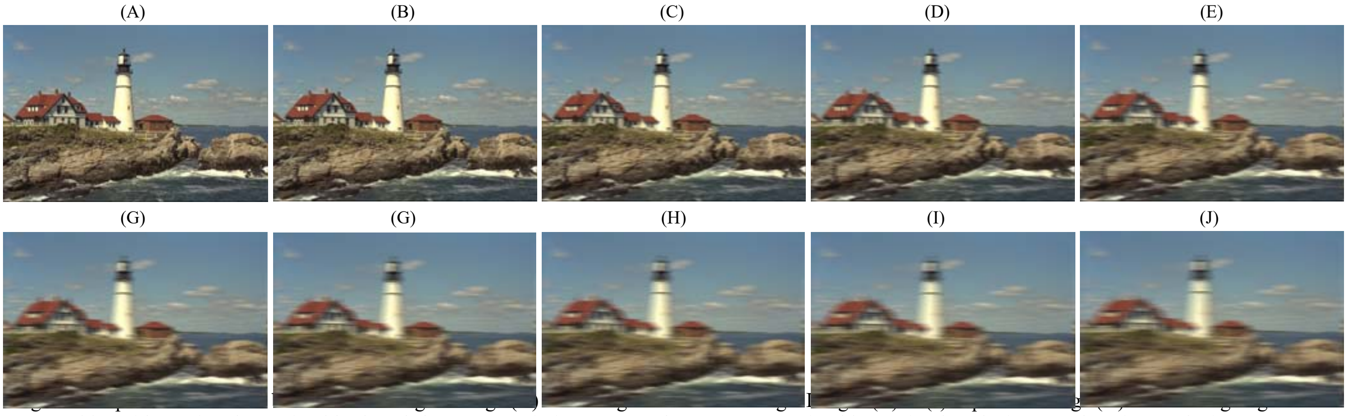


Fig. 3 Siemens star chart contains low and high spatial frequency information. Images (A) - (F) arranged from left to right in increasing order of sharpness.



Gaussian blur with standard deviations ranging from 0.5 to 4.5, in increments of 0.5. We use Gaussian blur to simulate the impact of defocus on the image.



blur with varying the motion blur length from 5 to 45, in increments of 0.5. We use motion blur to simulate the impact of motion on the image.

Fig. 6, Fig. 7, Fig. 8, and Fig. 9 illustrate the trend lines of CPBD, PSI, MGVG, and our metric HFEMS. Experimental results show that CPBD and PSI both exhibit inaccuracies in evaluating sharpness if the standard deviation of Gaussian blur exceeds 3. Our proposed metric HFEMS demonstrates great evaluation performance regardless of whether the standard deviation of Gaussian blur is low or high.

B. Motion Blur experiments

Motion blur can be modeled by two parameters: length and angle. The length specifies the quantity of pixels displaced due to camera movement, while the angle determines the direction along which the blur occurs. We experimented motion blur simulations on images, varying the motion blur length from 5 to 100 in intervals of 5. Several blurred images with varying motion-blurred lengths at 0 angle are shown in Fig .5, using the Kodak21 image. With an increase in the motion-blurred length, a corresponding decrease in the sharpness measure is anticipated. Fig. 10, Fig. 11, Fig. 12, and Fig. 13 illustrate the trend lines of CPBD, PSI, MGVG, and our metric HFEMS. Our metric exhibits exceptional and superior performance in evaluating image sharpness. Other existing metrics lack the capability to accurately measure the effect of motion blur.

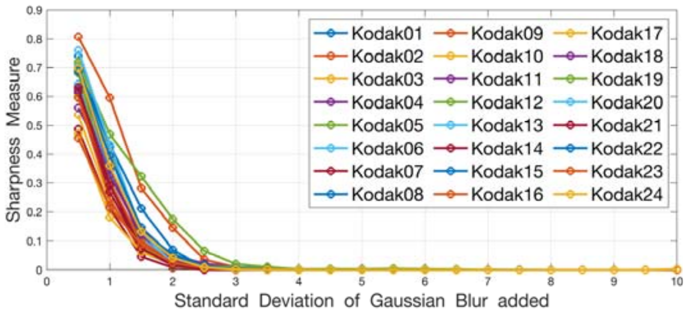


Fig. 6 The variation of CPBD [22] vs Gaussian Blur on Kodak 24 images

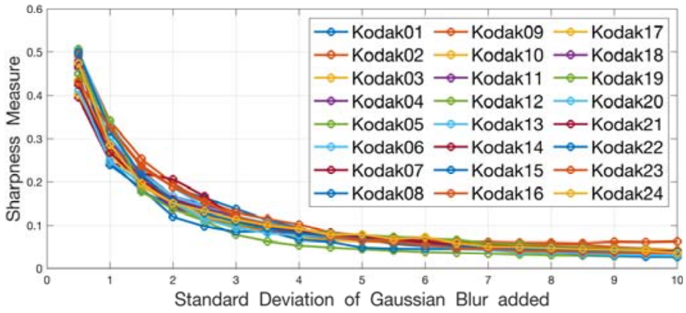


Fig. 7 The variation of PSI [23] vs Gaussian Blur on Kodak 24 images

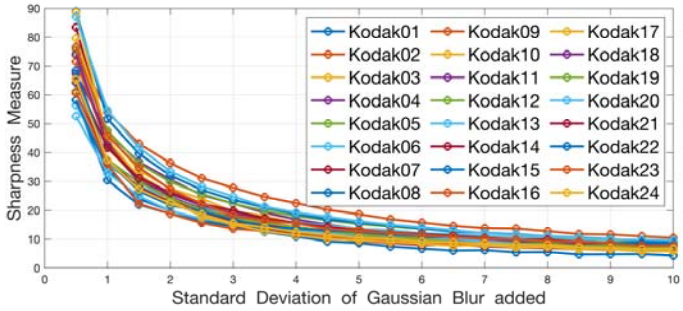


Fig. 8 The variation of MGVG [8] vs Gaussian Blur on Kodak 24 images

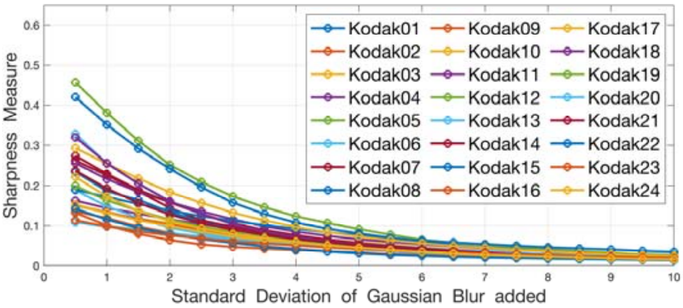


Fig. 9 The variation of HFEM vs Gaussian Blur on Kodak 24 images

IV. CONCLUSION

In this paper, we propose a novel no-reference image quality metric for evaluating sharpness using High-Frequency Edge Map HFEM. Our approach utilizes both spatial and transform domain information in an image to achieve precise image sharpness measurements. By integrating the concept of the relationship between the decomposition levels of the wavelet transform and the spatial frequency and our method aligns with the Human Visual System. Additionally, our proposed metric accurately assesses the impact of motion blur on the image, surpassing other state-of-the-art image sharpness metrics. In the future, our aim is to utilize this proposed image sharpness metric for the development of a new blind image deblurring method and a camera auto-focus algorithm.

REFERENCES

[1] Leclaire, Arthur, and Lionel Moisan. "No-reference image quality assessment and blind deblurring with sharpness metrics exploiting fourier phase information." Journal of Mathematical Imaging and Vision 52 (2015): 145-172.

[2] Xiang, Xinguang, Hao Wei, and Jinshan Pan. "Deep video deblurring using sharpness features from exemplars." IEEE Transactions on Image Processing 29 (2020): 8976-8987.

[3] Yao, Yi, et al. "Evaluation of sharpness measures and search algorithms for the auto-focusing of high-magnification images." Visual Information Processing XV. Vol. 6246. SPIE, 2006.

[4] Yousefi, Siamak, M. Rahman, and Nasser Kehtarnavaz. "A new auto-focus sharpness function for digital and smart-phone cameras." IEEE Transactions on Consumer Electronics 57.3 (2011): 1003-1009.

[5] Zhang, Zheng, et al. "Focus and blurriness measure using reorganized DCT coefficients for an autofocus application." IEEE Transactions on Circuits and Systems for Video Technology 28.1 (2016): 15-30.

[6] Chen, Guobin, and Maotong Zhai. "Quality assessment on remote sensing image based on neural networks." Journal of Visual Communication and Image Representation 63 (2019): 102580.

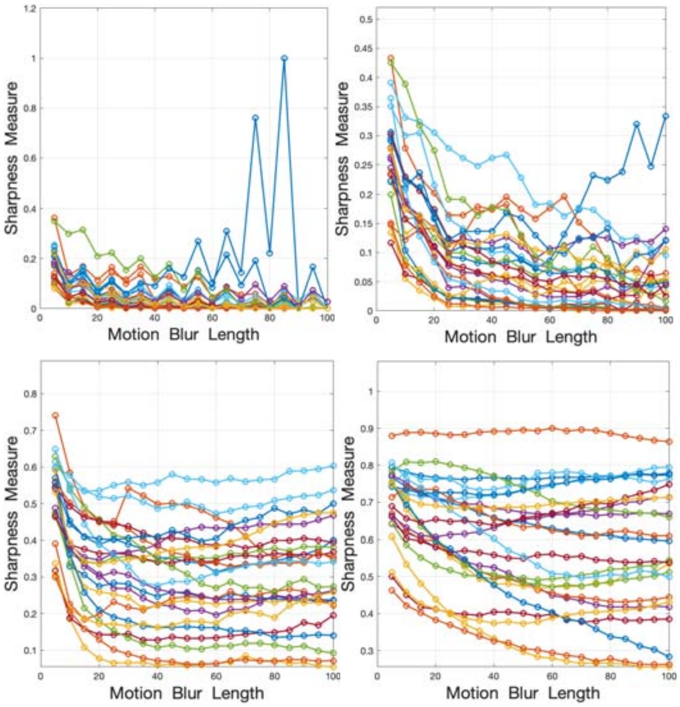


Fig. 10. Trend lines comparison: the variation of CPBD [22] vs motion blur on Kodak 24 images. The top left corner: trend with a motion blur angle 0°. The top right: trend with a motion blur angle 30°. The bottom left: trend with a motion blur angle 60°. The bottom right: trend with a motion blur angle 90°

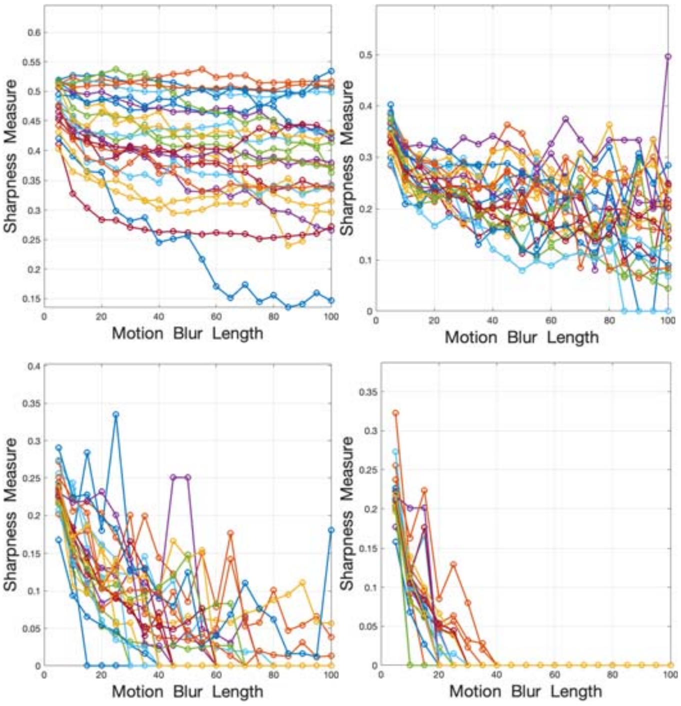


Fig. 11. Trend lines comparison: the variation of PSI [23] vs motion blur on Kodak 24 images. The top left corner: trend with a motion blur angle 0°. The top right: trend with a motion blur angle 30°. The bottom left: trend with a motion blur angle 60°. The bottom right: trend with a motion blur angle 90°

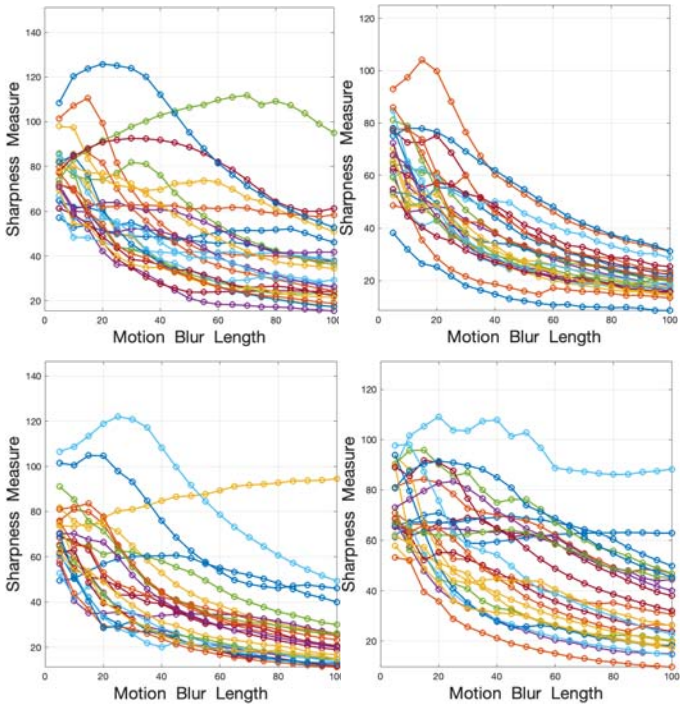


Fig. 12. Trend lines comparison: the variation of MGVG [8] vs motion blur on Kodak 24 images. The top left corner: trend with a motion blur angle 0°. The top right: trend with a motion blur angle 30°. The bottom left: trend with a motion blur angle 60°. The bottom right: trend with a motion blur angle 90°

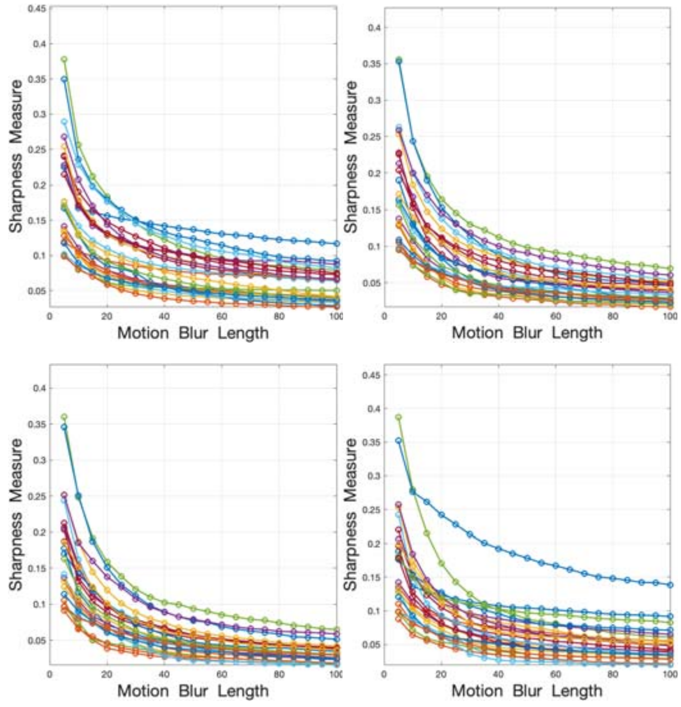


Fig. 13. Trend lines comparison: the variation of our proposed image quality metric of sharpness HFEM vs motion blur on Kodak 24 images. The top left corner: trend with a motion blur angle 0°. The top right: trend with a motion blur angle 30°. The bottom left: trend with a motion blur angle 60°. The bottom right: trend with a motion blur angle 90°

[7] Marziliano, Pina, et al. "Perceptual blur and ringing metrics: application to JPEG2000." Signal processing: Image communication 19.2 (2004): 163-172.

[8] Zhan, Yibing, and Rong Zhang. "No-reference image sharpness assessment based on maximum gradient and variability of gradients." IEEE Transactions on Multimedia 20.7 (2017): 1796-1808.

[9] Maini, Raman, and Himanshu Aggarwal. "Study and comparison of various image edge detection techniques." International journal of image processing (IJIP) 3.1 (2009): 1-11.

[10] Ahmed, Ahmed Shihab. "Comparative study among Sobel, Prewitt and Canny edge detection operators used in image processing." J. Theor. Appl. Inf. Technol 96.19 (2018): 6517-6525.

[11] Liu, Zhaoyang, et al. "An improved method for evaluating image sharpness based on edge information." Applied Sciences 12.13 (2022): 6712.

[12] De, Kanjar, and V. Masilamani. "Image sharpness measure for blurred images in frequency domain." Procedia Engineering 64 (2013): 149-158.

[13] Zhang, Zheng, et al. "Focus and blurriness measure using reorganized DCT coefficients for an autofocus application." IEEE Transactions on Circuits and Systems for Video Technology 28.1 (2016): 15-30.

[14] Tong, Hanghang, et al. "Blur detection for digital images using wavelet transform." 2004 IEEE international conference on multimedia and expo (ICME)(IEEE Cat. No. 04TH8763). Vol. 1. IEEE, 2004.

[15] Vu, Phong V., and Damon M. Chandler. "A fast wavelet-based algorithm for global and local image sharpness estimation." IEEE Signal Processing Letters 19.7 (2012): 423-426.

[16] Yitzhaky, Yitzhak, and Norman S. Kopeika. "Identification of blur parameters from motion blurred images." Graphical models and image processing 59.5 (1997): 310-320.

[17] Al Maki, Wikky Fawwaz, and Sueo Sugimoto. "Blind deconvolution algorithm for spatially-invariant motion blurred images based on inverse filtering and DST." International journal of circuits, systems and signal processing 1.1 (2007): 92-100.

[18] Chen, Xiaogang, et al. "Motion blur detection based on lowest directional high-frequency energy." 2010 IEEE International Conference on Image Processing. IEEE, 2010.

[19] Masaoka, Kenichiro, et al. "Modified slanted-edge method and multidirectional modulation transfer function estimation." Optics express 22.5 (2014): 6040-6046.

[20] Loebich, Christian, et al. "Digital camera resolution measurements using sinusoidal Siemens stars." Digital Photography III. Vol. 6502. SPIE, 2007.

[21] Cao, Frédéric, Frédéric Guichard, and Hervé Hornung. "Dead leaves model for measuring texture quality on a digital camera." Digital Photography VI. Vol. 7537. SPIE, 2010.

[22] Narvekar, Niranjan D., and Lina J. Karam. "A no-reference image blur metric based on the cumulative probability of blur detection (CPBD)." IEEE Transactions on Image Processing 20.9 (2011): 2678-2683.

[23] Feichtenhofer, Christoph, Hannes Fassold, and Peter Schallauer. "A perceptual image sharpness metric based on local edge gradient analysis." IEEE Signal Processing Letters 20.4 (2013): 379-382.

The Disease of the Canine Eye - From Image to Diagnosis Using AI

Matija Buric
[0000-0003-3528-7550]

Marina Ivasic-Kos
[0000-0002-1940-5089]

Sanda Martincic-Ipsic
[0000-0002-1900-5333]

Faculty of Informatics and Digital Technologies and Centre for Artificial Intelligence, University of Rijeka, Ul. Radmile Matejcic 2,

51000 Rijeka, Croatia

matija.buric@uniri.hr
marinai@uniri.hr
smarti@uniri.hr

ABSTRACT

This research examines the application of computer vision (CV) and large language models (LLM) in diagnosing eye diseases in dogs. The study utilizes a U-Net framework, incorporating convolutional neural networks (CNNs) such as ResNet, Inception, VGG, and EfficientNet, to enhance the segmentation of eye disease areas. Along the base U-Net model, four U-Net-based models were developed and evaluated on a dataset specifically generated for this purpose, classifying eye diseases into four categories. The performance of the enhanced U-Net architectures was found to be superior to that of the standard U-Net, with the U-Net modified with ResNet34 achieving the best segmentation accuracy, as measured by a Jaccard index of 66.6% on a custom test set. The segmented images were then diagnosed using various LLMs, including ChatGPT, Mistral, Gemini (Bard), Claude, and Llama-2, which were assessed using 15 different symptom sets. The study demonstrates that combining advanced image segmentation techniques with LLMs can improve diagnostic accuracy in veterinary medicine. The approach leverages the segmentation capabilities of U-Net for precise localization and the diagnostic ability of LLMs to interpret symptoms, facilitating enhanced diagnostic tools. This method could be applicable to other medical diagnostic areas requiring similar dual capabilities.

Keywords

Computer Vision (CV), Large Language Models (LLM), Image Segmentation, U-Net Architecture, Veterinary Ophthalmology, Disease Localization, Diagnostic Tools, Veterinary Diagnostic Imaging, Automated Medical Diagnosis.

1. INTRODUCTION

Major advances in computer vision and deep learning make these technologies more interesting for integration in veterinary medicine to facilitate early and accurate diagnosis to improve treatment outcomes [RoFB15].

The article discusses the development and performance of models trained for semantic segmentation of ocular symptoms in dogs with standard U-Net encoder-decoder network and enhanced with different CNN backbones. A specially generated dataset, featuring symptoms like corneal cloudiness and scleral redness, supports the training of our models [GĐLM20]. For the interpretation of detected symptoms and the support of the diagnostic processes the LLM chatbots ChatGPT [SPKS24], Mistral [JSMB23] and Llama-2 [TMSA23] are evaluated. Gemini (used to be called Bard) [HMHS23], and Claude [WKBB23].

2. RELATED WORK

The CNNs have significantly advanced the medical diagnosis, with U-Net architecture emerging as particularly effective for medical imaging analysis due to its robust feature extraction and segmentation capabilities [RoFB15]. Enhancements to this model include integration with advanced CNN backbones like ResNet [HZRS15] and Inception [SVIS15], improving symptom localization in various medical contexts [ChCh19, LXWJ19]. Competitive performance of U-Net-based models in medicine is demonstrated also in [AARJ22, PTRS21] and [JBZL22] that has shown that U-Net can outperform transformer-based architectures in medical image registration with minimal modifications. Research on canine ocular conditions remains limited, focusing primarily on diseases such as glaucoma [JoMK06, SHIS11]. The scarcity of datasets poses significant challenges, and while synthetic datasets have proven beneficial in other areas, they are less suitable for this field due to the variability in clinical presentations [DKKC21]. The efficacy of LLMs in medical settings

https://www.doi.org/10.24132/CSRN.3401.41

has been less explored, with existing methodologies for performance measurement often not meeting the complex requirements of healthcare interactions [ARKH23, DaZB18, WKBB23]. This research incorporates LLM technology with image segmentation to investigate their combined potential in enhancing diagnostic accuracy for canine eye diseases.

