ISSN 1213-6972 *Volume 31, Number 1-2, 2023*

Journal of WSCG

An international journal of algorithms, data structures and techniques for computer graphics and visualization, surface meshing and modeling, global illumination, computer vision, image processing and pattern recognition, computational geometry, visual human interaction and virtual reality, animation, multimedia systems and applications in parallel, distributed and mobile environment.

Vaclav Skala – Union Agency

Journal of WSCG

Editor-in-Chief: Vaclav Skala
c/o University of West Bohemia
Faculty of Applied Sciences
Univerzitni 8
CZ 306 14 Plzen
Czech Republic
http://www.VaclavSkala.eu

Managing Editor: Vaclav Skala

Printed and Published by:
Vaclav Skala - Union Agency
Na Mazinach 9
CZ 322 00 Plzen
Czech Republic

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

Hardcopy: ***ISSN 1213 – 6972***
CD ROM: ***ISSN 1213 – 6980***
On-line: ***ISSN 1213 – 6964***

ISSN 1213-6972 *Volume 31, Number 1-2, 2023*

Journal of WSCG

An international journal of algorithms, data structures and techniques for computer graphics and visualization, surface meshing and modeling, global illumination, computer vision, image processing and pattern recognition, computational geometry, visual human interaction and virtual reality, animation, multimedia systems and applications in parallel, distributed and mobile environment.

Vaclav Skala – Union Agency

Journal of WSCG

Editor-in-Chief: Vaclav Skala
c/o University of West Bohemia
Faculty of Applied Sciences
Univerzitni 8
CZ 306 14 Plzen
Czech Republic
http://www.VaclavSkala.eu

Managing Editor: Vaclav Skala

Printed and Published by:
Vaclav Skala - Union Agency
Na Mazinach 9
CZ 322 00 Plzen
Czech Republic

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

Hardcopy: ***ISSN 1213 – 6972***
CD ROM: ***ISSN 1213 – 6980***
On-line: ***ISSN 1213 – 6964***

Journal of WSCG

Editor-in-Chief

Vaclav Skala

c/o University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering
Univerzitni 8, CZ 306 14 Plzen, Czech Republic
http://www.VaclavSkala.eu

Journal of WSCG URLs: http://www.wscg.eu or http://wscg.zcu.cz/jwscg

Editorial Board

Journal of WSCG

Board of Reviewers

2023

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

Journal of WSCG

Vol.31, No.1-2, 2023

Contents

ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.31, No-1-2, 2023

Automatic Individual Identification of Patterned Solitary Species Based on Unlabeled Video Data

Vanessa Suessle[1], Mimi Arandjelovic[2,3], Ammie K. Kalan[4], Anthony Agbor[2], Christophe Boesch[2], Gregory Brazzola[2], Tobias Deschner[5], Paula Dieguez[3], Anne-Céline Granjon[2], Hjalmar Kuehl[3,6,7], Anja Landsmann[8], Juan Lapuente[2], Nuria Maldonado[2], Amelia Meier[2], Zuzana Rockaiova[8], Erin G. Wessling[9,10], Roman M. Wittig[11,12], Colleen T. Downs[13], Andreas Weinmann[14], Elke Hergenroether[1]

1. Department of Computer Science, University of Applied Sciences Darmstadt, Darmstadt, Germany
2. Max Planck Institute for Evolutionary Anthropology (MPI EVAN), Leipzig, Germany
3. German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig, Leipzig, Germany
4. Department of Anthropology, University of Victoria, Victoria, Canada
5. Institute of Cognitive Science, University of Osnabrück, Osnabrück, Germany
6. Senckenberg Museum of Natural History Goerlitz, Goerlitz, Germany
7. International Institute Zittau, Technische Universität Dresden, Zittau, Germany
8. Zooniverse Citizen Scientist, c/o Max Plank Institute for Evolutionary Anthropology, Leipzig, Germany
9. Department of Human Evolutionary Biology, Harvard University, Cambridge, Massachusetts, USA
10. School of Psychology & Neuroscience, University of St. Andrews, St. Andrews, Scotland
11. Ape Social Mind Lab, Institute for Cognitive Sciences Marc Jeannerod, UMR 5229 CNRS / University of Lyon 1, Bron, France
12. Taï Chimpanzee Project, Centre Suisse de Recherches Scientifiques, Abidjan 01, Côte d'Ivoire
13. School of Life Sciences, University of KwaZulu-Natal, Scottsville, Pietermaritzburg, South Africa
14. Department of Mathematics, University of Applied Sciences Darmstadt, Darmstadt, Germany