3. U-NET MODELS FOR CANINE OPHTHALMOLOGIC SYMPTOMS SEMANTIC SEGMENTATION

The diversity of eye diseases in dogs poses significant diagnostic challenges. For more effective classification, diseases were categorized into prevalent symptom groups to better focus the model on essential visual indicators. This categorization spans various conditions, enhancing the model's capability to generalize across different canine breeds and ocular pathologies.

Dataset Description and Augmentation

A custom dataset comprising 145 images sized 320x320 pixels was developed, sourced from specialized ophthalmology clinics, and reviewed by veterinary specialists. It represents diverse breeds and disease manifestations. Each image is paired with a uniquely colored mask highlighting four key symptoms, maintaining proportionate representation across different classes. Augmentation techniques like horizontal flips, rotations, and translations were employed to introduce variability and improve generalization; however, zooming was avoided to preserve mask integrity. The final dataset now consists of 200 images with distribution of symptoms appearances from 118 for excessive tearing to 154 for sclera redness because of multi-symptom occurrence on images.

U-Net Architectural Enhancements and Performance Evaluation

The U-Net framework was enhanced with various backbone networks to reinforce feature extraction. Configurations included ResNet34, known for its deep residual learning; Inception V3, which utilizes inception modules for efficient cross-channel correlation processing; VGG16, noted for its deep, straightforward convolutional structure [SiZi15], and EfficientNet B3, optimizing the balance among depth, width, and resolution [TaLe20]. Performance evaluations focused on the mean Jaccard index across different ocular conditions as detailed in Table 1. The U-Net with ResNet34 backbone exhibited the best performance in 3 out of 4 symptom segmentation accuracy tests. These modifications to U-Net architecture underscore its potential to enhance disease segmentation in veterinary ophthalmology.

Architecture	Ocular Opacity	Sclera Redness	Excessive Tearing	Colored Ocular Protrusion
U-Net	38.5	44	0.3	55.8
U-Net + ResNet34	73.9	**80.6**	**38**	**73.9**
U-Net + Inception V3	**78.3**	78.3	37.9	54.2
U-Net + VGG16	75.1	75.7	27.2	54.1
U-Net + EfficientNet	69.7	79.2	36.1	67.5

Table 1. Results of U-Net segmentation in Jaccard coefficient index showing performance using different backbones.

4. INTERPRETATION OF SYMPTOMS USING LARGE LANGUAGE MODELS

The integration of U-Net architecture in this experiment assures precise localization of ocular symptoms in dogs. This segmentation information, in the form of symptom presence, are fed into the diagnostic phase, where large language models (LLMs) like ChatGPT, Mistral, Gemini, Llama, and Claude analyze and interpret these symptoms, providing preliminary diagnoses. This dual approach combines U-Net's accurate localization with the analytical capabilities of LLMs to enhance the diagnostic accuracy and efficiency.

Evaluating Symptom Interpretation Using LLMs

To evaluate the LLMs performance in interpreting textual descriptions of disease symptoms, it was necessary to define an appropriate metric to measure the success of the model. Textual descriptions of symptoms are subjective, often containing ambiguous or incomplete information. Therefore, the evaluation metric focuses on effectiveness of LLMs to accurately interpret these descriptions and to extract relevant medical information and provide accurate, coherent responses that are consistent with medical knowledge.

To ensure the reliability of symptom interpretation, only segmentation results with an accuracy above 98% are considered. This strict threshold ensures that errors in segmentation do not compromise the diagnostic accuracy of the LLMs.

Comparative Evaluation of LLM Diagnoses based on U-Net Symptoms in Canine Eye Diseases

The study assessed the diagnostic accuracy of LLMs using 15 medical cases with varying symptom combinations, each derived from U-Net identified symptoms. The accuracy of the LLMs diagnoses was evaluated using a straightforward scoring system which was based on expert knowledge for each individual case: a score of 1 was given for a correct match, 0 for indeterminate responses, and -1 for incorrect or hallucinated diagnoses. An example of such evaluation on an image taken from the [ThMP00] show an American bulldog with protrusion of pink colored mass and epiphora symptoms. These two symptoms were taken as a reference in setting the initial diagnosis which was presented to the LLM models. The prompt directed to the LLM was divided into context, desired output and query as shown in the Table 2. The analysis of LLM performance in diagnosing canine eye diseases highlights significant insights into their diagnostic capabilities. ChatGPT4 demonstrates the strongest overall performance, particularly in accurately diagnosing single-symptom conditions such as Nuclear Sclerosis and Cherry Eye. However, it, along with other models, encounters difficulties with complex multi-symptom scenarios. ChatGPT3.5 shows variability, effectively diagnosing conditions with complete symptom presence but struggling in their absence. Gemini and Claude are effective in straightforward cases but face limitations in complex scenarios. Mistral AI shows steadiness in simple cases, cautiously avoiding overconfident errors in more complicated diagnoses. Llama-2 struggles across a range of scenarios and fails to recognize excessive tearing as a symptom.

5. CONCLUSION

This study indicates a promising future in veterinary ophthalmology through the integration of computer vision and LLM technologies.

The implementation of U-Net particularly enhanced with backbones like ResNet34 and EfficientNet B3, has effectively segmented ocular symptoms in canines. The U-Net with ResNet34 notably outperformed the standard model, especially effective in recognizing corneal cloudiness and scleral redness. Challenges remain in segmenting excessive tearing due to difficulties in annotation and boundary determination. The development of a specialized dataset was crucial for optimizing the U-Net models to detect ocular symptoms accurately.

The utilization of LLMs such as ChatGPT, Mistral, Gemini, Llama-2, and Claude for symptom interpretation and preliminary diagnoses revealed that while LLMs manage single-symptom cases effectively, their performance decreases with the complexity of multiple symptoms. This observation underscores the need for further enhancement of AI diagnostic tools but also suggest potential in supporting diagnosis.

6. ACKNOWLEDGMENTS

This research was supported by an Image-based AI-assisted diagnostics of canine ocular disease (AICODD) project funded by Animal Eye Consultants of Iowa. Iowa, USA.

LLM	response to a Prompt*	Score
Llama-2	*Excessive Tearing + Coloured Mass Protrusion in the Eye Corner Diagnosis: **Allergic Reaction***	**-1**
Mistral	*Excessive tearing or Watery eyes + Coloured Swelling in the Eye Edge: Although less commonly seen together, coexisting watery eyes and coloured swelling may warrant **referral to a specialized veterinarian** for proper evaluation and management of suspected immune-mediated disorders affecting multiple organs.*	**0**
GPT4	*Watery Eyes + Protrusion of a Red or Pink Mass - Diagnosis: **Cherry Eye** with secondary tear duct irritation.*	**1**
Claude	*Based on the symptoms you described, **I do not have enough information to make a definitive diagnosis**. The tearing and protruding mass indicate an abnormality affecting the eye and surrounding structures. But the specific diagnosis cannot be made based on these limited symptoms alone. Further veterinary examination of the eye would be recommended.*	**0**
Gemini	*The combination of excessive tearing and a red/pink mass protruding from the eye corner strongly suggests **cherry eye**, a prolapse of the third eyelid gland. This condition usually requires veterinary examination and potentially surgery for correction.*	**1**

** [Can you make a short pinpoint diagnosis based on the combination of these dog eye symptoms to assist a vet in preliminary decision-making for diagnosis:*
1. Excessive Tearing/Watery Eyes
2. Protrusion of a Red or Pink Mass in the Eye Corner]

Table 2. Examples of prompt presented to three LLM models, along with their responses and associated score values. The Correct diagnosis is Cherry Eye.

7. REFERENCES

[AARJ22] Azad, Reza, Aghdam, Ehsan Khodapanah, Rauland, Amelie, Jia, Yiwei, Avval, Atlas Haddadi, Bozorgpour, Afshin, Karimijafarbigloo, Sanaz, Cohen, Joseph Paul, et al.: Medical Image Segmentation Review: The success of U-Net, [object Object] (2022)

[ARKH23] Ahmed, Imtiaz, Roy, Ayon, Kajol, Mashrafi, Hasan, Uzma, Datta, Partha Protim, Reza, Md. Rokonuzzaman: *ChatGPT vs. Bard: A Comparative Study* (preprint) : Preprints, 2023

[ChCh19] Chakrabarty, Navoneel, Chatterjee, Subhrasankar: A Novel Approach to Glaucoma Screening using Computer Vision. In: *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. Tirunelveli, India : IEEE, 2019 — ISBN 978-1-72812-119-2, pp. 881–884

[DaZB18] Dandekar, Ashish, Zen, Remmy A. M., Bressan, Stéphane: A Comparative Study of Synthetic Dataset Generation Techniques. In: Hartmann, S., Ma, H., Hameurlain, A., Pernul, G., Wagner, R. R. (eds.): *Database and Expert Systems Applications*, *Lecture Notes in Computer Science*. Cham : Springer International Publishing, 2018 — ISBN 978-3-319-98812-2, pp. 387–395

[DKKC21] Deane, Jake, Kearney, Sinead, Kim, Kwang In, Cosker, Darren: DynaDog+T: A Parametric Animal Model for Synthetic Canine Image Generation, arXiv (2021). — arXiv:2107.07330 [cs]

[GĐLM20] Grozdanić, Siniša, Đukić, Slavica, Luzhetskiy, Sergei, Milčić-Matić, Natalija, Lazić, Tatjana: *Atlas bolesti oka pasa i mačaka*. Beograd : Oculus Vet, 2020

[HMHS23] Hirosawa, Takanobu, Mizuta, Kazuya, Harada, Yukinori, Shimizu, Taro: Comparative Evaluation of Diagnostic Accuracy Between Google Bard and Physicians. In: *The American Journal of Medicine* vol. 136 (2023), Nr. 11, pp. 1119-1123.e18

[JBZL22] Jia, Xi, Bartlett, Joseph, Zhang, Tianyang, Lu, Wenqi, Qiu, Zhaowen, Duan, Jinming: U-Net vs Transformer: Is U-Net Outdated in Medical Image Registration? In: Lian, C., Cao, X., Rekik, I., Xu, X., Cui, Z. (eds.): *Machine Learning in Medical Imaging*, *Lecture Notes in Computer Science*. vol. 13583. Cham : Springer Nature Switzerland, 2022 — ISBN 978-3-031-21013-6, pp. 151–160

[JoMK06] Johnsen, Devin A. J., Maggs, David J., Kass, Philip H.: Evaluation of risk factors for development of secondary glaucoma in dogs: 156 cases (1999–2004). In: *Journal of the American Veterinary Medical Association* vol. 229. Schaumburg, IL, USA, American Veterinary Medical Association (2006), Nr. 8, pp. 1270–1274

[JSMB23] Jiang, Albert Q., Sablayrolles, Alexandre, Mensch, Arthur, Bamford, Chris, Chaplot, Devendra Singh, Casas, Diego de las, Bressand, Florian, Lengyel, Gianna, et al.: Mistral 7B, arXiv (2023)

[LXWJ19] Li, Liu, Xu, Mai, Wang, Xiaofei, Jiang, Lai, Liu, Hanruo: Attention Based Glaucoma Detection: A Large-Scale Database and CNN Model. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA : IEEE, 2019 — ISBN 978-1-72813-293-8, pp. 10563–10572

[PTRS21] Petit, Olivier, Thome, Nicolas, Rambour, Clément, Soler, Luc: U-Net Transformer: Self and Cross Attention for Medical Image Segmentation, [object Object] (2021)

[RoFB15] Ronneberger, Olaf, Fischer, Philipp, Brox, Thomas: U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv (2015). — arXiv:1505.04597 [cs]

[SHIS11] Strom, Ann Refstrup, Hässig, Michael, Iburg, Tine M., Spiess, Bernhard M.: Epidemiology of canine glaucoma presented to University of Zurich from 1995 to 2009. Part 1: Congenital and primary glaucoma (4 and 123 cases). In: *Veterinary Ophthalmology* vol. 14 (2011), Nr. 2, pp. 121–126

[SiZi15] Simonyan, Karen, Zisserman, Andrew: Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv (2015). — arXiv:1409.1556 [cs]

[SPKS24] Savage, Cody H., Park, Hyoungsun, Kwak, Kijung, Smith, Andrew D., Rothenberg, Steven A., Parekh, Vishwa S., Doo, Florence X., Yi, Paul H.: General-Purpose Large Language Models Versus a Domain-Specific Natural Language Processing Tool for Label Extraction From Chest Radiograph Reports. In: *American Journal of Roentgenology* (2024), p. AJR.23.30573

[TaLe20] Tan, Mingxing, Le, Quoc V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, arXiv (2020). — arXiv:1905.11946 [cs, stat]

[ThMP00] Thamizharasan, A, Murugan, M S, Parthiban, S: Surgical Management of Cherry Eye in a Dog

[TMSA23] Touvron, Hugo, Martin, Louis, Stone, Kevin, Albert, Peter, Almahairi, Amjad, Babaei, Yasmine, Bashlykov, Nikolay, Batra, Soumya, et al.: Llama 2: Open Foundation and Fine-Tuned Chat Models, arXiv (2023). — arXiv:2307.09288 [cs]

[WKBB23] Wu, Sean, Koo, Michael, Blum, Lesley, Black, Andy, Kao, Liyo, Scalzo, Fabien, Kurtz, Ira: A Comparative Study of Open-Source Large Language Models, GPT-4 and Claude 2: Multiple-Choice Test Taking in Nephrology, arXiv (2023)

Study of Evolution in Virtual Worlds

Hassene Ben Amar
Polytech Marseille,
Doshisha University
163 Avenue de Luminy
France, 13009, Marseille
hassenebenamar03@gmail.com

Masashi Okubo
Doshisha University
Tataratanioku, Keikikan
Building
Japan, 610-0321,
Kyotanabe
mokubo@mail.doshisha.ac.jp

ABSTRACT

This paper investigates the evolution of agents within a virtual world, focusing on DNA transfer between generations, identification of significant genes and the explorations of parameters influencing survival and gene significance. To address these questions, we create an artificial life simulation within Unreal Engine 5, which mirrors real-life characteristics and behaviors of animals. The methodology involves running a genetic algorithm with binary tournament, uniform crossover and n-point mutation and analyzing the collected data to determine the most significant genes in different cases. We demonstrate that parameters such as sensory capabilities, resource availability and mutation thresholds greatly influenced species' survival and their success in the virtual environment. The difference of size, health and sight capabilities is crucial for the survival of deers and their interaction with the environment. In conclusion, this study offers an insight on evolution and evolution dynamics, denoting the influence of resource availability, competition between agents, mutation thresholds and sensory capabilities, with a substantial potential if time and resources were allocated to the research.

0.1 Keywords

Virtual Worlds, Predator-Prey Problem, Genetic Algorithms, Computer Generated Simulations

1 INTRODUCTION

In a world where the environment is constantly changing due to various factors such as climate change, migration to the cities, industrialization, etc, we would like to predict or at least have an idea on how the different species that live on the globe might evolve in the future. We would also like to be able to understand how species used to be in the past and how they did evolve from their past self to their current descendant. The method we will use to try to bring an answer to this problem is to simulate the evolution system by using the concept of Artificial Life itself in a graphic engine (in our case, Unreal Engine 5). In this paper, replicating life and simulate the evolutionary system would follow a modified version of the genetic algorithm based on multiple conditions - the agent attribute or A.A which contains multiple statistics that we will develop more in the later of the paper. Depending on the agent species we will evolve, we want to be able to have multiple levels that simulate different types of environments, for example - a mountain range, a cave, a forest, plains, etc. We bring a new approach to the traditional genetic algorithm by implementing different species in the same level and creating a randomness of environment with the different resources available to the agent during his artificial life. The agents that exist in the level are not at all static and live depending on goals we implement in their A.I behavior. Finally, the performance evaluation index is based on the overall performance of the simulation and the gap between reality and our virtual world.

2 CREATING A REAL WORLD SIMULATION

The purpose of this paper is to create a realistic life simulation in Unreal Engine 5 (UE5) and study the evolution process of agents inside of it. We have to consider different factors to achieve such a result. These factors include giving the agents a physical form to interact with the environment, with other agents (predator and prey alike), being able to differentiate the agents with the help of attributes, implementing sight and hearing capabilities and automatically creating and managing resources. In the following sections, we will examine each of these factors in detail to develop a comprehensive realist life simulation.

2.1 The agent

The natural world is home to a vast array of ecosystems, blooming with a wide range of flora and fauna. To simulate these ecosystems and the organisms that inhabit them in a realistic manner, we must strive to accurately replicate their biological processes. Nevertheless, this

https://www.doi.org/10.24132/CSRN.3401.42

task is challenging due to the limitations of computational power. Our objective is to create virtual models of life that closely approximate reality while maintaining optimal performance. Therefore, we have to first recreate the fauna.

2.1.1 External representation

To interact with its environment and other agents in the system, each agent is required to possess a physical form (a 3D mesh) in the virtual world. Compared to [Ric01a], we did not model the assets ourselves, which could limit the computational performance of the system during collisions' calculations. Meshes used were complex and irregular in shapes, leading to a difficulty in defining collisions using primitives shapes. A solution was found using "Bounding Spheres" or "Sphere Capsule Collision" to wrap the model in an invisible sphere hence a simpler method to calculate collisions.

2.1.2 Internal representation

An agent has a physical form but it lacks the complexity that stems from life. Each species has abilities, faculties that separate them from others. In a more local scope, each of the members of these species is unique and has a different value on how much those abilities express themselves. DNA contains the information required to determine an animal's abilities. Due to a current limitation of computational power, modeling an animal's DNA the way it appears in the real world would prove challenging. Therefore, we found that the creation of a simplified DNA structure along with attributes that depend on the structure fixes our issue.

The agent's DNA

In animals, the length of the DNA structure can vary.

Animal Class	Average Length (base pairs - bp)	Minimum Length (base pairs)	Maximum Length (base pairs)
Mammals	3.5 billion	242 million	6.3 billion
Avian	1.385 billion	1.15 billion	1.62 billion
Serpentes	2.8 billion	1.3 billion	3.8 billion
Fishes	1.4 billion	103 million	133 billion

Table 1: Average, minimum, and maximum length of the DNA structure in the mammal, avian, serpentes, and fish classes.

In terms of memory, 1000000 base pairs (bp) = 1 Mb and 1000000000 bp = 1 Gb. With the number of agents required for the simulation and the technical constraints, even when considering modern computational power, we need to simplify the DNA data structure. We will discuss the implementation of the simplified DNA in a later part.

The agent's attributes

An animal is defined by its DNA and the abilities that the structure codes. An example of one of those abilities is the speed of the Cheetah, which has the title of "Land's fastest animal" [cheetah01] They are able to run extremely fast (at 70 mph or 112,654 km/h) because of a combination of physical traits and genetic adaptations such as their slender, muscular body, their enlarged nostrils and lungs (more oxygen = more energy), their long tail (maintains balance and control) and adapted claws (better traction and acceleration). We formalize those abilities, those adaptations in the form of attributes that possess numerical values and that depend on the simplified version of the DNA mentioned in the previous part. We created different variables that encompass all that is necessary to define an agent's abilities, for example: AgentID (unique identifier), AgentDamage (amount of damage the agent can inflict on hostile agents), SightRadius (radius of the agent's field of view). It should be noted that these attributes are an oversimplification of the true nature, primarily due to computational limitations and the inherent complexities involved in managing such extensive data. Simplifying the DNA data structure was necessary to navigate these complexities effectively. To initialize all attributes of an agent and automate the task, we created an algorithm which is explained hereafter.

Algorithm 1 Agent Initialization and Attribute Calculation

```
Initialize DNA structure of the Agent
Set sensory attributes: SightRange, Radius, Offset, HearingThreshold
for each sense do
    Calculate sense bonus
end for
Calculate statistics for each agent species with species-specific formulas
Initialize float attributes: AgentSize, Speed, Damage, Health, Survivability Score
Calculate:
AgentSize
Speed
Damage
Health
Survivability Score
Initialize Int Stats:
Set AgentID by random
```

2.2 Birth and Death simulation

In our world, birth and death are two processes that can make the number in a population of animals grow, wither or stay the same. Birth is the way new children come to life whereas death is the way life purges populations to keep numbers stable. This cycle of birth/death is the most important part of life since it enables the ecosystems to preserve resources and maintain life as it is. The paper [Ric01a] defines the cycle in their simulation of Omosa with the creation of the agent as a baby (birth) that go through the multiple stages of life and death by old age or its health points dropping to 0 in case of a fight with a predator. Our simulation takes an entirely another approach to define the cycle of birth and death. Birth is defined by the live creation of an agent inside the level (one of the environments) or spawn for short. Death is defined by the live destruction of an agent inside the level or despawn.

2.2.1 Simulation of birth

To simulate said process in our virtual environment, we defined an invisible box called an AgentSpawner that possesses multiple characteristics - TotalToSpawn (the number of agents to spawn), MaxSpawnRange(the maximum distance at which the spawner can spawn agents), etc. If the location of the attempted spawn already contains an agent or another physical entity (actor in UE5) with collision enabled, the attempted agent spawn will fail and go to the next one.

Step 1: Check if the TotalToSpawn is higher than 0.
Step 2: Initialize i = 0
Step 3: Loop until i >= TotalToSpawn
  - Calculate a random position with the Perlin variables on X and Y
  - Spawn the actor

The first spawning process is completely different than for the rest of the simulation.

2.2.2 Simulation of death

In real life, death is essential to preserve the balance in ecosystems between the number of predators and prey, the availability and scarcity of resources. An animal can die of old age or from its wounds due to a fight with another animal, trying to defend its territory or while roaming freely in the environment. Our goal is not to simulate and solve the predator-prey but to see how DNA would travel between the generations. We don't create sub-populations of agents, which could lead to inner fighting over prey and resources and "unnecessary" deaths. Instead, all agents of the same specie are "friendly" with each other and don't have default interaction between them. The despawn of the agent would happen in two cases. The first case is that the AgentHealth attribute drops to 0. It is possible that there was a fight between predator(s) and prey(s) or if the AgentEnergy is under a threshold for a certain time. The second case is when the simulation passes to the next generation of agents. The figure 1 showcases a graph that sums up the cycle of birth/death for our simulation.

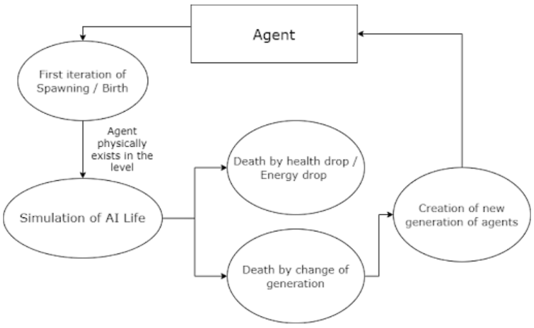

Agent
First iteration of Spawning / Birth
Agent physically exists in the level
Simulation of AI Life
Death by health drop / Energy drop
Death by change of generation
Creation of new generation of agents

Figure 1: Graph of cycle of birth/death in our simulation

2.3 The Environment

The purpose of our research is to be able to create a realistic life simulation. In the real world, animals live in a variety of ecosystems, such as forests, mountains, caves, seas, etc. Each of these ecosystems contains an array of resources that are necessary for the survival and development of each species. For example, cheetahs live in grasslands, savannas and some mountainous regions. Those environments are all open landscapes with a lot of resources (food, shelter and water) where the cheetah can use its speed to its maximum potential to chase down prey. Based on this example, we can infer that each animal has an ideal environment where it can use to its fullest its capabilities. Therefore, to mimic reality as closely as possible, we need to try to recreate to its maximum possible the vast array of environments where all of the agent species prosper.

2.3.1 Initial level

Mimicking reality is a process that takes a lot of time and resources, to imagine and create the environment itself. To be able to test our agents' fundamental features while not having our environments yet ready, we created a simple level field consisting of basic shapes provided by Epic Games in UE5. This level does not accurately reproduce the environments in which the agent species typically live.

2.3.2 Environment design

Multiple ecosystems nurture life as we know it. In reality, here is a random factor that will determine where the fauna and flora will end up living. To fully mimic an actual environment, we need to recreate that randomness and make our worlds more realistic. Furthermore, we study the evolution of a multitude of agents. Some agent species won't share the same ecosystem because their real-life counterparts could never survive there. Hence, a need to create a multitude of ecosystems in order to cover a certain percentage of the environments the animals live in.

Simulating the resources

To infuse the simulated environments with the unpredictability of life, we take inspiration from solutions implemented by level developers over the course of video level history. A common solution in level design to create this chance factor or randomness is to use procedural foliage. It would be a good idea if we didn't want to add randomness to each restart of the simulation. Unfortunately, there is no free procedural foliage library on the Epic Games marketplace, hence a need to develop a solution for ourselves. We believe that changing the environment every 2 generations could influence DNA changes between generations of agents. The solution we developed is based on the AgentSpawner. We reused the concept of a transparent box with a physical location in the environment. Although the AgentSpawner architecture makes it only able to spawn one kind of agent, the EnvironmentSpawner is able to spawn a vast array of special actors called EnvironmentActor. An EnvironmentActor is an spawnable or placeable actor (depending on the needs) in the level that has multiple variables required to simulate how its real counterpart would behave (while being static). All the meshes that we use for the EnvironmentActor come from the megascans collection in Quixel Bridge (free with UE5).

2.4 Predator and prey system

As we try to reproduce their behavior in virtual worlds. We recreate the diet plan of each agent species. Our solution was inspired from the paper [Geo01a]. Both types of agents (predator and prey) in the paper possess the capability to get information from their environment and act based on their perceptions, though in different manners. The predator agent can only see in a straight line with a high range of sight while the prey can see in all directions with a lower sight range to not give it an unfair advantage.

2.4.1 Agent's perceptions

The paper [Geo01a]'s agents use sensors to provide the perception information to the robots. Without a perception system to detect hostile agents or resources, an artificial life simulation would not be complete. The simulation in the paper [Ric01a] collected "the total population, numbers of births and deaths for both prey and predators as well as the number of predator kills and prey deaths from old age" with different settings (predator/prey awareness, flocking, herding). When the predator class isn't aware of the prey, it fails to function properly and when the prey class isn't aware of the predator, it quickly becomes extinct. Unreal Engine 5 provides some systems to implement senses in the level but after careful consideration, we decided to implement our own sense of sight. We use "Multi Object Sphere Trace" x times per second, which consists of sweeping a view range and detecting objects in the agent's field of view. This would compensate for the short-comings of the UE5 powered systems and enable herbivore and omnivore agents to detect sources of food as well as enemies. All agents share the same set of attributes, including SightRange, SightRadius, SightOffset and HearingThreshold for the senses configuration. Nevertheless, we define how much an agent's senses are expressed based on the real-life counterpart's known ability and a bonus coming from DNA structure.

The DNA's structure is defined with 8 values - SightRange, SightRadius, SightOffset, HearingThreshold, AgentSize, AgentSpeed, AgentDamage, AgentHealth.

The architecture of the DNA bonus is a bonus that is calculated (randomly between 0 and x) and added to the value of the attribute only if the gene is equal to 1. This could either advantage or disadvantage an agent depending on how its perceptions would affect its actions.

2.4.2 Agent's actions and decision making

Our perception system reacts to the stimuli around it (in a limited range). The agent is an A.I powered actor in the level that has predefined actions based on what the agent's perception system would register from the environment. If the stimuli detected comes from a hostile agent (H.A), the agent could react via two distinct ways. If the H.A is one of its predators, the agent would run away from it until reaching a safe location where the perception system doesn't detect the H.A anymore. In the case of the H.A being a prey, the agent would chase it until reaching a radius called the CombatRadius where it could start attacking it until one of them dies. In addition to the predator-prey pursuit and combat system, an agent would also be able to register what EnvironmentActors exist in its field of view. It would calculate automatically the closest resource and move in its direction in order to consume its content. The next figure shows a graph summarizing how an agent can register perceptions and the decision making it can make.

It is safe to say that we managed to reproduce the different parts of an ecosystem, fauna and flora as well as the animals' behavior. The reproduction of animals in the level was made by giving agents a physical and internal representation to help them interact with the world, while simultaneously remaking the cycle of birth and death with the AgentSpawner and destruction functions. We made the environment with heightmaps and artistic techniques and recreated the unpredictability of resources location with perlin noise [perlin01]. Finally, we created a system to reproduce an agent's sight, hearing being too complex to add on our own for now.

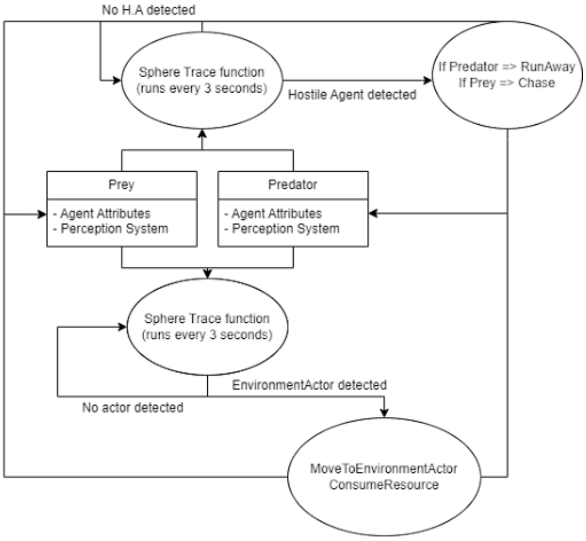


Figure 2: AI behavior, perceptions and decision making

Algorithm 2 Pseudocode of a Genetic Algorithm

```
Initialize the first generation
Evaluate the given generation based on a fitness function
if the first generation is the solution then
    return solution
end if
Selection of parent
Crossover of the parents' genetic representation and creation of children
Mutation of the offspring's genes
Check if we reached the termination condition
if not reached then
    Repeat from Step 2 with the offspring population
else
    Stop execution and print the found solution
end if
```

3 AGENTS' EVOLUTION

Evolution is the process over which heritable characteristics change in a certain population during the flow of time. The purpose of the paper is to study how our agents will evolve over the course of time and multiple generations, or tuning. Unfortunately, a problem such as this could not be solved using classical computing methods. Therefore, we follow the work of [Hol01a] with his well defined "Genetic Algorithm". In this part, we will view how this algorithm works, the different processes involved in it and how we remade them to fit our needs.

3.1 Overview of the genetic algorithm

3.1.1 Pseudo-code

A genetic algorithm or G.A is an algorithm that is used to tune certain traits in populations and arrive at a certain solution, best or accepted in a certain amount of time. It has the advantage of being able to look for solutions inside a wide solution space without needing a lot of information and being able to maintain a good performance.

3.1.2 Genetic representation

The reason why some problems require G.As is because the problem is way too complex to be solved using traditional methods. As stated in the introduction, these types of algorithms are commonly wielded to find solutions in a large search space and maintain a good execution time. However, G.As have a technical limitation. We have to represent all possible solutions of our problem with a linear representation. (for example, an array or a list) In consequence, we have to represent our DNA in a linear representation because it is the variable that we want to evolve. An agent's DNA is an array of size 8 that can only contain 0 and 1. We could fill it with numbers from R (real numbers) instead of just the {0, 1} set. Putting real numbers inside the DNA instead of 0 and 1 could possibly cause more harm than good because of two main reasons

- Truly random numbers are impossible to generate based on the deterministic nature of the algorithm used to create them in Unreal Engine 5[UE5Random]
- It would make the solution space go from 2^8 (256) to infinite.

The real-life counterpart of DNA can only take 4 possible values, which are adenine (A), cytosine (C), guanine (G), and thymine (T). These bases form specific pairs (A with T, and G with C), which makes the possibilities even lower. A possibility would be to change the set {0, 1} to {0, 1, 2, 3} but that would be something to discuss more in the next parts of the paper.

3.1.3 Fitness function

We implement a fitness function, which is simply a function that returns how close a solution is to the optimal solution of a problem. At the evaluation time in our simulation, we use the survivability time of alive agents as a fitness function. The survivability score is calculated based on the following formula:

$$\begin{aligned} SurvivabilityScore = (1.5 \cdot \log_2(\text{Speed}) + \\ 3 \cdot (\log_{10}(\text{Size}) + \log_{10}(\text{Health})) \\ + 3.5 \cdot \text{Damage}) + \text{SenseBonus}; \end{aligned}$$

The different constants {1.5f, 3.5f, 3} were chosen and assigned a weight to underline the importance of each value into the hypothetical chance of survival. We also added a SenseBonus to the equation of the SurvivabilityScore since it's theoretically accurate to hypothesize on the fact that better senses is equal to a survival chance for an agent. It is calculated using the following formula:

$$SenseBonus = \frac{\text{SightRange}}{\text{SightRadius}} \cdot \text{SightOffset} \cdot \left(\frac{1}{\text{HearingThreshold}} \right);$$

The ratio (SightRange / SightRadius) is used to perform the sphere trace function. A high ratio indicates a wide field of vision. However, a larger radius will increase the time needed to perform the trace, leading to a lower bonus. The higher the SightOffset is, the higher the bonus will be. This takes root in the fact that height is an advantage that can lead to agents being able to detect enemies as they approach as well as find resources easily. Finally, the (1.f / HearingThreshold) is such as when an agent's hearing ability is good, the higher the bonus will be.

3.1.4 *Termination criteria*

If a termination criteria is not implemented for a G.A, it will continue to run indefinitely because it wouldn't know what solution to look for in a given problem. In a problem such as ours without "conventional solutions", a termination criteria could probably be the number of generations, which could be changed between runs). The reason for such criteria is because of the intent behind our study. How genetic information would get transferred between generations and which combinations were the most interesting to keep by nature's law ? To avoid having the same solution over and over again, It could also be a good idea to implement a convergent protection. That means to stop running the algorithm if the N+1 generation's DNA set is very close to the N generation's DNA set.

3.2 Genetic operations

When new offspring are created in our world via reproduction, they undergo multiple stages and multiple processes until reaching birth. The real-life processes are vastly numbered and not the subject of this paper. However, the G.A is rendered useless if some of those processes are not reproduced in a virtual environment. They are called selection, crossover and mutation in a genetic algorithm.

3.2.1 *Binary Tournament*

After the evaluation is done, we have to select suitable parents for the crossover (C.R.O) step. In classical genetic algorithms, there are two methods to select parents. The "roulette wheel" method, which follows the "survival of the fittest" concept and consists of giving a probability of choosing a parent based on the ratio of its fitness to the sum of all fitnesses in a generation. This raises an issue of selecting only the best solutions in a set, which might lead to getting a suboptimal or not optimal solution for a given problem. The other method is called the binary tournament (B.T). It simply consists of taking two random solutions from the set and comparing the return value of their fitness function. The solution with the higher fitness will end up winning and get chosen. The paper [Eib01a] describes the selection process as a way to exploit current solutions and improve their fitness function. We changed its implementation to suit our need of having two parents/ two children and maintaining the population's numbers.

3.2.2 *Uniform crossover*

In real-life, crossover would happen when same type chromosomes meet during meiosis. Both can switch different parts when they are lined up. In our virtual world, we will avoid this method since our chromosomes do not have the same length as the real life DNA. C.R.O is crucial to G.As, it explores the set of solutions and looks for new ones that weren't available beforehand. In our paper, we shall use the uniform crossover. It separates the linear representation we evolve into n bits and randomly gives one of the parents' bits to the first child and the other parent's bit to the second child (based on a uniformly generated number). Therefore, such a method ensures we have explored more solutions. If we only had the selection process, it would mean staying in an area where the fitness might converge to a sub-optimal value.

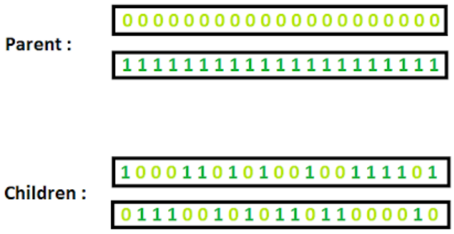


Figure 3: Graph of the uniform crossover with two parents and two offsprings

3.2.3 *N-point mutation*

Mutation in biology is described as the alteration of the DNA sequence of a bacteria, an organism, etc. It is said to be able to create changes in one's phenotype (observable characteristics). Nature of the changes are to be determined during the life of the individual whose DNA has changed. In traditional genetic algorithms, there are a few methods to reproduce the mutation (M.U.T) step (in a binary linear representation) such as the random mutation, which randomly selects a bit and changes its

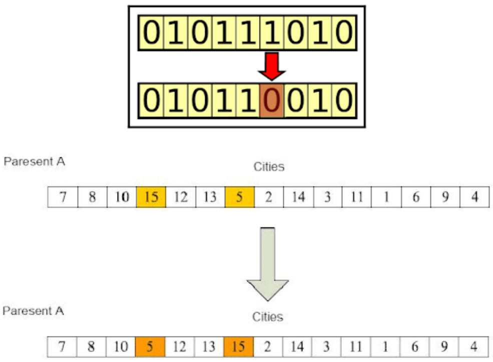


Figure 4: Comparison of the random mutation (binary-coded chromosomes) [Av01a] and the swap mutation (decimal-coded chromosomes) [Shih01a]

value, the swap mutation which selects two genes randomly and switches their value.

These mutation methods can bring new information to light but we consider them quite limited for tuning our agent's DNA. That being the case, we developed a novel yet simple algorithm called the n-point mutation which steps are developed in Algorithm 3.

Algorithm 3 N-point mutation algorithm

```
Get an agent's DNA
if length of the DNA not reached then
    Generate a uniformly distributed float between 0 and 1
    if generated float > 0.79 then
        change the value of the bit
    else
        Go to the next bit
    end if
end if
```

Keeping the threshold to a high value (0.79) helps us create a balance between creating new genetic representations for the offspring (exploration of the solution search space) and preserving the DNA structure from an unnatural disruption. We used a derivation of [Hol01a] to be able to fine tune and observe how a DNA structure would evolve through time and a limited amount of generations. Our agents' DNA are represented by a binary array of size 8, which could be changed. We evaluate them with a score that depends on their theoretical aptitude to survive in their environment, select them via B.T and reorganize them in an array for C.R.O, ensuring protection from premature convergence to a suboptimal solution. We apply the N-point crossover method as well as the N-point mutation with the hope of discovering more solutions during our run. To gap between reality and our world by changing the representation of the DNA to a linear array with {0, 1, 2, 3} as the possible genes instead of a binary array.

4 RESULTS AND FURTHER POSSIBILITIES

Creating the simulation was half of the process. The other half is to run some experiments with it. In order to obtain results with our G.A, we created a code which saves the agent species, survivability score, generation and DNA structure. In the following sections, we will analyze and interpret the data, discuss the results, and draw some observations. We would also like to define the next possible steps for this research, keep what is positive and understand where the simulation could be improved. (More data is available in the full thesis [Hba01])

4.1 Analysis of the results

These runs were made by changing the formulas and making the genes more significant for each attribute, and we did runs for four cases

- 75t and 150t with a M.U.T threshold of 0.79
- 75t and 150t with a M.U.T threshold of 0.83

We hope to see if making mutation a rarer occurrence would make it easier for the population to stay stable or if it would make it harder to survive. In this paper, we will only cover the most interesting cases out of the 4 : 75t - 0.79 M.U.T and 75t - 0.83 M.U.T.

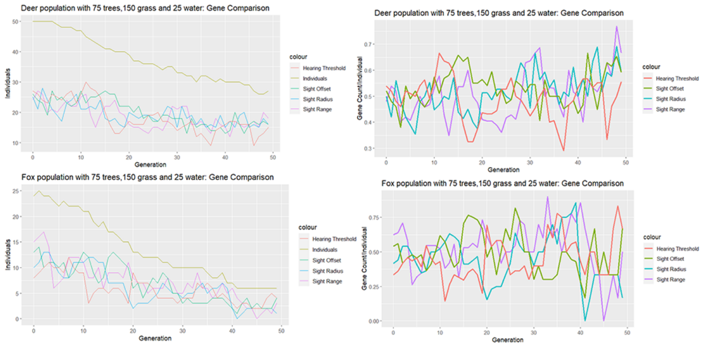


Figure 5: Fox and deer populations: Gene comparison and individuals evolution (75t - 0.79 M.U.T)

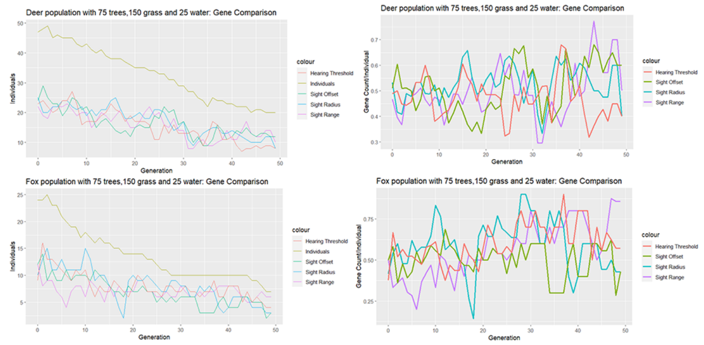


Figure 6: Fox and deer populations: Gene comparison and individuals evolution (75t - 0.83 M.U.T)

In the 75t - 0.79 M.U.T graph, the deer species seems to be thriving with a count of approximately 26 agents by the last generation. We also observe an augmentation of population as well as a higher count of deers individuals possessing the 3 genes for Sight just before generation 50. However, we also notice that, at the same time, the gene for Hearing gains a higher gene count/individual (g.c/l) too. A hypothesis could be that the M.U.T threshold is too low and makes gene mutation too common in this case. The fox species doesn't seem to be thriving compared to the deer species as their final numbers decreased by 72% (25 agents to 7 agents). We can interpret this as the result of competitiveness between the species and the foxes themselves (inter and intra species competitiveness). Furthermore, we note a decrease in population multiple times as the count of sight genes drops in the foxes' generation DNA. By generation 41, the number of individuals seems to stabilize, as the number of environment actor (E.A) always stays the same and the number of deers also decreases, leading to a better distribution of resources between all the agents in the 75t case. In the 75t - 0.83M.U.T case, the deer species doesn't thrive as well as their 75t - 0.79M.U.T counterpart. Their final population numbers decreased by approximately 25.93%. Gene count/individual pattern seems to be close to the 75t-0.79M.U.T case, except for the Hearing Threshold gene, whose g.c/l decreased drastically because of the lower mutation rate. The fox species seems to be thriving more in this case than their 0.79M.U.T counterpart. In fact, the hypothesis that sight genes count/individual and survival rate was linked tend to be validated by our data as we can see in Figure 7. In generation 10, we had an increase of $\approx 14.3\%$ for the fox population as well as an increase of 23.52% of Sight Range gc/l. Having mutation as a rarer occurrence could be beneficial for the system in its whole as it maintains balance between the two species used in the simulation. Making it a more common occurrence seems to tilt the favor to stronger species (deers) as the weaker species will not be able to focus on genes that should close the gap between them. It could also be conceivable that a higher number of resources isn't necessarily the best as their distribution in a system should be taken into consideration too. Foxes in the 75t - 0.79 M.U.T showcases that hypothesis. We will try to discuss more on the data, about other hypotheses and various possibilities for this research.

4.2 Discussion

The research is conducted to gain understanding about which genes are the most significant in our DNA structure, depending on the environment and other parameters. In the previous part, we analyzed the gene count/individual (g.c/l) for a few cases but we didn't focus on the significance of the genes.

< 0.001 : xxx	< 0.01 : xx	< 0.05 : x	> 0.05 : X

Table 2: Legend of the significance levels used in the following table

M.U.T	Fox 75	Fox 150	Deer 75	Deer 150
SRange	X	xx	xxx	xxx
SRadius	X	xxx	xxx	xxx
SO	X	x	x	xxx
HT	xx	xxx	xxx	xxx
ASize	X	X	x	X
ASpeed	X	X	X	X
AD	X	X	X	X
AH	X	x	X	X

Table 3: Fox and deer populations : Gene significance (second runs - 0.83M.U.T)

We note that AgentSize, Speed, Damage and Health are the most significant. It can be explained as all those define how an agent will survive in its environment, find resources and fare against its enemies. A real life example could be the gazelle and the lion. A lion might be stronger than a gazelle but a gazelle can outrun a lion on long distances. All sight genes significance for deers are inferior to 0.001 except for SightOffset in the deer75 (is inferior to 0.05). For the fox75, sight genes all possess a significance higher than 0.05. That could be explained by the scarcity of the resource and the competition the species has to go through with deers. Fox150, doesn't seem to possess that much significance for SightRadius (inferior to 0.001) and SightRange (inferior to 0.01) as the resources are abundant and fairly close to the foxes. The lower distance between foxes and resources could also explain why its AgentHealth significance went from X to x in table 3. Less health and less energy is needed to find resources. The Hearing Threshold gene always has a significance inferior to 0.001 (except for fox75 in 0.83 M.U.T). This gene, no matter this value, will never hold a significance in this version of the simulation as the hearing sense of agents wasn't replicated. In real life, foxes have a very sensitive hearing and use it to gather information about their environment. It enables them to hear sounds from behind, better than a deer, giving them a clear advantage during hunting. During all the runs, no combat happened between prey and predator as each agent teleports to its next location and never managed to get closer to its enemy. If agents moved in a traditional way, we could probably observe multiple foxes teaming up to chase one or multiple deers, cornering them and defeating them in battle. Agents shouldn't be dying as combat doesn't happen. Nevertheless, population numbers decrease as a result of an U.E5 bug where the calculation of a random position inside the navigable radius leads to the void and an automatic destruction of the falling agent by the engine.