ABSTRACT

The manual processing and analysis of videos from camera traps is time-consuming and includes several steps, ranging from the filtering of falsely triggered footage to identifying and re-identifying individuals. In this study, we developed a pipeline to automatically analyze videos from camera traps to identify individuals without requiring manual interaction. This pipeline applies to animal species with uniquely identifiable fur patterns and solitary behavior, such as leopards (*Panthera pardus*). We assumed that the same individual was seen throughout one triggered video sequence. With this assumption, multiple images could be assigned to an individual for the initial database filling without pre-labeling. The pipeline was based on well-established components from computer vision and deep learning, particularly convolutional neural networks (CNNs) and scale-invariant feature transform (SIFT) features. We augmented this basis by implementing additional components to substitute otherwise required human interactions. Based on the similarity between frames from the video material, clusters were formed that represented individuals bypassing the open set problem of the unknown total population. The pipeline was tested on a dataset of leopard videos collected by the Pan African Programme: The Cultured Chimpanzee (PanAf) and achieved a success rate of over 83% for correct matches between previously unknown individuals. The proposed pipeline can become a valuable tool for future conservation projects based on camera trap data, reducing the work of manual analysis for individual identification, when labeled data is unavailable.

Keywords

individual identification, SIFT algorithm, CNNs, automatic pipeline, pattern matching, open set problem, wildlife conservation, camera traps.

1. INTRODUCTION

With nearly 40,000 species classified as threatened by the IUCN and a general upward trend [1], efficient and reliable monitoring of wild animals in their natural habitats is essential for wildlife conservation. Monitoring is a complex and time-intensive task for ecologists and is a crucial step to answer hypotheses on the abundance, behavior, territory, social relationships and anthropogenic interaction. Conducting a population monitoring on a species gives scientists insights into the species' endangerment and helps to achieve conservation objectives to protect the population adequately and is an integral part of adaptive conservation cycles [2, 3]. Individual identification is a common method to estimate a population size [4]. Over recent years, camera traps have become an increasingly popular tool to monitor wildlife unobtrusively. The low acquisition and maintenance costs make camera traps an effective tool to collect large volumes of data without invading the habitat and disrupting the animal's natural behavior [5]. The affordability of camera traps generally results in an immense amount of collected videos and images. However, analyzing the enormous amount of data is time consuming, monotonous and exceeds the processing workload experts can manually accomplish in a short time [6].

Computer vision (CV) and artificial intelligence (AI) have the potential to automatize selected tasks and support ecologists in their work to identify individuals based on visual characteristics [4]. Convolutional neural networks (CNNs) have the power to learn features and quickly classify images. The drawback of current supervised classification methods is the relatively large amount of required labeled training data, which is not available in most cases for individual identification in wild environments [4]. The pipeline developed for this study, was composed of different components to automatize the manual steps typical of individual identification, combining deep learning and classical vision for feature detection as motivated in other studies [7]. The analytical steps included the detection and location of the animal, filtering of empty images and videos, extraction of meta information (e.g. from video files), detection and description of an individual's features, comparing the identified features among individuals and finally, the decision making about potential matches.

We demonstrated the usability of the pipeline with a dataset of leopard (*Panthera pardus*) videos collected with camera traps by the Pan African Programme: The Cultured Chimpanzee (PanAf) [8] (Figure 1). A leopard's coat pattern has the same characteristic as a human fingerprint. Both uniquely identify an individual [9]. We aimed to label and match individuals' appearances in the dataset and assign an ID for each individual, if the available data allowed. This task can be challenging because the data were