4.2.1 Outlook and Future Directions

In this paper, we use a binary representation for the agents' DNA, to simplify operations. As stated in that part, life doesn't represent genes with a {0, 1} set but with a {C, G, A, T} set. The letters that creates the real DNA are nucleobases, which can only link in a certain way :

- Adenine (A) with thymine (T)
- Cytosine (C) with guanine (G)

Having 8 genes with {0, 1} as the values is simple but limited, as we only have 256 possible combinations for all agents. Using a numbered set like {0, 1, 2, 3} could be the solution as it would make the number of possibilities equal to

$$4\verb|^|8 = 65536$$

However, DNA is a polymer in the form of a double helix where nucleobases only link a certain way as stated before. Replicating the double helix form would mean using a non linear representation for the AT, CG pairs. It is impossible to keep using a G.A for this kind of representation as a G.A is only usable for linear representation per the definition of its operators (selection, crossover, mutation). We could use a set equivalent to {AT, CG} in order to achieve that with a G.A but it would basically lead to the same result as using {0, 1} as our gene set (256 combinations) It is possible to completely change the DNA structure to get more combinations but that would mean changing how we calculate the A.A. It is unnecessary to modify the DNA as {0, 1} can be considered like {AT, CG} and the mathematical model seems to hold itself vis a vis the ecological dynamic patterns for our species survival. A direction the study should take is the correction of the agent teleporting, which would lead to more combat between species and hence more data concerning the predator/prey problem, as well as refining the agent attributes calculations. In the current state of the simulation, we overlook certain complexities real organisms face, simplifying interactions between agents. Moreover, we didn't develop parameters that have the potential to turn around the ecological dynamics in our simulation such as temperature variations, diurnal cycle, season changes, natural catastrophes, diseasesand the effect of human activities on the environment. Those should prove themselves extremely complicated to replicate in the U.E5 engine but could have a really interesting effect on our results and should be taken into consideration for the future of this research. It is also possible to expand the range of species in the simulation as only two species exist in the virtual world for now. We hope that bringing new species with unique genetic traits and behavior will help us observe more complex population dynamics and ecological relationships. Another key component for this research is the investigation of the role of mutation and its effect on survival and evolutionary process. Exploring different thresholds (example : 0.79 M.U.T, 0.83 M.U.T) and adding other types of mutation could enable us to gain understanding on the M.U.T operator's influence on survival, persistence of species and genetic variations over time.

5 CONCLUSION

The problematic behind this research was studying the evolution of agents inside a virtual world. We sought to understand the mechanisms of DNA transfer between generations, identify the genes of utmost significance and pinpoint parameters influencing survival and gene significance. Those were some of the questions we asked ourselves while working on this project. The methodology used to respond to these questions and the problematic were to create an artificial life simulation in Unreal Engine 5, which aims to mirror real life, reproduce animals' traits and behavior, run a genetic algorithm with binary tournament, uniform crossover and n-point crossover on each generation and analyze the data to find what genes hold the most significance per case. Analyzing the collected data during multiple runs enabled us to understand which parameters influence the most survival and a species' success in our virtual environment. Some species would thrive more than others in the virtual life environment, and thus due to recreating or not senses like sight and hearing. An example of this is how deers always managed to keep between 60% and 40% of their initial population whereas foxes always kept between 40% and 20% of theirs. It is explained by how recreating the hearing sense in Unreal Engine 5 showed itself to be a challenging task compared to the sight sense's recreation. In our simulation, deers were bigger than foxes and held a better sight sense than them, hence giving them an advantage when searching for resources or finding predators in case of danger. The analysis of the results denotes how important Sight genes were during all the runs. Resource availability and the absence of competition internally and externally of a species is a key factor for survival and a species' success. We noted that mutation, as a process, is important to create genetic diversity but a low threshold for mutation would impact agents' life and thus, survival in a negative way. Table 3 shows other genes held a lot of significance - AgentSize, Damage, Speed and Health. The genes are the ones defining how well an agent interacts with its environment (A.Size) and enemies (A.Speed and A.Damage) and how long it can search for food before needing to rest (A.Health). While the simulation, collected data and their analysis may bring to light some new information about ecological dynamics and artificial life simulation, we should acknowledge the limitations of our research. Agents do not move in our simulation, they teleport, which causes a problem in terms of chasing/running away between predator and prey. Agents' behavior is quite simplistic and could be improved to create more cases to study. Our formulas for calculating agent attributes remain quite simple, refining or modifying them might be a route to pursue in this research's future. The paper [Ric01a] created an artificial life simulation where certain parameters like predator/prey awareness, flocking or herding can be enabled or disabled. It focuses on the cycle of life and death and the predator/prey problem whereas ours focuses on DNA transmission and evolution in a virtual world. Implementing these parameters in the future of our research, as we implement newer species, could lead to a better and deeper understanding of ecological dynamics and predator/prey problem. This research endeavored to explore the complex concepts of evolution and ecological dynamics using UE5 and artificial life simulation. Mimicking animals' behavior and the environment in a realistic manner proved itself a formidable challenge, but results were satisfactory. The paper contributes to bringing new knowledge as it underlines the significance of resources availability, mutation threshold and sensory capabilities. Given how this research overlaps an array of fields, it has the potential to benefit our knowledge of evolution if resources and time were allocated to it.

-

ACKNOWLEDGMENTS

I would like to express my sincerest gratitude to a few people that helped me achieve my thesis.

- My French supervisor, Dr. Alexandra Bac.
- My Japanese supervisor, Dr. Okubo Masashi.
- Ms. Rhiannon Follenfant

I also thank my family and friends for their continuous support as well everyone else that was involved and helped me during this paper, no matter how much they were involved.

6 REFERENCES

[Hol01a] Holland J. Adaptation in Natural and Artificial Systems, 1975.

[Ric01a] Richards D, Jacobson M. J. Evaluating the Models and Behaviour of 3D intelligent Virtual Animals in a Predator-Prey Relationship, 2012.

[Geo01a] Georgiev M, Tanev I, Shimohara K, and Ray T. Evolution, Robustness and Generality of a Team of Simple Agents with Asymmetric Morphology in Predator-Prey Pursuit Problem, 2019

[Eib01a] Eiben A. E, Schippers C. A. On Evolutionary Exploration and Exploitation, 1998

[Hba01] Hassene Hba Ben Amar, Masashi Okubo. Study of Evolution in Virtual Worlds. Neural and Evolutionary Computing [cs.NE]. Doshisha University; Polytech Marseille, 2023.

[Shih01a] Shih-Hsin C. and Mu-Chung C., Operators of the Two-Part Encoding Genetic Algorithm in Solving the Multiple Traveling Salesmen Problem, 2011

[Av01a] One-point mutation figure: `https://www.analyticsvidhya.com/blog/2021/06/genetic-algorithms-and-its-use-cases-in-machine-learning/`

[cheetah01] Fastest animals on earth: `https://www.britannica.com/list/the-fastest-animals-on-earth`

[UE5] Unreal Engine 5 Documentation : `https://docs.unrealengine.com/5.0/en-US/`

[UE5Random] Random function documentation in UE5: `https://dev.epicgames.com/documentation/en-us/unreal-engine/BlueprintAPI/Math/Random?application_version=5.0`

[perlin01] Perlin noise `https://en.wikipedia.org/wiki/Perlin_noise`

7 ANNEX

- The source code of this project can be accessed via the link : `https://github.com/hassenebenamar/Research_Project`

Analysis of Natural Lighting Condition for the Digitization of Artwork in an Art Gallery Interior

Irena Drofova[0000-0002-8543-0098]
Tomas Bata University in Zlin
Faculty of Applied Informatics
Nad Stranemi, 4511
760 01, Zlin, Czech Republic
drofova@utb.cz

Milan Adamek[0000-0002-0306-6852]
Tomas Bata University in Zlin
Faculty of Applied Informatics
Nad Stranemi, 4511
760 01, Zlin, Czech Republic
adamek@utb.cz

ABSTRACT

The paper discusses the analysis of natural lighting conditions for digitizing art. The emphasis is on a realistic 3D digital reproduction of a work of art in natural lighting conditions in the interior of an art gallery. The art object is scanned and digitized in two natural lighting conditions. The photogrammetry method was used for a realistic 3D reconstruction of the artwork. This experiment aims to analyze the influence of lighting conditions on the quality of 3D reproduction of an art object concerning image processing and color reproduction. In this study, no accessories were used to increase the quality of the captured image, such as reflective and diffusive plates or lights to illuminate the art object. The art object was scanned and digitized in two natural lighting conditions. This study aims to analyze the influence of actual lighting conditions on the quality of a realistic digital 3D reproduction of a work of art.

Keywords

3D model, artwork digitization, photogrammetry, image processing, point cloud, color, art gallery

1. INTRODUCTION

Currently, digitization processes are reflected in all areas of human activity. Digital technologies are used in commercial, scientific, and artistic fields. [Näs20a] In art in particular, emphasis is often placed on the highly realistic quality of digital reproduction. Digital technology and image processing processes are projected into artistic creation. [Gul18a] Nevertheless, the digital reproduction of works of art is still a great support for the application of new procedures and methodologies such as machine learning. [Cst20b] Especially in connection with new trends such as 3D and virtual presentations in the online environment. [Bia17a]

3D realistic digital reproduction of a work of art also brings many challenges and unsolved problems in image processing. The high-quality reproduction of the object in conjunction with its texture and color are often variable due to light and weather conditions. The method and procedure of digitization depends primarily on the final output of the digitized object. It can be 2D and 3D printing, 3D online presentation or using the object in a virtual and augmented reality (VR/AV) environment in interaction with the user.

Realistic 3D digital image reproduction aims to get as close to the original. The same attribute for color evaluation of reproduction quality is directly related to light and human vision. Colorimetry, color and human vision deal with this issue. [Mol24b] This text responds to current trends in art digitization and the issue of realistic digital reproduction using the photogrammetry method. [Cer20c] Also considers the use of LiDAR (Light Detection and Ranging) sensors. [Gue24c]

The presented experiment aims to find to what extent the ambient light conditions can affect the 3D digitization of the artwork in connection with the chosen modeling method. The following sections describe the digitization of the artwork using the photogrammetry method and LiDAR scanning in daylight in the interior. The influence of lighting conditions on the color reproduction of the 3D model is analyzed in 3D point cloud models of one precisely defined color. The findings from this experiment are used for follow-up research in the field of realistic 3D digital reproduction of works of art.

2. ARTWORK DIGITIZATION

With the development of digital technologies and sensing devices, image digitization processes and graphic software for image processing have evolved and improved according to the type and purpose of the resulting output, as well as the development of digital and 3D printing and 2D / 3D online and virtual presentations. This section describes image digitization using ground image photogrammetry to create a realistic 3D digital model.

https://www.doi.org/10.24132/CSRN.3401.43

An artwork for creating a 3D model

The artistic object for 3D digital reproduction was painted with acrylic paints on canvas. The artwork is dominated by green and brown colors, as seen in Figure 1.

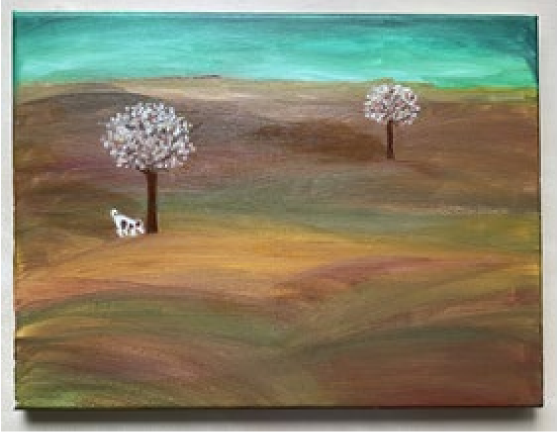


Figure 1. Artwork object - acryl on canvas

The object was captured in the interior of an art gallery, in which the daylight conditions were suitable. This experiment influences these lighting conditions on the resulting color reproduction of the resulting 3D model.

Digital Image Capture

An innovative mobile device with LiDAR technologies was used in this experiment. A smart tablet from Apple, which has a high-quality camera with high resolution, was chosen to capture and digitize the work of art. This smart device was used for a 3D reconstruction of a work of art using terrestrial image photogrammetry. The experiment used the free Scaniverse application to compare the quality of a 3D model, which is intended directly for the 3D digitization of objects and spaces using a LiDAR sensor. Both methods are described in the following section.

The Photogrammetry Method

The SfM (Structure from Motion) photogrammetry method calculates the position of an object in 3D space based on the description of information obtained from individual images taken from multiple angles. In the case of a specific object, the 3D reconstruction described below includes 17 photographs. From the basic information contained in the sparse point cloud, the points they create in the model, or they build a dense point cloud, and this large number of points already concretely displays the object and its position in space. This creates a complete point 3D model. This cloud of points is also a source of information about color values. The individual steps of creating a 3D model are visualized by the Figure 4 in the next section.

3D Reconstruction by LiDAR Sensor

To compare the quality of the digital reproduction of the artwork, a 3D scanning method using a LiDAR sensor on the same sensing device, the iPad 11 Pro tablet, was chosen. The Scaniverse application was used for scanning purposes. The resulting image before export to the 3D modeling software Agisoft is shown in Figure 5 in the next section.

3. COLOR SEGMENTATION

In this experiment, in which a real object is transformed into a digital form, the RGB (red, green, blue) color model and the sRGB color space (gamut) are used. In the Agisoft 3D modeling, the SW environment can work with information about the color of individual points or group points in a Dense cloud, as shown in Figure 2.

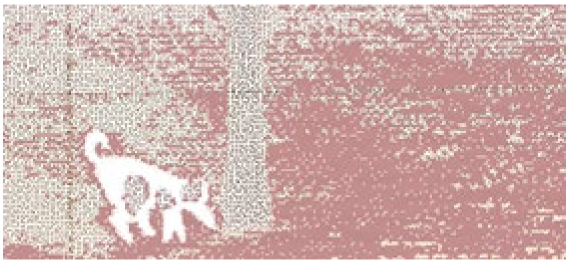
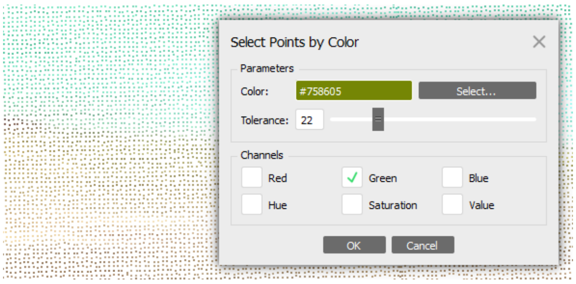


Figure 2. Definition points of the color #758605

As Figure 2 shows in SW, Agisoft can define the exact number of points based on the definition of colors (Hex) and work with RGB and HSV (Hue, Saturation, Value) color models. Figure 10 shows the points in the dense cloud 3D model that carry color information. That means we can define the points with a color value of #758605 in the green color channel, as shown in Figure 3.



Select Points by Color

Parameters

Color: #758605

Tolerance: 22

Channels

☐ Red ☑ Green ☐ Blue
☐ Hue ☐ Saturation ☐ Value

Figure 3. Points #758605 in the 3D model Dense cloud

4. VISUALIZATION

This section visualizes the creation process of a digital 3D model of a work of art using the photogrammetry method and the LiDAR sensor. The Figures 4 and 5 shows the segmented color measure with color value #758605 in the 3D model.

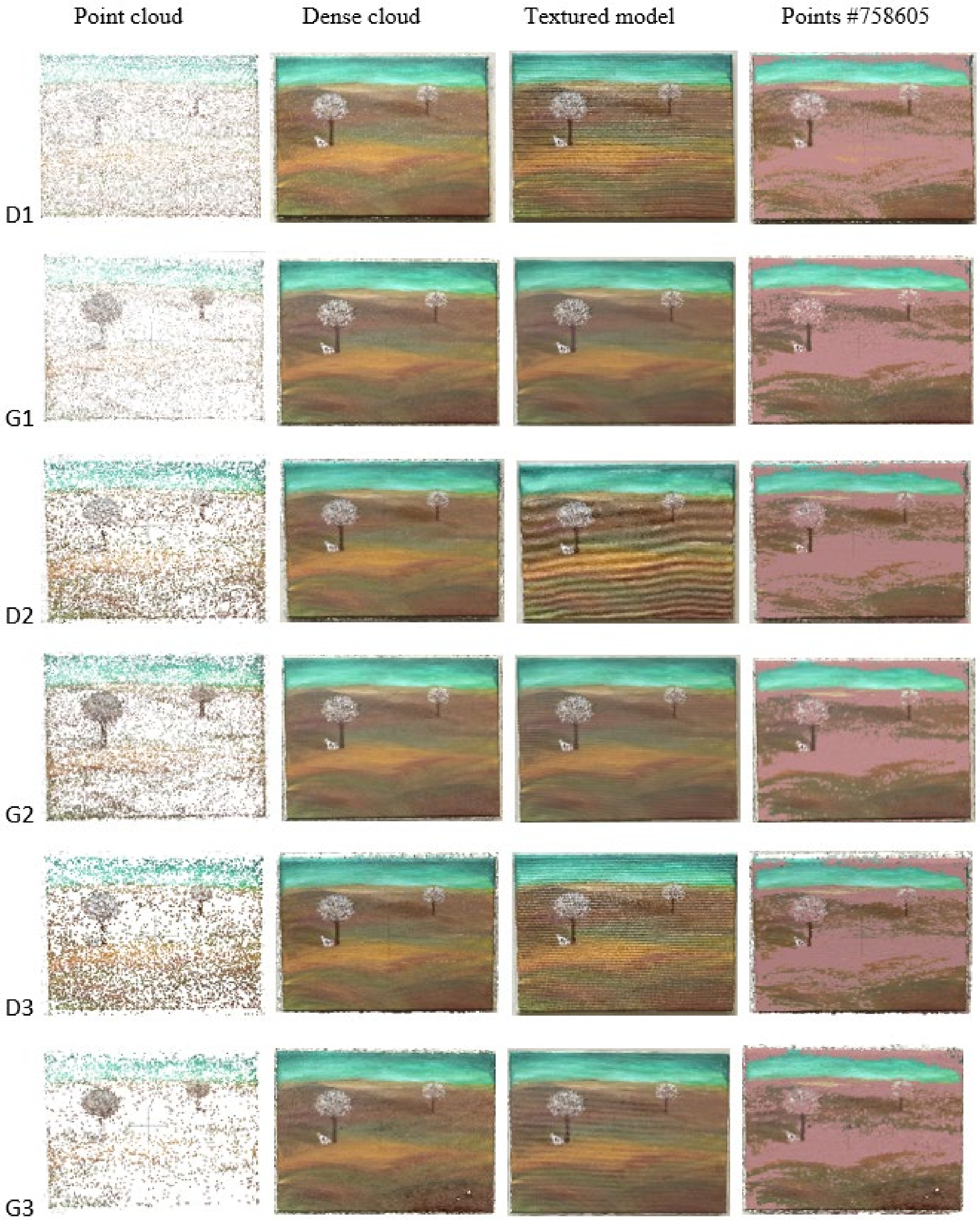


Figure 4. Points in Dense cloud by Color

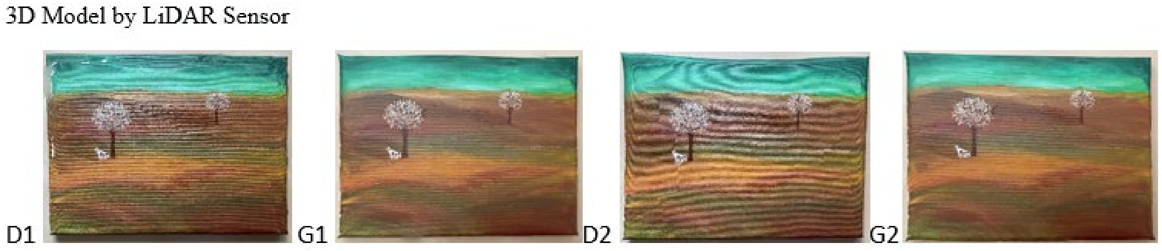


Figure 5. 3D models by LiDAR sensor

Figure 4 visualize the 3D models of the object in individual digitization steps by SfM method. 3D models marked D1 visualize the 3D reconstruction of the object in natural daylight. The G1 images visualize the same 3D models of objects in natural indoor darkness. Three series of 3D models in denim light D1-D3 and three series of object models in dark G1-G3 were produced. Even during the first visual analysis of the image, significant differences in the created 3D models are visible.

SfM	**Images**	**Point Cloud**	**Dense Cloud**	**Points #758605**
D1	24	13 828	383 170	236 418
G1	17	10 710	413 688	248 703
D2	13	9 327	405 385	221 648
G2	29	16 219	310 913	178 275
D3	8	7 100	325 107	194 873
G3	9	4 617	363 348	208 232
LiDAR	**Images**	**Point Cloud**	**Dense Cloud**	**Points #758605**
D1	/	/	21	/
G1	/	/	21	/
D2	/	/	23	/
G2	/	/	18	/

Table 1. Values of the Points in 3D models

Figure 5 shows the 3D models created by LiDAR technologies and using the Scaniverse mobile application on a mobile device. 3D models D1 and D2 were captured in natural daylight, and object models G1 and G2. However, this 3D modeling method cannot generate a dense point cloud with enough points to identify the points with color value #758605, as shown in Table 1. The different differences in 3D reconstruction require different image processing.

6. DISCUSSION AND CONCLUSIONS

The contribution contains a partial issue of 3D realistic digital reproduction of a work of art. The experiment was carried out in the interior of an art gallery. The ambient light conditions of daylight and dusk were used to take the picture. The iPad 11 mobile device camera and LiDAR sensor captured the object. This work aimed to compare the effect of light on color reproduction. The photogrammetry (SfM) method was used to create the 3D model. A LiDAR sensor also scanned the object. A visual analysis of the 3D texture models was performed, and it is clear that direct daylight is not suitable for this type of 3D reconstruction. The same visual result can also be observed on 3D models of LiDAR sensor scans. Table 1 contains the attributes for object reproduction analysis, including the number of generated points of points with a well-defined color value of #758605. The minimum number of points was generated in the case of 3D captured by a LiDAR sensor. This sub-experiment using the mentioned 3D modeling methods shows the research direction in the realistic digitization of art.

5. ACKNOWLEDGMENTS

This research was based on the support of the International Grand Agency of Tomas Bata University in Zlin, IGA/CebiaTech/2024/004.

6. REFERENCES

[Näs20a] Näslund A., Wasielewski, A., Cultures of Digitization: A Historiographic Perspective on Digital Art History. Visual Resources 36, pp339–359,https://doi.org/10.1080/01973762.2021.1928864, 2020.

[Gul18a] Gultebpe, E., Thomas, E., Conturo, M. M., Predicting and grouping digitized paintings by style using unsupervised feature learning. In: Journal of Cultural Heritage, pp. 13-23, ISSN 1296-2074. doi: 10.1016/j.culher.2017.11.008, 2018.

[Cst20b] Cstellano, G., Vessio, G., Deep Convolutional Embedding for Painting Clustering: Case Study on Picasso’s Artworks. In: Advanced Data Mining and Applications. Advanced Data Mining and Applications. pp. 68–78. Available on Internet: doi:10.1007/978-3-030-61527-7_5, 2020.

[Bia17a] Bialkova, S., Van Gisbergen, M.S.,When sound modulates vision: VR applications for art and entertainment. Available on Internet: doi: 10.1109/wevr.2017.7957714, 2017.

[Mol24b] Molada-Tebar, A. at all., Practical RGB-to-XYZ Color Transformation Matrix Estimation under Different Lighting Conditions for Graffiti Documentation. Sensors 24, https://doi.org/10.3390/s24061743, 2024.

[Cer20c] Cerasoni, J. at all, Do-It-Yourself digital archaeology: Introduction and practical applications of photography and photogrammetry for the 2D and 3D representation of small objects and artefacts. PLOS ONE. 17. e0267168. 10.1371/journal.pone.0267168. 2020.

[Gue24c] Guenther, M., at all, Tree Diameter at Breast Height (DBH) Estimation Using an iPad Pro LiDAR Scanner: A Case Study in Boreal Forests, Ontario, Canada. Forests 15, doi:org/10.3390/f15010214, 2024.

Analysis of different color recognition methods for active markers in a motion capture system

Przemysław Kowalski
Kipertech Consulting,
ul. Szwedzka 52,
30-315, Kraków, Poland

przemyslaw.kowalski@kipertech.com

Jan Mrzygłód
Kipertech Consulting,
ul. Szwedzka 52,
30-315, Kraków, Poland

jan.mrzyglod@kipertech.com

ABSTRACT

The article focuses on a method for reliably identify moving colored artificial markers in real-time. The marker was used to determine the 3D position in the space of the user(s).

The goal was to ensure that points were found and identified predictably and reliably by many cameras simultaneously, which, with appropriate calibration, merging, and processing of the data, could provide reliable information about the current 3D position of a given point in real-time. This information was crucial to other components of the broader vision system (VR platform).

The problems encountered and the remedial methods discussed in the presentation concern several aspects that we encountered during research, such as changes in lighting conditions, the quality (and stability) of the generated light and color, the dependence of color recognition on the distance of the light source from the camera matrix, aspects of light reflections, and many others. During our research, we analyzed various RGB/RGBW LED light sources from different manufacturers, which are characterized by different light generation characteristics. We also used a light diffuser. Using different sets of cameras and lighting conditions, we conducted several studies and experiments.

During the research, we managed to find basic colors for our marker-tracking visual system that met the goals. We have proposed an algorithm to deal with the problem and demonstrate the reliability of the visual layout with the algorithm. During our research, we used both conventional and alternative techniques related to ML.

Keywords

Color Space, Color Detection, Marker Detection, Image Processing, Computer Vision, Virtual Reality, Machine Learning, Real Time

1. INTRODUCTION

Our goal was to track users in real-time and visualize their position in VR. For our system Virtual Entertainment Enhanced Platform (VEEP), we proposed a vision-based motion capture system [9]. The system uses cameras to track and identify artificial active markers in 2D (application: tracker) and an additional module to calculate 3D positions from the positions reported by cameras (application: coupler; fig. 1). Systems that combine marker and motion tracking are a popular solution [10, 16]. In our application, we wanted to combine the real position with the virtual one so that the user feels the space naturally at any time.

Markers are placed on the player's head, and the aim of the system is to locate the player so that players can be located in real time. The players are wearing VR head-mounted display (HMD); the system should ensure safe and comfortable movement for the players, where their sensors are responsible for the orientation of the HMD and our vision system is responsible for the location, i.e., rendering real movement. Players thus move freely in space; their position is transmitted from the system to the HMD (Occulus Go or those based on Samsung Android phones) via a wireless network.

There are two reasons for placing the marker on the head: to reduce the obstacles between the camera and the marker, and to easily connect the position with the position of the player from which he or she is observing the world.

Such assumptions lead to requirements related to reliability and efficiency (working in real-time at not less than 60 fps, preferably 90 fps [13] or more). Our test experience confirms these requirements: delivering positions at a frequency of

https://www.doi.org/10.24132/CSRN.3401.44

approx. 30–40 fps induced poor player experience (VR sickness). The unpleasant feeling could be caused by the erratic data delivery itself (we were forced to ensure that the real-time requirements were met by using the Xenomai library on Linux computers (i5-6400T CPU 2.20GHz, 16GB RAM). We replaced the standard v412 Linux interface by communicating directly with the cameras using the LIBUSB library, and we experimentally selected a WiFi router with a 5GHz band to reduce data latency for an Android phone).

Image processing is simplified – pixels exceeding the threshold value are searched for (unless they have been used before), the neighboring area (with a lower threshold and an additional condition of maximum color change) is then filled in as being used, and the pixels are counted to determine the average color (RGB) and center of gravity (thus achieving sub-pixel accuracy of the marker indication). In extreme cases a distant marker may be represented by only 2 pixels.

The coupler calculates the 3D position from the submitted vectors (which originate at the focal point of the camera and indicate the marker in 3D). The procedure uses the method of calculating the nearest point to a given straight line, on a second straight line. For a pair of straights, it is performed twice and the two resulting points are compared. If their distance is within acceptable accuracy (as determined by the calibration quality of the cameras), an average is calculated. We apply the same principle to multiple pairs of cameras – averaging positions as long as they are within an acceptable distance and discarding them when they exceed this value.

Delays or instability also create a lack of confidence in the system among users who do not feel free to move around the room.

Experience gained during the construction of the system has shown that speed (low maximum delay) and certainty of location are decisive.

The system also requires tremendous reliability, assuming 99% correct marker identifications (color detection and recognition) per frame for a camera, At 120FPS and eight cameras (current configuration), this means 9.6 errors per second. We can't allow a single error in about twenty minutes of gameplay. Of course, this is a higher level of reliability than the described algorithm provides; we use numerous methods to verify and correct the result during further processing, but our primary goal is the speed and reliability of marker recognition, which has led to many simplifying assumptions: using active markers, darkening the room, or limiting the distinguishable features of the marker.

We can track markers through time using the distance between consecutive positions, but such a solution is prone to error because of the higher possibility of calculating and tracking errors.

Using the characteristics of markers can reduce calculation time and be less error-prone. Because we chose to work with (various) cameras working in the visible light range, the obvious characteristic of the marker was its color (we also considered the frequency), but it leads to two problems: we need to recognize the color of the marker with the changing light conditions, and we should distinguish the colors. In other words, our system should always be able to identify a color.

In our system, we use a dark room and active markers. We adopted such simplifications due to the requirement of fast (lasting a few milliseconds) and reliable identification of markers in the image. In practical implementations, "darkness" differs depending on the methods used to block external light sources and is practically never complete; it should rather be understood as a significantly limiting amount of light in a room than its complete elimination. The light conditions still differ from acquisition to acquisition, although the differences are reduced.

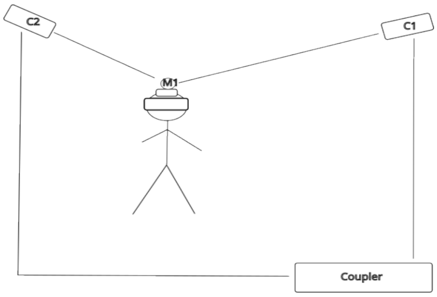


Figure 1. VEEP-system idea. Two (at least) cameras with trackers (C1 & C2) have a localized marker (M1) worn by a user. Vectors (C1→M1; C2→M1) are sent to the coupler. The coupler calculates user position, which is sent to the VR set (HMD) worn by the user.

Theoretically, the task should be simple: the combination of a color invariant and an active marker should result in a constant color.

In practice, the measured color of the marker changes. There are several reasons: the design of the marker itself (slight surface irregularities, quality of the diodes used, power supply), changes in lighting conditions, differences in color recognition by cameras, noise, and the and the small size of the marker expressed in the number of pixels (a large percentage of pixels partially representing the marker, whose color corresponds to the combination of marker color and background color).

Cameras

During the development our solution, we faced the problem of choosing cheap, fast, and reliable cameras. We have tested different models of cameras, and we chose the PlayStation PS4 cameras, which are significantly cheaper than their counterparts and equipped with a lens that produces very little distortion (although we also tested other solutions, and the system allows us to work with a heterogeneous set of cameras). While working with the cameras, we also faced a choice of operating mode—whether we preferred resolution or speed. In the practice of our issue—providing reliable positioning to a player moving with HMD — operating time proved critical. By choosing to run at 120FPS, we reduced the resolution requirements, which reflected positively on computation time and negatively on image resolution (and color recognition quality).

Active Markers

The active markers (Fig. 2) were developed as a programmed embedded system with color LEDs and a silicone sphere that uniformly diffuses the light. We have used different kinds of LEDs; we focused on two products, one of which is SMD (Surface Mounted Diode) and the other is THT (Through Hole Technology), and we were able to set one of the 360 colors (colors are set by using RGB LED combinations with filling; there is one color per degree of arc on the color circle). For simplicity, the results for both types of diodes will be described using the assembly method (THT, SMD). To control the LED, we use the microcontroller's PWM module, which generates a signal at 490 Hz.

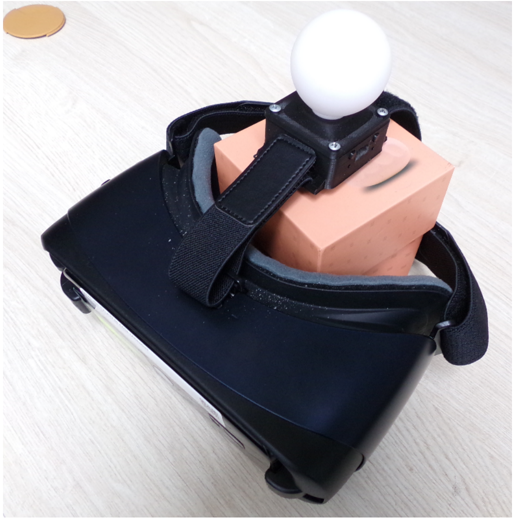


Figure 2. Active marker connected to VR head-mounted display.

We have prepared a test to see how the 360 programmed colors are visible by the camera; we displayed successive colors in a loop (the beginning is the same as the end). In our example, a PS4 camera was used.

The results show Figs. 3 and 4. We can see that not all changes in LED light are visible for the camera (for both types of markers). It means the real number of colors to find is much smaller than 360.

To propose which colors to use, we have analyzed the RGB (in fact, normalized RGB) distance between neighboring colors with different steps, which suggests basically six colors to use: red, green, blue, cyan, yellow, and magenta. Such colors give us the maximal distances, i.e., they are the easiest to detect.

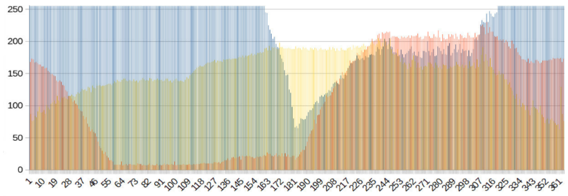


Figure 3. RGB components for SDM LEDs. (The blue bars correspond to the blue component, the yellow bars to the green component and the red bars to the red component.)

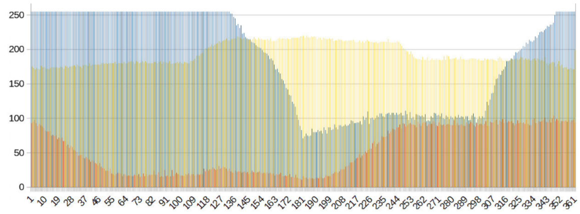


Figure 4. RGB components for THT LEDs. (Color designation – as in fig. 3.)

Search for solution

Basically – cameras return RGB images. The RGB image does not give us color-invariant components. The simplest solution is normalized RGB, where the red, green, and blue values are normalized by the pixel brightness. We have tested the normalized RGB, but the analysis of the output had to contain rules for at least three components and tolerate "gray" pixels (too light or too dark to represent marker color properly). The system worked with such a solution but was sensitive to light condition changes.

The color identification algorithm consists of image transformation into a chosen color space and a color classification method. The main color spaces used are: normalized RGB [2], HSV [3], YCbCr, LAB [4], RGB-L*a*b* [12], MCSS [5]. We have tried to use other color spaces: YUV (with two chrominance components: U and V), YCbCr [14], LAB, and HSV [1, 6, 7, 8, 11, 14, 15]. The best results (i.e. easiest color detection) were given to us by HSV, and we have focused on this color space.

Tests indicated that practically the best results, taking into account differences in distance and marker reproduction quality, are given by maximizing the spacing between colors in the H (hue) component.

Practical problems

Our active markers were developed to maximize color uniformity on the surface, but in fact, they are not uniform. It may be justified by the limitations shown in Figs. 3, 4, and 5 (and Fig. 5 for a marker seen by a camera): the color is distorted by too-light areas of the marker. We used the average color that reduces the observed distortion.

The other problems are connected with camera noise—the values of the pixel was changing from frame to frame.

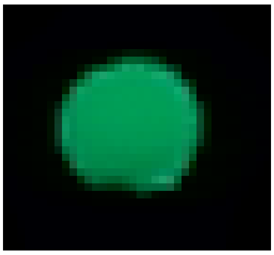


Figure 5. Example image of a marker captured by the camera at medium distance. (The image quality matches that of the camera, and visibility of the marker in the darkened room.)

Additionally, there was a problem with lightning conditions. We assume the system works in a dark room where the light sources are turned off (for artificial) or occluded (for natural). Additional sources of light are markers themselves; if many markers are used in the room, the whole room can be slightly illuminated. We keep the high contrast between the marker and the background, but the brightness of the marker (and its color) changes.

There is a similar problem with the marker's power supply; a discharged battery first changes the brightness of the marker, which may previously have affected the camera's perception or been perceived by a human being.

The basic research was done in the office, with limited distances between markers and cameras. We have overlooked the scale problem caused by the color of small marker images, i.e., a situation with the distant markers when we increase the size of the scene. The problem was even bigger because we have reduced image resolution (from 1280×800 to 640×400), which was caused by an increase in camera speed (from 60 fps to 120 fps), which is essential for VR immersion. The marker surface becomes less than 20 pixels at a distance of about 5 meters. With the decrease in marker size, the quality of color detection using hue (from HSV) decreases, and often, the hue changes for the same markers.

The problems identified above means that the color of the marker differs during the system life-cycle. We can increase the quality of the analyzed images, e.g., by equipping our system with higher-quality cameras (frequency greater or equal to 120FPS at higher resolution), thus reducing the problem of unevenness of color perception. However, such a decision affects the cost of the system; we considered the adopted solution of Sony PS4 cameras as a reasonable compromise between quality and price.

2. EXPERIMENTS

We have tested six markers, each in one color. The markers were in the same position, which facilitated the acquisition and limited the impact of changes in brightness. The hue was represented by one byte, i.e., the possible range for hue is 0..255. We have tested markers twice: first, as static (fixed position), and as moving marker (marker moved by the user) in a small office room.

	Static			Moving		
	Min	Max	Diff	Min	Max	Diff
MG	211	222	11	202	244	42
GN	82	85	3	76	85	9
RD	174	176	2	171	178	7
BL	0	10	10	0	39	39
CY	33	48	15	29	55	26
YL	127	131	4	113	154	41

Table. 1. Hue range for marker tests at short distances (less than 5m).

For the static marker – the changes in the hue value were small; for such conditions, we can define much more color range and precisely identify markers. For the moving markers, the differences are bigger, and we can see that the fixed ranges for camera types cause the marker to misinterpret the color.

We have proposed an algorithm that uses hue and marker tracking. If the marker can be tracked, we smooth the marker position and compare the hue with the previous color. If the previous color can be tolerated (i.e., the distance between hue and color hue limits is acceptable), the previous color is set; otherwise – we use hue ranges for colors.

The change of space made it necessary to work at distances of up to seven meters. Tests quickly showed that the method that works well in office conditions is effective up to approximately 5 meters. An additional two meters forced a change in the acceptable size of the marker. For the dark markers (displaying the primary color (red, blue, green) means one component of the RGB LED is lit, while the other three colors (magenta, yellow, and cyan) mean two components are lit and the brightness is higher), we have to reduce the minimal brightness of the acceptable marker pixel, which increases the influence of the camera noise. The color identification algorithm has become unreliable.

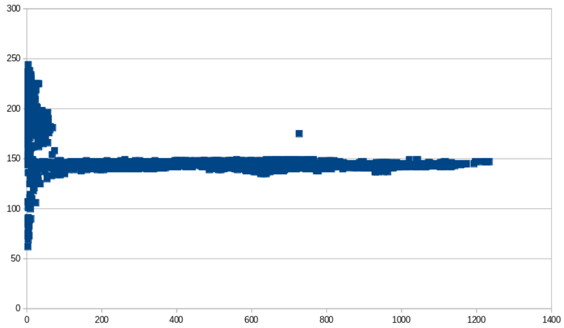


Figure 6. Relation between hue (y) and the size (x) of the marker in pixels (for yellow marker).

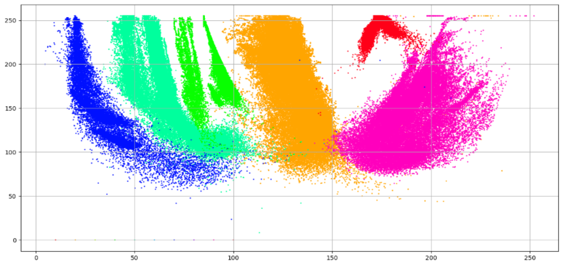


Figure 7. Pairs hue saturation for all six used colors. Hue: horizontal; saturation: vertical. The colors used correspond approximately to the colors of the markers.

As an example of the problem, we can see (Fig. 6) the hue values for one yellow marker in relation to the marker size.

A part of the wrong H values for the marker comes from the unnecessarily recognized marker reflection, but it is a potential source of error, and we should be ready to handle it.

The tests indicate that markers bigger than 20–25 pixels can be properly recognized (the ranges for colors do not overlap), but for smaller markers, overlapping ranges are possible. The tests also suggest that while hue ranges are often overlapped, a pair hue-saturation, or hue-value, is rarely overlapped, and triplet hue-value-saturation is even more rarely overlapped.

The data obtained for the calculation of the hue-saturation (fig. 7) and hue-value (fig. 8) dependence come from eight PS4 cameras for individual colors: blue was calculated using 21033 measurements of moving marker's color (worn and moved by the player, simulating gameplay behavior and using all available room space), cyan using 112769 measurements, green using 105709 measurements, magenta using 68132 measurements, red using 83070 measurements, and yellow using 124841 measurements. Totally, 515554 measurements were used with the final setting of the system (camera parameters, etc.).

On figs. 7 and 8, we can see the overlapping areas for hue ranges: blue, cyan, and green have seamless transitions; a similar situation can be seen for green and yellow, and red and magenta (the magenta range covers the whole red range).

For practical purposes, we have reduced our usage of blue (we have some problems with this color for marker detection: the marker was poorly detected and color was sometimes confused with cyan) and proposed a new algorithm for color detection.

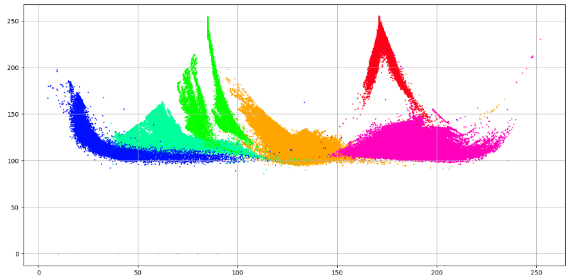


Figure 8. Pairs hue value for all six used colors. Hue: horizontal; value: vertical.

3. PROPOSED ALGORITHMS

There are two main procedures: color recognition and marker tracking/recognition.

The first procedure is used for the base recognition of the color: we have developed tables (fig. 9, 10) for this application that are based on the data from fig. 7, 8; the space has been transformed into a table (one table field represents space 1x1).

In the table we store:

- certain colors: where hue-saturation and hue-value gives the same color, or where certain color was matched with uncertain;
- uncertain colors; for pairs where data comes from at least two color markers, the more likely color is pointed;
- unknown color – there are no data and no guesses what color the pair represents.

The color of the found marker is represented by two parts: the main color and the complementary color. At the same time, both colors can be described as 'uncertain'.

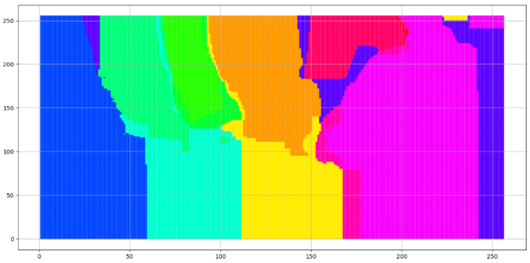


Figure 9. Hue-saturation table for color detection (man-made based on measurement data).

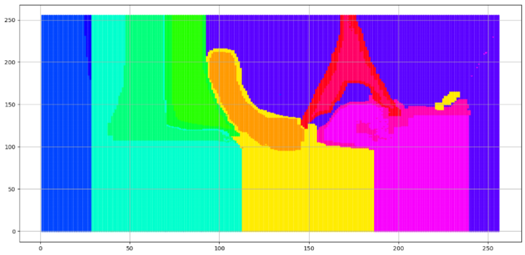


Figure 10. Hue-value table for color detection (man-made based on measurement data).

The algorithm calculates HSV for the found marker and looks for the proper color in both tables (hue-saturation, Fig. 9, and hue-value, Fig. 10).

```
//color:
// mainCol, complementaryCol, certain

colorUsingHSV(h, s, v) → color
if hsCol[h,s].val==hvCol[h,v].val
  return color(hsCol[h,s].val, null, true)
else
  if (hsCol[h,s].certain) and
            (hvCol[h,v].certain)
     return color(hsCol[h,s].val,
            hvCol[h,v].val, false)

  else if (hsCol[h,s].certain)
     return color(hsCol[h,s].val, null, true)
  else if (hvCol[h,v].certain)
     return color(hvCol[h,v].val, null, true)
  else
     return color(hvCol[h,v].val, null, true)
```

The second procedure is used for markers tracking:

```
pos2d = findCenterOfGravity(marker)
rgb = findRGB(marker)
hsv = rgb2hsv(rgb)
col = colorUsingHSV(hsv)

for old in allMarkersPrevFrame:
  spatDist = dist(odl.pos2d, pos2d)
  colDist = colDist(old.hsv, hsv)
  if (spatDist<maxSpatial) and
         (colDist<maxCol)
     new2d = (a*pos2d+b*old.pos2d)/(a+b)
     if (old.col == col)
       newMarkers.add(marker(new2d, old.col,
           hsv, true))
     else if (old.col.certain)
       newMarkers.add(marker(new2d, old.col,
            hsv, true))
     else if (col.certain)
       newMarkers.add(marker(new2d, col,
           hsv, true))
     else
       newColor = color(old.col, col, false)
       newMarkers.add(marker(new2d,
           newColor, hsv, true))
  else
    newMarkers.add(marker(pos2d,col,hsv,false))
```

The marker "is visible" (sent to the further elements of the system) if it was found in a series of markers in consecutive frames (with a given threshold). If the marker was not found in the analyzed frame but was found in the series of previous frames, it is still "visible" (if the visibility break was not too long). This increases the stability of our system.

4. EXPERIMENTAL RESULTS

Using a system of 8 cameras (PS4) and 4 computers, we collected data on calculated HSV triplets for markers. The tests were carried out in conditions imitating the target ones, i.e., the user walked with a marker at different speeds throughout the entire system operation area.

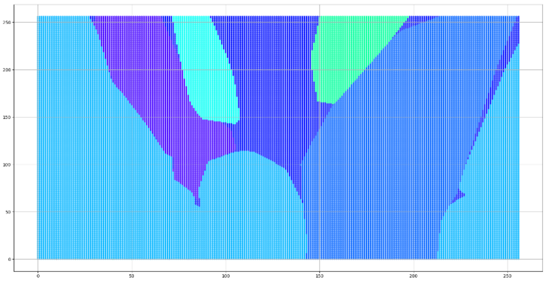
Tests were conducted separately for subsequent colors to ensure that the collected HSV triplets corresponded exactly to a given color. For the final tests we collected a new of 262281 samples divided into six colors. In the tests, we divided the dataset into a training and a test subset.

When tested for a sample of 268281 measurements, there were 15497 false matches using the above algorithm; that is 94.2% correct matches. In practice, the results obtained are corrected in the system based on the measurement history, and then the change in position is analyzed, practically ensuring that the room can be navigated safely and realistically.

We have tested some machine learning methods to compare results. We used saved test data to determine quality (ML techniques were not tested with a real-time system). SVC classificators (with

RBF kernels—independent classifiers for different colors) gave 1276 wrong outputs for 53657 HSV positions (size of test set; the remainder of the 268281 measurements was used as a learning set)—which means proper results where achieved for 97.6% (for polynominal kernels: 95.2%). Some colors were classified better (e.g., for yellow, red, and magenta, the proper results were above 99%), while others were classified worse—for blue, it was only 86.8%, and for cyan, it was 96.7%.

For decision trees, the results were similar: 97.5% properly classified triplets h, s, and v; the best results (better than 99%) for yellow, green, magenta, and red; the worst for blue (87.7%). We have also tested neural networks (simple perceptrons) from SciKit-Learn. Simple perceptron (for three parameters) gives us 97.7% proper results. The worst results were given by a blue marker – 91.5%. We have used the same method (but models were built for pairs: h, s, and h, v) to create new tables, and our simple algorithm yielded 97.1% correct matches. Perceptron build for h-s pairs, gave 64.5% proper answers, and for h-v pairs, gave 97.5% proper answers.



Figiue 11. Color maps h-s created by multilayer perceptron. Note: The colors do not correspond to the actual marker colors or the colors in the earlier illustrations.

The worst results where given by the blue marker; after eliminating it (it means using blue and cyan as one color), the neural network gave 99.4% proper results for triplets h, s, and v, while the neural networks for the h-s pair and the h-v pair gave respectively 99.2% and 99.4%.

We have used perceptron to create new color maps for h-s (fig. 11) and h-v (fig. 12). Tests suggest that new color maps give slightly better results with our algorithm.

We performed the above tests using Python (with the Anaconda environment and the SciKit Learn library), which we also used to build tables describing the relationships of h-s and h-v pairs with colors.

When testing the algorithms above, we were interested in quality—to identify candidates for system improvement. So we didn't test the execution time.

Otherwise, in the real system, we put great importance on the calculation time. There, we examined the time needed to calculate position information, and we built both programs (detecting markers and their colors, calculating 3D positions for markers) using the Xenomai library, which ensures real-time implementation. We assumed an 8-ms loop for both programs.

In the tested configuration our system consists of four computers to which 8 cameras are connected to (2 per computer), and each camera corresponds to one running marker detector application. One of the computers runs a “coupler" app, a program that calculates the 3D position from the positions sent by the marker detectors; and on another of the 4 computers, there is a server prepared in Unity, which is used to coordinate game events between client applications (running on stand-alone VR HMD).

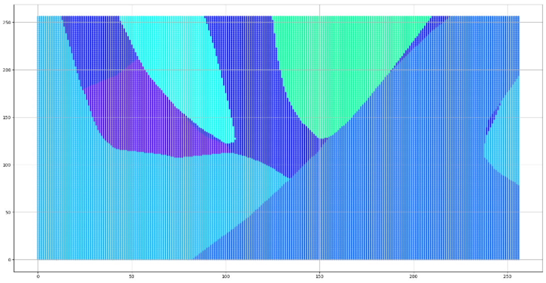


Figure 12. Color maps h-v created by multilayer perceptron. Note: the colors do not correspond to the actual marker colors or the colors in the earlier illustrations.

Average calculation time for 1003 test frames in a typical configuration (2 PS4 cameras connected via USB ports to a computer with Linux, 16GB of RAM, and an i5 processor, 2.20GHz; so two marker detection programs were running in parallel on one computer), total time for calculating the position and color of the marker (from the image acquisition to sending output data to the next program) was on average 1.51ms. The calculation in the second program, which determines the player's position from the sent markers and transmits it to the players, takes an average of 1.08 ms.

5. SUMMARY

To summarize, we developed an algorithm for identifying the marker color. We demonstrated that the changes in the observed marker are significant and require the development of an appropriate algorithm. The algorithm is simple, highly reliable, and fits our problem. We have also tested ML algorithms, and simple perceptrons give better results than our algorithm. While this is better solution, transferring it to a real-time tracker could present difficulties, so we only used this solution to improve the recording of the color tables for our algorithm. The transfer of neural network solutions to the real-time tracker can be done as part of the work to improve the system in the future.

Acknowledgments

This work was supported by the NCBiR (National Centre for Research and Development), project VEPP (Virtual Entertainment Enhanced Platform), no. POIR.01.02.00-00-0155/17-00.

BIBLIOGRAPHY

[1] Muhammet Fatih Aslan, Akif Durdu, Kadir Sabanci, *Shopping Robot That Make Real Time Color Tracking Using Image Processing Techniqes*, Journal of Applied Mathematics, Electronics and Computers, 5(3):62-66, 2017, DOI: {10.18100/ijamec.2017331881}

[2] Homayoun Bagherinia, Roberto Manduchi, *Robust real-time detection of multi-color markers on a cell phone*, Journal of Real-time Image Processing, 8:1-17, 2013, DOI: {10.1007/s11554-011-0206-9}

[3] Fabrizio Cutolo, Cinzia Freschi, Stefano Mascioli, Paolo Parchi, Mauro Ferrari, Vincenzo Ferrari, *Robust and Accurate Algorithm for Wearable Stereoscopic Augmented Reality with Three Indistinguishable Markers*, Electronics, 5, 2016, DOI: {10.3390/electronics5030059}

[4] Joseph DeGol, Timothy Bretl, Derek Hoiem, *ChromaTag: A Colored Marker and Fast Detection Algorithm*, Proceedings ofInternational Conference on Computer Vision, Venice, 2017, DOI: {10.1109/ICCV.2017.164}

[5] Hesam Eskandari, *Detection and tracking of sphere markers*, Master Thesis, École de technologie supérieure, Montreal, 2009

[6] Rabah Hamdini, Nacira Diffellah, Abderrahmane Namane, Abderrahmane, *Robust Local Descriptor for Color Object Recognition*, Traiment du Signa, 36(6):471-482, 2019, DOI: {10.18280/ts.360601}

[7] Allan Hanbury, *Constructing cylindrical coordinate colour spaces*, Pattern Recognition Letters, 29(4): 494-500, 2008, DOI: {10.1016/j.patrec.2007.11.002}

[8] Priyanto Hidayatullah, Miftahuddin Zuhdi, *Color-Texture Based Object Tracking HSV Color Space and Local Binary Pattern*, International Journal on Electrical Engineering and Informatics, 7(2):161-176, 2015, DOI: {10.15676/ijeei.2015.7.2.1}

[9] Przemysław Kowalski, Krzysztof Skabek, Jan Mrzygłód, Jan, *VEEP – The System for Motion Tracking in Virtual Reality*, Proceedings of 6th International Conference on Man-Machine, Cracow, 1:12-22, 2019, DOI: {10.1007/978-3-030-31964-9_2}

[10] Liu Jiamin, Chen Shuo, Sun Hongxing, Qin Yongxu, Wang Xibo, *Real Time Tracking Method by Using Color Markers*, Proceedings of \em International Conference on Virtual Reality and Visualization, X'ian, 1:106-111, 2013, DOI: 10.1109/ICVRV.2013.25}

[11] Martin Loesdau, Sébastien Chabrier, Alban Gabillon, *Hue and Saturation in the RGB Color Space*, Proceedings of International Conference on Image and Signal Processing, Cherbourg, 1:203-212, 2014, DOI: {10.1007/978-3-319-07998-1_23}

[12] Chanh-Nghiem Nguyen, Van-Thoai V, Nguyen Cong Ha, Developing a computer vision system for real-time color measurement – A case study with color characterization of roasted rice, Journal of Food Engineering, vol. 316, 2022, DOI: {10.1016/j.foodeng.2021.110821}

[13] Erin Pangilinan, Steven Lukas, Vasanth Mohan, *Creating Augmented & Virtual Realities: Theory and Practice for Next-Generation Spatial Computing*, O'Reilly, 2019

[14] Patrick Sebastian, Vooi Yap, Ross Comley, *The effect of colour space on tracking robustness*, Proceedings of IEEE Conference on Industrial Eletronics and Applications, Singapore, 1:2512-2516, 2008, DOI: {10.1109/ICIEA.2008.4582971}

[15] Minjie Wan, Guohua Gu, Weixian Qian, Kan Renm, Qian Chen, *Hue preservation based color space transformation for brightness-robust tracking*, Optik - International Journal for Light and Electron Optics, 144, 2017, DOI: {10.1016/j.ijleo.2017.06.073}

[16] Yueting Zhuang, Yunhe Pan, Jun Xiao, *Human Motion Capture Using Color Markers*, in A Modern Approach to Intelligent Animation: Theory and Practice, Springer-Verlag, Berlin Heidelberg New York, 59-75, 2008, DOI: {10.1007/978-3-540-73760-5}

Dynamic Many-Light Importance Sampling for Real-Time Ray Tracing

Pedro da Silva Freire[1][a]
pedro.silva.freire@tecnico.ulisboa.pt

João Madeiras Pereira[1][b]
jap@inesc-id.pt

[1] INESC-ID, Instituto Superior Técnico, University of Lisbon, Rua Alves Redol, 9, 1000-029 Lisbon, Portugal

November 2023

Abstract

One of the main performance-heavy areas in raytracing is light sampling. Light sampling is solved in a process called next-event estimation(NEE), where light samples are taken at each ray intersection. Since real-time rendering is an objective, instead of sampling all the luminaries, just a set of lights deemed more important are sampled by using a technique called Monte Carlo Importance Sampling. Aiming further acceleration, some importance sampling-based approaches build hierarchical data structures over all light sources, which results in high maintenance costs for dynamic scenes. This paper describes a two-level light bounding volume hierarchy (BVH) of mesh-based lights to accelerate light sampling while minimising the quality loss in dynamic scenes common in similar algorithms. Its main advantage is the ability to have dynamic lights without losing excessive accuracy or performance to maintenance operations. Our approach was developed as an extension to the Mogwai's PathTracer application from NVIDIA. The algorithm rebuilds the top-level structure in the CPU allowing it to retain its accuracy and refits only the bottom-level structures in need of updating on the GPU. The CPU rebuild is a relatively costly operation, to avoid excessive performance loss it is done asynchronously only being used in the next frame. We tested several quality metrics as well as frame times, our implementation achieved an up to 36% increase in MSE and 6% in SSIM with an average of 7% slowdown when compared with the single-level BVH.

Keywords: Ray Tracing, Bounding Volume Hierarchies, Vulkan Ray Tracing, Illumination, Acceleration Structures, Importance Sampling

1. Introduction

[a] [b] Until recently raytracing has proven overly demanding for real-time rendering. Now, with the hardware acceleration available in modern RTX GPUs and highly efficient algorithms real-time raytracing can be achieved.

One of the main performance-heavy areas in raytracing is light sampling. Light sampling refers to the problem of calculating lighting for a surface from all light sources in the scene without considering reflections(indirect lighting). This is done by tracing rays to lights to see if the light is occluded by any object along the way, for complex scenes this requires many ray intersections that are impossible to calculate in real-time as the number of rays to be traced each lit patch scales linearly with the number of lights.

Modern algorithms use importance sampling to reduce the amount of rays being traced. One popular approach to importance sampling is through the use of tree data structures[8]. These usually have problematic upkeep costs but have proven to be an adequate solution for real-time rendering.

1.1. Objectives

The goal of project was to implement a two-level light bounding volume hierarchy (BVH) to improve mesh-based light sampling and reduce quality loss in dynamic scenes. Based on Pierre Moreau's work, this BVH extends NVIDIA's Mogwai PathTracer in the Falcor framework, using the Vulkan ray tracing API for real-time renders.

We compared our two-level BVH with Mogwai's single-level BVH and the ReSTIR algorithm, using new metrics for better picture quality evaluation.

The submission includes a video comparing the single-level and two-level BVHs.

2. Background

2.1. Light Equation

The rendering equation introduced by Kajiya[3] is a mathematical formulation, to predict the intensity of light passing from a point x' in a surface to a point x, it takes into account intricate phenomena such as reflection, refraction, and shading, this equation enables us to calculate the appearance of objects with high accuracy and realism.

The equation is as follows:

$$I(x,x') = g(x,x')\left[\epsilon(x,x') + \int_S \rho(x,x',x'')I(x',x'')\,dx\right] \tag{1}$$

Where $I(x,x')$, is related to the light intensity going from point x' to point x in $joule/m^4sec$, $g(x,x')$, is related to geometry, it is zero if the surface of x' is occluded. $\epsilon(x,x')$, is related to the light intensity emitted from emissive surfaces from point x' to x. $\rho(x,x',x'')$, is related to the light intensity scattered from an emissive point x" to x by a patch of surface at x'. And S, is the union of surfaces.

The light equation is a useful tool for realistic rendering, however, solving the integral over S would require an in-

---

[a] https://orcid.org/0009-0008-6672-5940
[b] https://orcid.org/0000-0002-8120-7649

finite amount of samples. Monte Carlo Methods are used to solve quantitative problems in science through the use of statistical sampling[7]. Sampling can be improved by using importance sampling, meaning we try to take more relevant samples to increase quality at the same sample count. In our case, we want to estimate $I(x, x')$ using a limited amount of samples acquired from traced rays.

2.2. Mesh Light Sampling

When using mesh-based lights it is too computationally demanding to sample every emissive triangle. Furthermore, in more complex ray tracing algorithms, direct lighting calculations are performed many times per pixel, making it also excessively demanding to try to sample every light.

This research focuses on importance sampling algorithms for light sampling, mainly algorithms based on a BVH data structure. By sampling only a select few lights deemed more important good quality and performance can be obtained.

2.3. BVH Data Structure

The bounding volume hierarchy is a tree data structure where each node is associated with a volume of space and the lights contained within. The root node is associated with the entire scene, subsequent child nodes result from splitting the parent node's set of elements into smaller sets, one for each child node.

When building BVHs for light sampling, it is important to properly estimate each node's contributions. Alejandro Estevez and Christopher Kulla [2] proposed an algorithm to ensure that each node's importance is calculated with enough accuracy.

Additionally, Estevez and Kulla proposed a new heuristic for evaluating the quality of a node split. The new heuristic is called the surface area orientation heuristic(SAOH) it expands on the surface area heuristic by adding a light orientation component.

In this work we implement a novel approach using a two-level BVH introduced by P. Moreau et al. [5]. With a two-level BVH we can use both refits and rebuilds to achieve real-time rendering while preserving quality.

3. Implementation

3.1. Development Environment

NVIDIA's Mogwai is an open-source platform that is modular allowing fast prototyping of algorithms. Mogwai's modularity is achieved through the use of renderGraphs. The renderGraph in which we implemented our algorithm is the PathTracer renderGraph.

The PathTracer renderGraph is capable of running a simple unbiased path tracer great for creating ground truth images. It also already includes some NEE algorithms such as the single-level BVH in which our implementation is based, and ReStir [1] the current best algorithm for direct lighting.

3.2. Two-Level Light BVH

The two levels of the BVH divide the scene differently, the top-level acceleration structure(TLAS) divides the scene into groups of emissive meshes, each BLAS divides an emissive mesh into groups of emissive triangles. Each leaf node of the TLAS contains only one emissive mesh and a link to the corresponding bottom-level acceleration structure(BLAS) that will divide said emissive mesh, this structure is as seen in Figure 1.

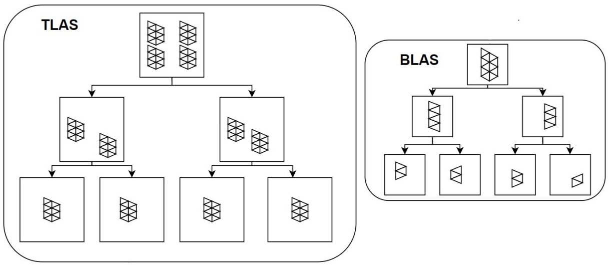


Figure 1: Two-level BVH structure. The TLAS keeps the meshes while the BLASes subdivide meshes.

By building the tree this way the refit operation only has to be applied to BLASes of changed mesh lights and the TLAS. When lights change TLAS and BLASes that need updating get refitted. The refitted TLAS is an asynchronously rebuilt TLAS from the previous frame.

Next, we describe each of the operations needed to use and maintain a two-level light BVH

3.3. BVH Building

Building the BVH is done in a top-down approach, in the case of a two-level BVH we start at the root of the TLAS, this node has an AABB that encompasses the entire scene and a list of all lights.

Then we split the current node being processed into two child nodes, the parent AABB and light list are divided between the two child nodes. Then we repeat this slitting for created child nodes until we only have one light, creating a TLAS leaf node.

The BLAS build is done in the same way as the TLAS, subdividing the corresponding emissive mesh into subsets of triangles.

To split nodes we use binning to divide the element list into several sets of elements(bins). Then we calculate the cost of splitting between each of those bins with SAH or SAOH and finally choose the lowest cost split.

To evaluate the cost of a split that divides the list of elements into two groups, we use one of two heuristics, these are SAH Eq.(2) which takes into consideration the number of triangles and their total area, and SAOH Eq.(3) which also considers the light's orientation yielding better picture quality.

$$cost_S AH(L, R) = \frac{n(L)a(L) + n(R)a(R)}{n(L \cup R)a(L \cup R)} \tag{2}$$

Where n(G) and a(G) refer respectively to the number of triangles in group G and their area.

$$cost_S AOH(L, R, s) = \\ k_r(s)\frac{\Phi(L)a(L)M_\Omega(L) + \Phi(R)a(R)M_\Omega(R)}{a(L \cup R)M_\Omega(L \cup R)} \tag{3}$$

Where $k_r(s)$ is a regularization factor given by $k_r(s) = \frac{length_max}{length_s}$ this factor is used to penalize the choosing thin boxes and $M_\Omega(L)$ is an orientation based scalar.

3.4. BVH Refitting

Refitting is a much simpler operation than building, it is done on the GPU, the refit is done bottom-up as seen in Figure 2, processing the lowest level of BLASes first, then we refit the TLAS also starting at the lowest level. This means a node is refitted based on its child nodes. This implies we need to first process leaf nodes and only then their parent nodes.

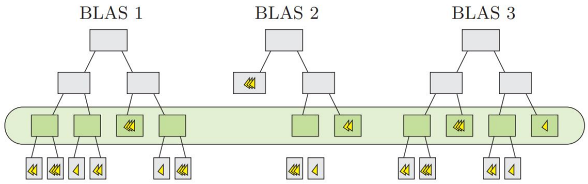


Figure 2: Figure form [5], showing all the BLASes being refit one dispatch per level with a bottom-up approach

To maximize refitting efficiency GPU batching is used, which means several GPU function calls are batched together to avoid delay overheads from multiple GPU calls.

To refit a BLAS leaf node we iterate over all of its triangles and calculate its new AABB and normal bounding cone. Internal nodes have their AABB and normal bounding cones adjusted to encompass their child nodes, and TLAS leaf nodes copy the corresponding BLAS root node properties.

3.5. BVH TLAS Rebuilding

The rebuilding of the TLAS happens in the same way as the building, but when a leaf node is created instead of building a BLAS it links to the correct already built BLAS.

Rebuilding happens asynchronously to minimize performance costs. With asynchronous processing comes a synchronisation problem.

The async rebuild is launched immediately after the TLAS refit, we allow syncing at the beginning of the next frame, and wait if needed for it to end before refitting in the next frame.

3.6. BVH Sampling

To generate a light sample from a two-level light BVH we traverse the BVH starting at the TLAS root until we reach a BLAS leaf.

To navigate the BVH (Bounding Volume Hierarchy), a random number is initially generated. This number determines which branch of the tree to explore. At each node, the random number is compared to the probability of selecting each branch.

The probability of choosing each node varies with their importance. This importance can be measured with the combination of the following metrics, distance, light flux, light orientation and light visibility.

Combining these metrics the following expression is used to calculate each node's importance:

$$importance(X, C) = \frac{\Phi(C)|\cos\theta_i'|}{||X - C||^2} x g(\theta') \tag{4}$$

Where X is the shading point and C is the center of the AABB, $g(\theta')$ is $cos\theta'$ if $\theta' < \theta_e$ and zero otherwise, $\theta_i' = max(0, \theta_i - \theta_u)$ and $\theta' = max(0, \theta - \theta_o - \theta_u)$where θ_i is the incident angle and θ_u is the uncertainty angle for the bounding cone to cover all emissive triangles, all these can be found in Figure 3.

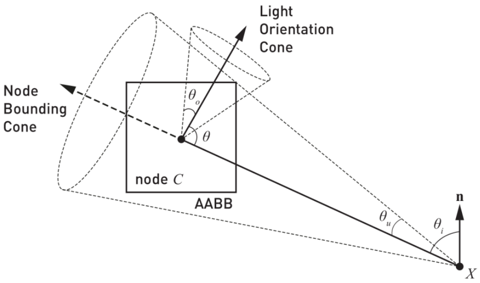


Figure 3: Image taken from[5].Geometry used for computing the importance of each node.

4. Experiments

To evaluate the performance of the developed algorithm we tested it against its single-level variant and ReStir. This way we can study the advantages of rebuilding the top structure as well as the consequences of restricting splitting to entire meshes in the top structure. We also compared the quality of the SAH and SAOH splitting heuristics with the two-level BVH.

The rendered images were tested against ground truth images we generated with Mogwais PathTracer.

We used two scenes to test the algorithms. The first, NVIDIA´s Bistro [4], features a long street allowing us to showcase quality decay over long stretches of movement. The other test scene was Emerald Square[6], it offers a more expansive environment filled with numerous lights, creating a complex lighting scenario for analysis.

The metrics taken in testing are MSE, SSIM, and PSNR. For performance we measured total frame time, as well as duration of rebuild and refit.

4.1. Results

In the Table 1 we have the averages over the several samples tested for each frame. Then in Figure 4 we show the graphs that map those averages shown in Table 1. We can see the better updating from the two-level BVH mitigates some of the quality loss.

When comparing the BVH´s we see the two-level BVH outperforming the single-level in MSE by 17% on average with an up to 36% improvement and in SSIM by 6% on average.

Bistro Exterior					
SingleLevel Average Frames:	0	50	100	150	Average
MSE	460,76	516,73	515,09	846,48	**584,76**
SSIM	0,525	0,500	0,509	0,462	**0,499**
PSNR	23,253	22,628	22,800	21,263	**22,486**
TwoLevel SAOH Average Frames:					
MSE	426,26	440,77	438,56	688,05	**498,41**
SSIM	0,542	0,538	0,538	0,493	**0,53**
PSNR	23,583	23,462	23,591	22,407	**23,26**

Table 1: Table with the average results for all metrics in tested frames of our version of Bistro.

The average SSIM results obtained in Emerald Square (Figure 5) show minimal difference between both BVH´s, the SAOH outperforms SAH heuristic. The ReStir results were considerably better than the other algorithms with

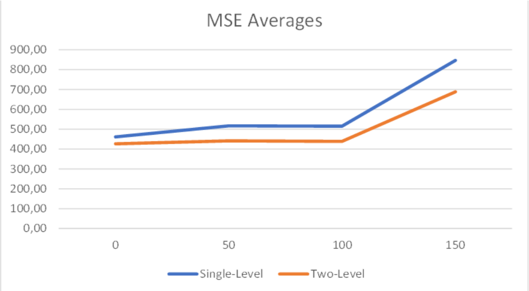


Figure 4: Graphs mapping the averages of MSE from Table 1.

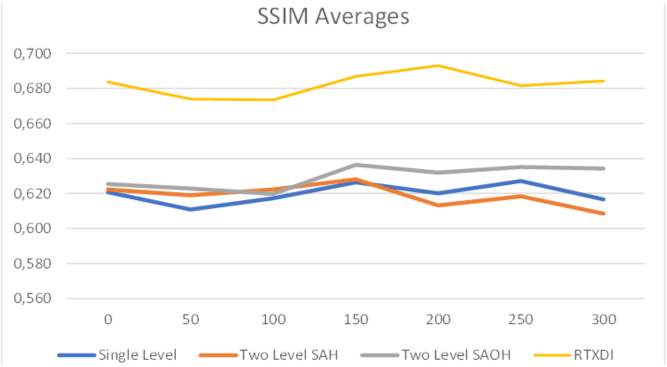


Figure 5: Graphs mapping the averages of SSIM in Emerald Square[6].

a 60% improvement in MSE and 8% in the SSIM metric. When looking at the close-up images of Emerald Square provided in Figure 6, we see much better results for the two-level SAH and SAOH pictures than for the single-level image.

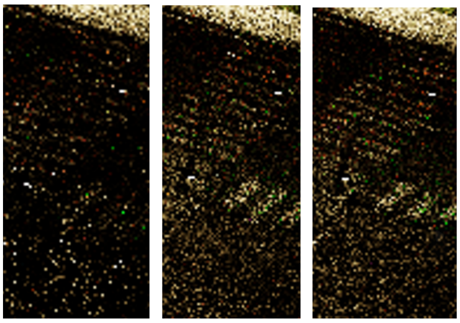


Figure 6: Zoomed in pictures of frame 150 with 4 samples. Left-Light BVH SAOH, center -two-level with SAH, right - two-level BVH with SAOH.

In Table 2 we have the frame time results for the single-level BVH and two-level BVH both using SAOH. These results were obtained in a PC with an Intel i5-4690k at stock frequency and an NVIDIA RTX 2080. The results are averages over all frames of animation with a simulated 25fps.

Overall the studied metrics showed the expected improvements and were close to what was reported by the work described in [5]. We can see from the close picture that the two-level BVH is working as expected improving quality in areas affected by moving lights.

Performance was as expected we saw an average of 7% slow down which is consistent with what was reported in [5]

	Measure One		Emerald Square		Bistro Exterior	
Algorithm:	Single Level BVH	Two Level BVH	Single Level BVH	Two Level BVH	Single Level BVH	Two Level BVH
BVH Refit (ms)	4	7	7	9,2	4,2	6,3
BVH Rebuild (ms)	- -	7,2	- -	12,5	- -	3,2
Frame Time (ms)	61(16,4FPS)	66(15,2FPS)	60(16,7FPS)	63(15.9FPS)	36(27,7FPS)	38(26,3FPS)

Table 2: Performance Results

5. Conclusions

Not all scenes showed improved metrics metrics, however, when examining picture quality from zoomed-in pictures we saw more detail in certain areas with the two-level BVH compared to the single-level BVH.

Performance was as expected the added processing from the TLAS rebuild had little to no effect on performance due to its asynchronous processing but the refit was slightly slower.

When testing ReSTIR we saw an overall improvement in quality as was expected from the current best solution.

ReSTIR is the best solution of the three tested for all scenarios.

The two-level BVH showed some improvements over the single-level version, especially evident in scenarios where lights had undergone substantial updates.

Acknowledgements

The work reported in this article was partially supported under the auspices of the UNESCO Chair on AI & VR by national funds through Fundação para a Ciência e a Tecnologia with references DOI:10.54499/ UIDB/-50021/2020, DOI:10.54499/DL57/2016/CP1368/CT0002 and 2022.09212.PTDC (XAVIER project).

The work was also carried on under DL 60/2018, 3-08 – I&D activity at INESC -ID, project PEST 2020 -GI (graphics & interaction (UIDB/50021/2020).

References

[1] B. Bitterli, C. Wyman, M. Pharr, P. Shirley, A. Lefohn, and W. Jarosz. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (TOG)*, 39(4):148–1, 2020.

[2] A. Conty Estevez and C. Kulla. Importance sampling of many lights with adaptive tree splitting. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(2):1–17, 2018.

[3] J. T. Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986.

[4] A. Lumberyard. Amazon lumberyard bistro, open research content archive (orca), July 2017. `http://developer.nvidia.com/orca/amazon-lumberyard-bistro`.

[5] P. Moreau, M. Pharr, and P. Clarberg. Dynamic many-light sampling for real-time ray tracing. In *High Performance Graphics (Short Papers)*, pages 21–26, 2019.

[6] K. A. Nicholas Hull and N. Benty. Nvidia emerald square, open research content archive (orca), July 2017. `http://developer.nvidia.com/orca/nvidia-emerald-square`.

[7] M. Pharr, W. Jakob, and G. Humphreys. *Physically based rendering: From theory to implementation.* MIT Press, 2023.

[8] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, June 1980.

A novel silhouette extraction method for binary images based on the Wall-Follower algorithm

Mario A. Aguirre-López
Autonomous University of Chiapas, Faculty of Sciences in Physics and Mathematics
Carretera Emiliano Zapata km 8.0
29050, Tuxtla Gutiérrez, Mexico
marioal1906@gmail.com

Omar Efraín Izaguirre-Prado
Autonomous University of Nuevo León, Faculty of Earth Sciences
Carretera a Cerro Prieto km 8.0
67700, Linares, Mexico
omar.izaguirrepr@uanl.edu.mx

Roberto Soto-Villalobos
Autonomous University of Nuevo León, Faculty of Earth Sciences
Carretera a Cerro Prieto km 8.0
67700, Linares, Mexico
roberto.sotovll@uanl.edu.mx

Filiberto Hueyotl-Zahuantitla*
Cátedra CONAHCyT-UNACH
Carretera Emiliano Zapata km 8.0
29050, Tuxtla Gutiérrez, Mexico
fhueyotl@conahcyt.mx
*Corresponding author

ABSTRACT

Silhouette extraction involves separating objects of interest from their background, which has several applications in image processing. Among the silhouette extraction techniques, contour tracing is commonly applied to images with a uniform background. This paper introduces a novel contribution to contour tracing techniques, utilizing the Wall-Follower Algorithm (WFA) to extract silhouettes with uniform backgrounds, or binary images. The algorithm is based on the analogy of a follower sequentially walking aside the external boundary of a wall, without separating a hand from it; then, the follower walks tagging silhouette pixels along the way until returning to the initial position and direction. Experimentation on vehicle technical drawings, satellite views of bodies of water and photographs of plants shows its effectiveness in producing high-quality silhouettes while showing some advantages over existing techniques. They include quickness in obtaining a solution, efficiency and ability to handle complex contours, and the option to simplify the results by reducing the percentage of saved points that trace the perimeter, based on object characteristics. The robustness of the algorithm suggests it as a promising alternative with diverse applications in image analysis, computer-aided design, and 3D object reconstruction, by extruding silhouettes, the latter being the main motivation for this contribution.

Keywords

contour tracing, silhouette extraction, wall-follower algorithm, image analysis, pixel following, technical drawing, CAD

1 INTRODUCTION

When processing an image to identify and object of interest from it, edge detection is a basic operation that uses the idea of detecting abrupt changes in image intensity. Classic methods to perform this task use first or second derivatives of the image. Examples of the first kind are the Canny, Roberts, Prewitt and Sobel algorithms which compute the magnitude and direction of the intensity changes at every pixel; while, the Marr-Hildreth and Harlick algorithms are examples of those that use second derivative, these include additional steps to reduce noise before detecting edges in the image. A good review of these methods and pseudo codes can be found in [SC15, ALM17]. Several of the above mentioned algorithms are implemented in well know software, such as the free-software GIMP (GNU Image Manipulation Program), Python and Matlab.

A step beyond edge detection is the silhouette extraction or contour tracing of a pattern (also called border tracing or boundary tracing). It is defined as the separation of the object of interest in the image (foreground) from the rest of them (background) [AYA+20]. Silhouette based methods have become an important tool in Computer Aided Design (CAD), for example, in object classification [MgS+14], technical illustrating [GSG+99, KBB17, PS23], and body feature extraction and recognition [PPJ14, LW11, CHK+06, DLR+24, DTGÇ06, ST02].

Contour tracing is a technique applied to digital images (representation as an array of $m \times n$ pixels), where each pixel having a certain value according to their intensity color. Contour tracing algorithms usually uses bi-level (binary) images, images that consists on pixels that can have only one of exactly two values: 1 when it is part of the pattern or 0 when it is part of the background. In order to extract the silhouette of a pattern, 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.

https://www.doi.org/10.24132/CSRN.3401.46

identification of the boundary pixels is not enough, it is required an ordered sequence of the boundary pixels (a set of connected pixels). Contour tracing is a major contributor to the efficiency of the feature extraction process of silhouettes, for example to implement Freeman's chain-coded Curves, [FD77, LW11]. The most common contour tracing algorithms are the Square Tracing [Pra01], Moore-Neighbor Tracing [SHB08], the Radial Sweep [RAB12], Theo Pavlids' algorithm [Pav82], Snakes algorithm [KWT88], Amoeba algorithm [IV00], Topological-hierarchical algorithms [KC14]; among others so called Fast Contour-Tracing algorithms [BD18, SCS$^+$16].

A more complex method is introduced by [BH98], who proposed a curve tracing method fitting into the first category of both classifications, [ZQ04] and [AYA$^+$20]. It consists of three steps: i) locate a point of the surface by means of a search along a line of sight; ii) to find the silhouette by using a gradient method to walk along it; iii) to trace the silhouette by numerically integrating the corresponding equations and closing the form of the shape. As punctuated by the authors, this method highlights by the simplicity in construction and quickness in performance, however, it also has some disadvantages, mainly when dealing with pointed or sharped shapes.

In this context, our work provides an alternative method based on the Wall-Follower Algorithm (WFA) to extract silhouettes with known or uniform background. Details of the foreground that our technique can deal with are fully described in Section 2, together with its limitations.

The WFA was originally designed as a maze solver method. Its name arises because it reflects a person who tries to get out of the maze following the strategy of not removing one hand from a continuous wall, so that person follows a "hand on wall rule". Then, the method is a very effective but slow solver for perfect mazes, i.e. for those that have only two external continuous walls defining the entry and exit of the maze [NZ20]. Through years, the efficiency of the WFA has maintained in exploration which, in turn, has lead to the existence of some improved variants to solve specific types of mazes, such as: [DR06], who implemented a repulsive potential field with constrained active contours to set up large autonomous robots for navigation into diverse tunnel geometries, solving the problem of steering in narrow curved spaces; then, [DSG10] optimized the WFA for flood-mazes by sensing lateral walls to detect channels and avoid unnecessary fillers, so the greater the number of channels, the shorter the solving time; and more recently, [AAA$^+$23] developed an algorithm based on a modified Right-Left-Front hand rule to minimize the distance path while addressing the characteristic infinite loop-back problem of WFAs, which occurs when searching the entire area is required.

Clearly, the characterization of the silhouette is a mandatory task in several areas. However, to our understanding, no version of the WFA has been adapted for silhouette extraction, which is the primary focus of this study. As suggested above, it is well known that reconstructing 3D shapes from 2D images is a basic research area in computer vision, see for example [BCL01, BJL03] and references therein. The motivation for developing our algorithm is then to create a tool to generate 2D silhouette data which could then be used to extrude 3D shapes. In this sense, we oriented this work to vehicle shapes, in which technical draws in gray scale are commonly used. The structure of the paper is as follows: Section 2 describes the details of our method and its implementation in the statistical software R; Section 3 shows some study cases and discuss them; Section 4 highlights the concluding remarks of our work.

2 METHODOLOGY

2.1 The Wall-Follower Algorithm

In general terms, the WFA we programmed consists of a main routine (Algorithm 1) that sequentially calls two functions with the objective of tracing the contour of the foreground, by moving the "wall follower" in a right-hand lining procedure from a starting point up to the wall follower returns to its initial position with its initial direction. The procedure requires a matrix R, in which the original image is stored; a matrix S, with the same dimensions of R which will serve to store the silhouette; and a 2D-vector T to save pixel coordinates of the silhouette, arranging them as they were stored. In the algorithm, xy_{Start} represent the initial position at a side of the foreground and d_{Start} refers to the direction, which is parallel to the silhouette.

Algorithm 1 Pseudo-code for the WFA

```
procedure CONTOUR_LINER($R$,$S$,$T$,$xy_{Start}$,$d_{Start}$)
    $xy = xy_{Start}$
    $d = d_{Start}$
    $finish = 0$
    $i = 0$
    while $finish = 0$ do
        Call MOVE_TAG
        Call TAG_TURN
        Update $finish$
return $S,T$
```

In the Algorithm 1, the function `MOVE_TAG` is in charge of moving forward one step or turning left when a wall is in front; while the function `TAG_TURN` aims to tagging one cell at right or turning right when that cell is not part of the wall. Figs. 1 and 2 help to schematize the cases faced by the above functions.

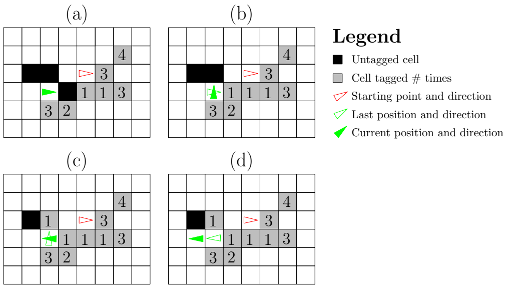

(a)
(b)
(c)
(d)
Legend
Untagged cell
Cell tagged # times
Starting point and direction
Last position and direction
Current position and direction

Figure 1: Visualization of the performance of the function `MOVE_TAG`. Cells in gray are tagged the number of times indicated inside them. Cells in black are part of the silhouette but they are still not tagged. (a) The follower in green is positioned after tagging by third time a cell by means of function `TAG_TURN`. (b) The cell in front is tagged by first time while the follower turns left. (c) The subsequent cell in front is tagged by first time while the follower turns left. (d) The follower can move forward.

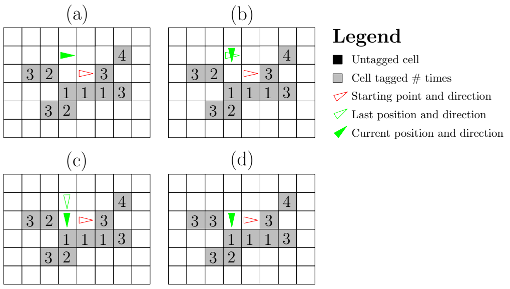

(a)
(b)
(c)
(d)
Legend
Untagged cell
Cell tagged # times
Starting point and direction
Last position and direction
Current position and direction

Figure 2: Visualization of the performance of the function `TAG_TURN`. Cells in gray are tagged the number of times indicated inside them. Cells in black are part of the silhouette but they are still not tagged. (a) The follower in green is positioned after tagging by second time a cell and moving forward by means of function `MOVE_TAG`. (b) The follower turns right since the cell in right is not part of the silhouette. (c) As a part of the same conditional, the follower moves forward. (d) The cell in right is sensed and tagged since it is part of the silhouette.

In detail, the first function to be called is shown in Algorithm 2, which is in charge to move the follower one cell in straight direction (L8), but considering that there could be scenarios where a wall is located in front of the follower, as schematized in Fig. 1. For that reason, the follower is able to turn left (or rotate in relative left direction) up to avoid walls in front (L3). The reader should be noticed that a wall implies that the associated cell is, indeed, part of the silhouette due to the assumption that we are drawing the outside contour. Thus, L5 tags that cell as a part of the silhouette by calling an auxiliary function shown in Algorithm 3. The reader could also note in Fig. 1 that up to two tags of different cells and one movement forward could be performed with calling this first function only once.

The auxiliary functions in Algorithm 3 (L2 and L4) tags and identifies if the follower has returned to the initial directional position (L5), modifying the index *finish* to 1 (L6) in such a case, in order to break the main loop in Algorithm 5.

Algorithm 2 Pseudo-code for the first function

```
1: function MOVE_TAG($R, S, i, T, xy, d, xy_{Start}, d_{Start}$)
2:     Update the cell in front $xyd$
3:     while $R[xyd]$ is part of the silhouette do        ▷ If not possible to go straight
4:         $xy^* = xyd$; $d^* = d_{Left}$
5:         Call TAG_SIMPLE                                 ▷ Tag that cell
6:         $d = d_{Left}$                                  ▷ Turn left
7:         Update the new cell in front $xyd$
8:     $xy = xyd$                                          ▷ Move on one cell
   return $S, T, xy, d, i, finish$
```

Algorithm 3 Pseudo-code for the auxiliary function

```
1: function TAG_SIMPLE($S, i, T, xy^*, d^*, xy_{Start}, d_{Start}$)   ▷ tags the $xy^*$-cell
2:     $S[xy^*] = S[xy^*] + 1$        ▷ by adding 1 to that cell,
3:     $i = i + 1$
4:     $T[i] = xy^*$                  ▷ by adding the 2D-coordinates
5:     if $xy^* = xy_{Start}$ & $d^* = d_{Start}$ then
6:         $finish = 1$
   return $S, T, i, finish$
```

Once the first function in Algorithm 1 has moved the follower to a new cell, the second function (Algorithm 4) looks for tagging the cell located at right-hand of the current xy-cell, considering the updated d-direction. This is only applied when the cell is part of the silhouette; in such a case a tag is produced (L6-L9) and the function ends. However, if the right-hand cell can not be tagged because the follower completed a past tag and forward movement without changing direction, i.e., it was at an external border of the silhouette such as shown in Fig. 2; then the follower needs to turn (or rotate to) right, while advancing one step without breaking the conditional loop (L10-L12). Therefore, this process is repeated until a cell is tagged (L3).

A special remark on this construction is that, since tagging is performed by adding one unit to the considered cell, a cell could be tagged more than once when the follower is at an external border, so that the value of each cell corresponds to the number of faces belonging to the outside contour of the foreground, up to four in 8-connected patterns. Indeed, in every step, the follower sens the cell at right of it without coming into the foreground then, it is feasible to trace the contour of any type of 8-connected patterns (foregrounds with an external silhouette whose cell components either share an edge or a vertex), including 4-connected (those that only share edges). On the other hand, the user has two ways of exporting the extracted silhouette. A matrix representation (S) that identifies the number outside

Algorithm 4 Pseudo-code for the second function

```
function TAG_TURN($R, S, i, T, xy, d, xy_{Start}, d_{Start}$)
    $t = 0$                                   ▷ auxiliary variable
    while $t = 0$ do      ▷ If cell in right direction is not tagged
        Update $d_{Right}$
        Sens that cell $xyd_{Right}$
        if $R[xyd_{Right}]$ is part of the silhouette then
            $xy^* = xyd_{Right}$; $d^* = d$
            Call TAG_SIMPLE                   ▷ Tag that cell
            $t = 1$                           ▷ And break the loop
        else
            $d = d_{Right}$  ▷ Turn right to look for a cell that is part of the silhouette
            $xy = xyd_{Right}$                ▷ And go straight
return $S, T, xy, d, i, finish$
```

faces of the silhouette, and a 2D-vector (T) which is able to delimit the directional contour.

This construction, although with some limitations, has good performance and interesting features. Next, we briefly list limitations and characteristics. The last are described in depth in Section 3. The algorithm is limited to:

- Extract silhouettes consisting of simple closed curves, since a single crossover or overlap produces an early stop due to the L5 condition of the Algorithm 3;
- Cases with a simple background, or one that could be removed by an external filter.
- Silhouettes whose set of points are fully contained in the image, i.e., can not be part of the limits because the follower walks next to the outer face of the silhouette.

In turn, the characteristics are:

- Directional contour, the cells of the extracted silhouette could be saved/indexed in an orderly manner with respect to adjacent cells.
- Silhouette identifying borders of the object with a maximum of three faces to be tagged, the algorithm identifies how many faces of a cell belong to the outside contour of the object.

2.2 Implementation

The complete process proposed for the extraction of a silhouette consists of five parts and is structured as in Algorithm 5. First, an image needs to be read and saved in RGB or gray scales with decimal color code; such a matrix is referred as R. Then, appropriate filters should be applied to R in order to obtain an image fulfilling the restrictions mentioned in Section 2.1. The third step is to initialize the matrix S and the vector T, where the silhouette points and the directional contour will be saved in, respectively. Thus, the fourth step is to locate the starting point xy_{Start} and direction d_{Start} that would initialize the WFA; this could be done by any search along a line of sight until detecting the outside pixel of the silhouette since the algorithm performance is independent from its starting point. Finally the calling of WFA is carried out in the fifth step.

Algorithm 5 Main pseudo-code (integrator algorithm)

```
Read image and associate it to matrix $R$
Apply filters to $R$
Create a matrix $S$ for saving the silhouette, and save orderly the coordinates in a 2D-vector $T$
Initialize $xy_{Start}$, $d_{Start}$ for the WFA by a search along a line of sight
Call Algorithm 1
```

3 RESULTS

We tested our method with some images fulfilling the restrictions mentioned in Section 2.1. They consist of different views of technical draws of vehicles, which were freely obtained from Free Cad Blocks [Fre24], see Figs. 3-5. A pre-processing was applied to the images: imperfections such as those in the right wheel of the truck in Fig. 4 (b) and (c), and in the (disjointed) mirrors of the truck with platform in Fig. 5 (a).

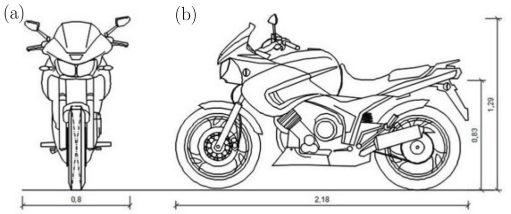


Figure 3: Front (a) and lateral (b) views of a motorcycle draw [Fre24].

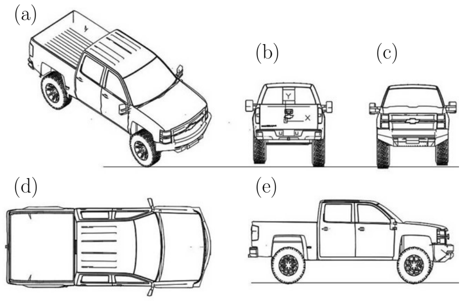


Figure 4: Perspective (a), rear (b), front (c), top (d), and lateral (e) views of a truck draw [Fre24].

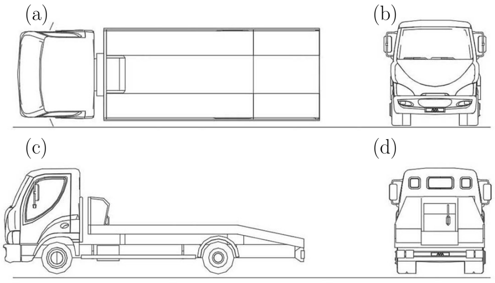


Figure 5: Top (a), front (b), lateral (c), and rear (d) views of a truck with platform draw [Fre24].

Following the process mentioned in Algorithm 5, images were read considering a gray scale from 0 (white) to 1 (black). A simple filter consisting of mapping all values < 0.8 to 0-value and the rest of them to 1 was applied. The search for xy_{Start} and d_{Start} was initialized by testing cells of the image in downward direction from the top-center; when an analyzed cell was found to belong to the silhouette, then the search stopped, as schematized in Fig. 6. Finally, the Algorithm 1 was called.

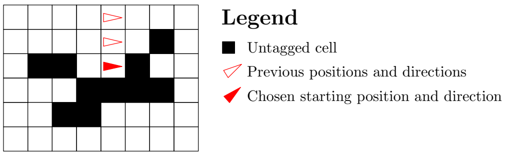

Legend
Untagged cell
Previous positions and directions
Chosen starting position and direction

Figure 6: Visualization scheme of the performance of searching for a starting point. The search starts from the columns closest to the top-center of the image.

Figs. 7-9 show the resulting silhouettes of some representative views: Fig. 7 (a) and (b) for cases (a) and (b) in Fig. 3; Fig. 8 (a) and (b) for cases (a) and (e) in Fig. 4, respectively; Fig. 9 (a) and (b) for the cases (a) and (c) in Fig. 5, respectively. Two silhouettes are plotting per graphic, indicating the number of points defining them. Continuous gray lines refer to the full contour obtained by applying our technique. Red dots refer to a simplified contour considering only silhouette cells that are tagged two or more times with our technique.

The full silhouettes provide high quality solutions, capturing the directional contour of the body in such a manner that the target object can be easily recognized. On the other hand, the simplified silhouettes reproduce a contour free of straight lines, achieving to reduce considerably the number of points, mainly when flat surfaces are present. However, some details of the original images could be lost when large flat zones are combined with right angles, like those ones at the connection of the truck with the platform in Fig. 9 (a) and (b). The resulting silhouette would not be reliable reconstruction of the vehicle.

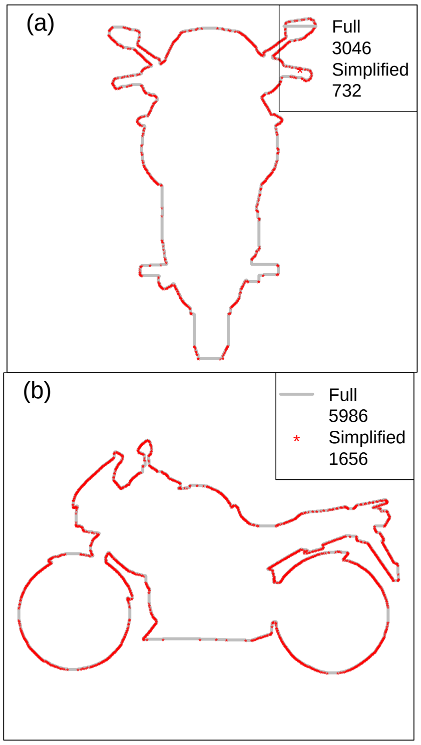


Figure 7: Full and simplified silhouette extraction of the motorcycle draws shown in Fig. 3. (a) Front view. (b) Lateral view.

Table 1 summarizes the percentages of points saved when utilizing the simplified approximation for all the analyzed cases. Since the structure of the truck cases is similar, full reconstructions of them are achieved with almost the same points, around 3000. The lateral view of the motorcycle requires about twice as many points. The behavior of the simplified reconstructions contrasts with this, depending mainly of the presence of (horizontal and vertical) flat zones, as suggested before. Such an example, the full reconstruction shown in Fig. 9 (a) is reduced in 95%, and the most of the points in the simplified reconstruction take place at the mirrors of the truck.

Thus, the reconstruction to consider for a specific application will depend on the flatness zones of the analyzed object and the type of edges at the borders of those zones, being useful a simplified reconstruction when there are cells exposing more than one face at the edges. In this context, external factors such as (original or added) noise at the edges could be helpful to avoid the coarseness problem. This, and other lines of research like the identification of shadows, the completeness of edge discontinuities and the extraction of a second internal silhouette are proposed as future extension of this work.

Now, beyond the high potential use of our algorithm for technical draws and meshing of vehicles, we applied the technique to other kind of images in order to discuss its feasibility in different scenarios. Fig. 10 shows two

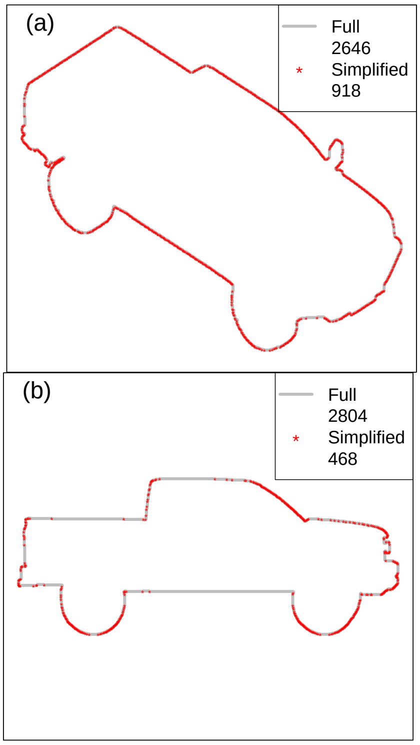


Figure 8: Full and simplified silhouette extraction of the truck draws shown in Fig. 4. (a) Perspective view. (b) Lateral view.

Object	View	Full (# points)	Simplified (%)
Moto	Front	3,046	24%
Moto	Lateral	5,986	28%
Truck1	Perspective	2,646	35%
Truck1	Lateral	2,804	17%
Truck2	Top	3,212	05%
Truck2	Lateral	2,886	12%

Table 1: Comparison of the number of points for each extracted silhouette. Simplified silhouettes are written as a percentage of their corresponding full versions. Truck1 corresponds to the single truck, while Truck2 to the truck with platform.

satellite photos of water bodies in Mexico. Their reconstructions are shown in Fig. 11. Even though the original images have not a uniform or known background, there is a high color contrast (for both images) between the soil and the water bodies, which allows a good performance of our algorithm, either through a full reconstruction or a simplified one (about 25% of the full one).

On the other hand, Fig. 12 shows two photos of plants with uniform background but many internal holes. As a consequence of this, only their external silhouettes are successfully extracted with details that increase the number of points (up to 28,000 for the full version of the palm case), but the internal details are lost, as occurred with all the contour tracing algorithms dealing with such a kind of images. The weight ratio between full and simplified reconstructions is also 20-25%.

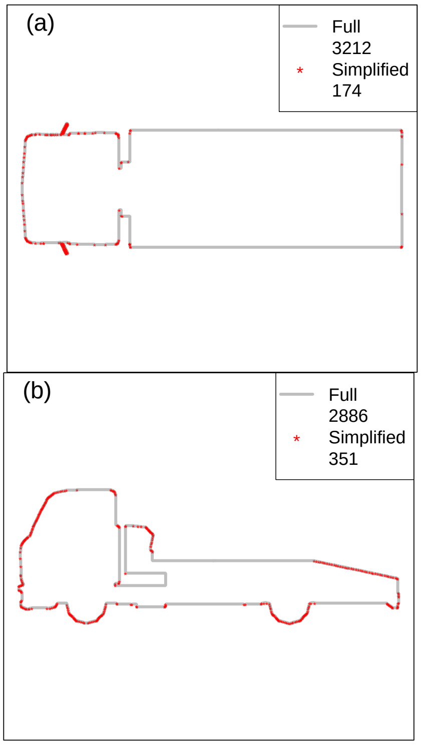


Figure 9: Full and simplified silhouette extraction of the truck draws shown in Fig. 5. (a) Top view. (b) Lateral view.

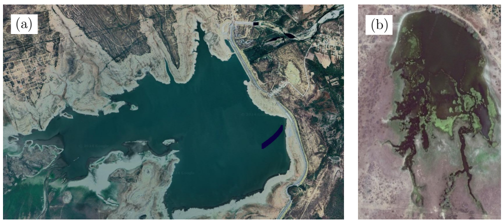


Figure 10: Satellite views of water bodies. (a) Dam Cerro Prieto in Linares, Nuevo León, Mexico. (b) Natural water body in China, Nuevo León, Mexico.

Beyond the characteristics of independence from the choice of the starting point and the directional contour, it is important to remark three advantages of our method over the reported techniques mentioned in Section 1:

- Quickness: taking into account that the Square Tracing algorithm is the one that requires the minimum of cell inspections among the existing techniques, while it is the most similar to ours, having as main difference that the WFA does not need to enter and leave the contour, since the follower always remain outside it.
- Complex contours: similarly to the Moore-Neighbor Tracing, Redial Sweep, Theo Pavlidi's,

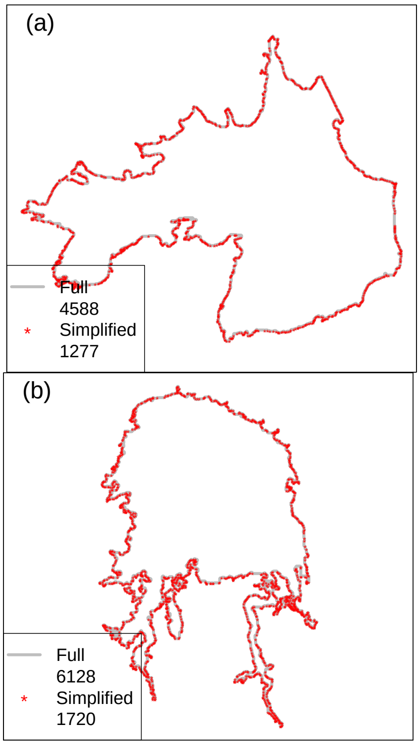


Figure 11: Full and simplified silhouette extraction of the two dams shown in Fig. 10.

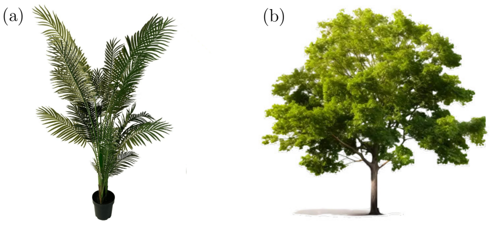


Figure 12: Photos of plants with uniform background. (a) A palm. (b) A tree.

and Fast Backward Contour Tracing algorithms, our implementation of the WFA is able to deal with 8-connected patterns that are not 4-connected, which is not possible for the Square Tracing algorithm.

- Simplified contour: our technique posses the duality of a) delimiting the complete contour, providing the 100% of the perimeter such as the Square Tracing algorithm; b) or well, it can simplify the contour, reducing the number of points that represent the perimeter by up to 83% for the analyzed cases, which possessed a complex geometry. This is a significant characteristic for saving computational storage when addressing high resolution images. In this sense, our algorithm can compete with traditional Freeman's chain codes (8- and 4-directional) [Fre61, FD77] and chain code compression schemes [ÅML15]. In turn, our results contrast with those of the experiments shown in [BD18] for the above mentioned algorithms, in which the Fast Backward Contour Tracing algorithm provided the most reduced contours, only up to 35%.

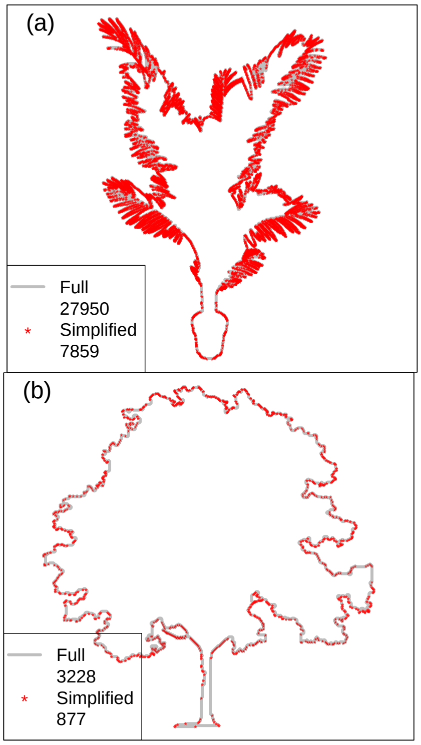


Figure 13: Full and simplified silhouette extraction of the plants shown in Fig. 12.

It is important to note that the simplified silhouette obtained as a vector of ordered points could be used as entry data for 3D automotive design in CAD software by extruding the silhouette data on each direction and then, intersecting the extruded objects, as shown in Fig. 14. This is a line of research that could be useful for novice designers and engineers looking for a simple and specialized method, emphasizing usability of the resulting designs from technical draws.

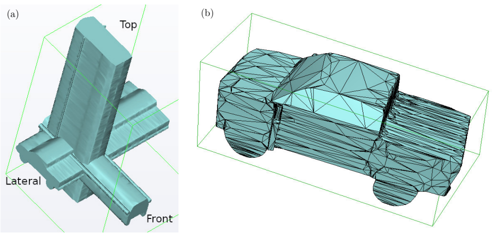


Figure 14: Application of ordered silhouettes in the 3D reconstruction of Truck1 in FreeCAD software, with cases the lateral, front, upper views shown in Fig. 4. (a) Extrusion of the silhouettes. (b) Final object.

4 CONCLUSIONS

We developed a technique to extract an external silhouette from a binary image, based on the Wall-Follower algorithm. The method is robust under considerable entry data conditions and presents some advantages over existing similar techniques, such as: independence of the starting point, quickness, reliability when dealing with complex and spiked contours, and the possibility of minimizing the number of points to trace the final contour. The property of the algorithm to store the complete contour and at the same time the simplified silhouette (which can be considered as a compressed form), as a matrix and as ordered vector elements, enhances its application in various areas that require the use of silhouettes.

5 FUNDING

This research was funded by Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCyT), grant number 839412.

6 ACKNOWLEDGMENTS

The third author thanks FCFM-UNACH and the support from CONAHCyT through the program “Investigadoras e investigadores por México”, Cátedra 873.

7 REFERENCES

[AAA+23] Shatha Alamri, Hadeel Alamri, Wejdan Alshehri, Shuruq Alshehri, Ahad Alaklabi, and Tareq Alhmiedat. An autonomous maze-solving robotic system based on an enhanced wall-follower approach. Machines, 11(2), 2023.

[ALM17] Naurin Afrin, Wei Lai, and Nabeel Mohammed. Performance analysis of corner detection algorithms based on edge detectors. Computer Science Research Notes - CSRN, CSRN 2701:21–28, May 2017.

[AYA+20] Guido Ascenso, Moi Hoon Yap, Thomas Allen, Simon S. Choppin, and Carl Payton. A review of silhouette extraction algorithms for use within visual hull pipelines. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 8(6):649–670, 2020.

[BCL01] Andrea Bottino, L. Cavallero, and Aldo Laurentini. Interactive reconstruction of 3-d objects from silhouettes. In Journal of WSCG, Vol.09, No. 1-3, pages 230–236, 01 2001.

[BD18] Yuriy Batko and Vitalii Dyminskyi. Fast contour tracing algorithm based on a backward contour tracing method. In Mykola Dyvak and Libor DostÃ¡lek, editors, Proceedings of the International Conference Advanced Computer Information Technologies, ACIT 2018, Ceske Budejovice, Czech Republic, June 1-3, 2018, volume 2300 of CEUR Workshop Proceedings, pages 219–222. CEUR-WS.org, 2018.

[BH98] David Bremer and John F. Hughes. Rapid approximate silhouette rendering of implicit surfaces. In Proceesings of Implicit Surfaces 98, 1998.

[BJL03] Andrea Bottino, Luc Jaulin, and Aldo Laurentini. Finding feasible parameter sets for shape from silhouettes with unknown position of the viepoints. In Proceedings of the The 11-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision’2003, WSCG 2003, Plzen, Czech Republic, Februay 7-13, 2003.

[CHK+06] Xi Chen, Zhihai He, J.M. Keller, D. Anderson, and M. Skubic. Adaptive silhouette extraction in dynamic environments using fuzzy logic. In 2006 IEEE International Conference on Fuzzy Systems, pages 236–243, 2006.

[DLR+24] Arindam Dutta, Rohit Lal, Dripta S. Raychaudhuri, Calvin-Khang Ta, and Amit K. Roy-Chowdhury. Poise: Pose guided human silhouette extraction under occlusions. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 6153–6163, January 2024.

[DR06] Elliot S. Duff and Jonathan M. Roberts. Wall Following with Constrained Active Contours, pages 51–60. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[DSG10] Hongshe Dang, Jinguo Song, and Qin Guo. An efficient algorithm for robot maze-solving. In 2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics, volume 2, pages 79–82, 2010.

[DTGÇ06] Yiğithan Dedeoğlu, B. Uğur Töreyin, Uğur Güdükbay, and A. Enis Çetin. Silhouette-based method for object classification and human action recognition in video. In Thomas S. Huang, Nicu Sebe, Michael S. Lew, Vladimir Pavlović, Mathias Kölsch, Aphrodite Galata, and Branislav Kisačanin, editors, Computer Vision in Human-Computer Interaction, pages 64–77, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[FD77] Freeman and Davis. A corner-finding algorithm for chain-coded curves. IEEE Transactions on Computers, C-26(3):297–303, 1977.

[Fre61] Herbert Freeman. On the encoding of arbitrary geometric configurations. IRE Transactions on Electronic Computers, EC-10(2):260–268, 1961.

[Fre24] Free CAD Blocks. CAD library. `https://freecadblock.com/`, 2024. Online; accessed March 1, 2024.

[GSG+99] Bruce Gooch, Peter-Pike J Sloan, Amy Gooch, Peter Shirley, and Richard Riesenfeld. Interactive technical illustration. In Proceedings of the 1999 symposium on Interactive 3D graphics, pages 31–38, 1999.

[IV00] G. Iannizzotto and L. Vita. Fast and accurate edge-based segmentation with no contour smoothing in 2-d real images. IEEE Transactions on Image Processing, 9(7):1232–1237, 2000.

[KBB17] Hakki Can Karaimer, Ipek Baris, and Yalin Bastanlar. Detection and classification of vehicles from omnidirectional videos using multiple silhouettes. 20(3):893–905, 2017.

[KC14] Pisut Koomsap and Natthavika Chansri. Topological hierarchy-contour tracing algorithm for nests of interconnected contours. The International Journal of Advanced Manufacturing Technology, 70:1247–1266, 02 2014.

[KWT88] Michael Kass, Andrew P. Witkin, and Demetri Terzopoulos. Snakes: Active contour models. Int. J. Comput. Vis., 1(4):321–331, 1988.

[LW11] Yueh-Ling Lin and Mao-Jiun J. Wang. Automated body feature extraction from 2d images. Expert Systems with Applications, 38(3):2585–2591, 2011.

[MgS+14] Begüm Mutlu, Murat Haciömeroğlu, Guzel Mehmet Serdar, Mehmet Dikmen, and Hayri Sever. Silhouette extraction from street view images. International Journal of Advanced Robotic Systems, 11(7):114, 2014.

[NZ20] R. Niemczyk and Stanisław Zawiślak. Review of maze solving algorithms for 2d maze and their visualisation. In Stanisław Zawiślak and Jacek Rysiński, editors, Engineer of the XXI Century, pages 239–252, Cham, 2020. Springer International Publishing.

[Pav82] T. Pavlidis. ALGORITHMS FOR GRAPHICS AND IMAGE PROCESSING. Computer Science Press, 1982.

[PPJ14] R. Pulido, F. Paulano, and J.J. Jiménez. Reconstruction & interaction with 3d simplified bone models. In Proceedings of the 22nd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2014, 2014.

[Pra01] W.K. Pratt. Digital Image Processing: PIKS Inside. Wiley, 2001.

[PS23] A. Padinjarathala and R. Sadleir. Using the Adaptive HistoPyramid to Enhance Performance of Surface Extraction in 3D Medical Image Visualisation. Computer Science Research Notes, 3301:331–339, 2023.

[RAB12] Pratiksha R. Reddy, Vadipina Amarnadh, and Mekala Bhaskar. Evaluation of stopping criterion in contour tracing algorithms. 2012.

[SC15] Haldo Spontón and Juan Cardelino. A Review of Classic Edge Detectors. Image Processing On Line, 5:90–123, 2015. `https://doi.org/10.5201/ipol.2015.35`.

[SCS+16] Jonghoon Seo, Seungho Chae, Jinwook Shim, Dongchul Kim, Cheolho Cheong, and Tack-Don Han. Fast contour-tracing algorithm based on a pixel-following method for image sensors. Sensors, 16(3), 2016.

[SHB08] Milan Sonka, Vaclav Hlavac, and Roger Boyle. Image processing, analysis and and machine vision (3. ed.). 01 2008.

[ST02] Cristian Sminchisescu and Alexandru Telea. Human pose estimation from silhouettes - a consistent approach using distance level sets. In Journal of WSCG, Vol.11, No. 1-3, pages 413–420, 02 2002.

[ZQ04] Gang Zeng and Long Quan. Silhouette extraction from multiple images of an unknown background. In Proceedings of the Asian Conference of Computer Vision, Citeseer. Citeseer, 2004.

[ÅML15] Borut Å½alik, Domen Mongus, and Niko Lukač. A universal chain code compression method. Journal of Visual Communication and Image Representation, 29:8–15, 2015.