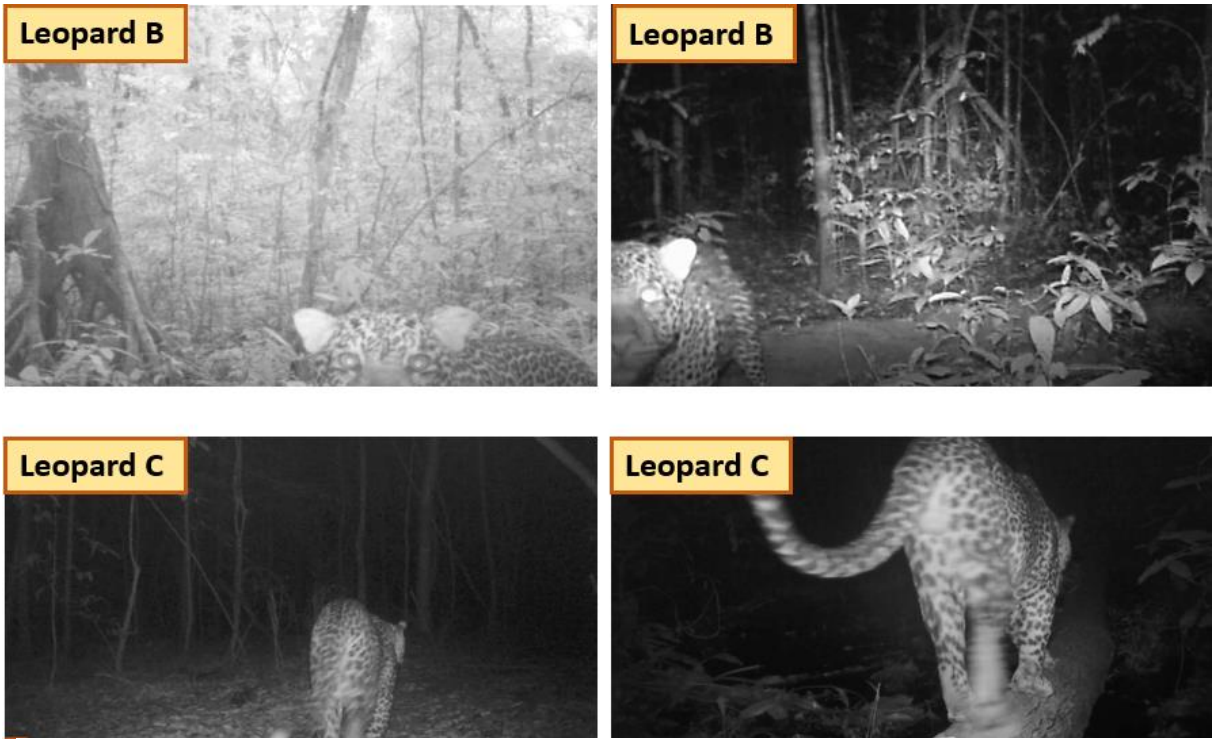


Figure 1. Matches found during analysis with pipeline

Correct matches of individuals from different videos of low illumination and quality, with only parts of the animals being visible. The individuals in the images on the left were matched to the individuals in the images on the right respectively.

Figure 2. Footage from camera traps
Captions by the PanAf with different quality, lighting, visibility and posture of the animal.

collected in the wild with varying conditions and differed in terms of lighting, quality, occlusions and included false triggers. Additionally, animals can be hidden, appear from various viewpoints or distances as well as in diverse poses (Figure 2). Leopards fulfill the requirement of being a solitary species [10] and thus an automatic labeling of the data with the presented pipeline is applicable. We therefore could reasonably assume that within one motion-triggered video, the same individual was seen throughout the frame sequence, which enabled the collection of different footage of the individual. A general drawback of CNNs for classification tasks is the open set problem [11, 12]. A traditional classifier can only re-identify and sort into a dictionary of known classes it was trained on. The classifier is compelled to pick the class that fits the most, even if none of the classes fit from a human perspective. For individual identification, this means that for an unknown individual, the classifier assigns it to one of the known individuals that fits the most. For population monitoring studies the total population is not known in advance and identifying unknown individuals is of high relevance. Our aim was to assist ecologists with a tool for individual identification of fur-patterned solitary species without requiring a large, labeled dataset of known individuals and the necessity of constant user inputs. We developed a modular pipeline that covers the subtasks of data preprocessing to individual identification.

2. RELATED WORK ON CAMERA TRAP DATA ANALYSES

Detection & Classification

Non-prefiltered datasets taken by motion-triggered cameras usually include a relatively large amount of falsely triggered images or videos, and footage on a spectrum of many species living in the ecosystem. An automated detector and classifier are essential for ecologists to process the automatically captured data, in a reasonable time frame [13]. For both filtering tasks, a type of classification is needed, either for the species or more generally separating into ‘non-empty’ and ‘empty’ classes. For the detection and localization of wildlife, which covers the task of filtering empty images, the MegaDetector [14] model is state-of-the-art. The trained CNN model returns bounding boxes around the detected animals. It was trained on many different datasets, including different species taken in diverse ecosystems. The model is constantly improved, and new versions are released regularly. On an ordinary GPU, the MegaDetector can process between 150,000 and 250,000 images per day [14].

While the MegaDetector is applicable to ecosystems around the globe, species classification approaches are usually tied to a specific ecosystem and its inhabitants. Pre-trained CNN models for species in North America [15, 16], Africa [16, 13, 17], Europe [18, 19] and Australia [16] are available, but they mostly cover focal species. Trained CNN models are also prone to the open set problem and can only classify the species they were trained on and are not sensitive to unknown species. Training such models require a large amount of labeled training data, and manual labeling is time-consuming [20]. For datasets with thousands or even millions of records, the labeling of the data may last multiple years [21]. To speed up this process, platforms were created to involve volunteers labeling the data.

Citizen Science

Volunteers who label data for projects are called citizen scientists [13, 22]. Platforms like Zooniverse [23], Wildlife Insights [24] and Wildbook [25] offer an option for research projects to open their data to citizen scientists who sort the images into predefined classes. With this approach, organizations can process the data relatively faster, and with the positive side effect of drawing the public’s attention to wildlife conservation. Besides the progress for the current case

study, labeling the datasets benefits the training of machine learning algorithms to support future wildlife conservation projects. The Snapshot Safari project [6, 26] is one of the world's largest camera trapping initiatives that used citizen scientists. From 2013 to 2020, over 138,000 volunteers from across the globe labeled more than nine million images. The drawback of this approach is that volunteers usually do not have many years of expertise and lack the knowledge to label rare species or individuals which can be challenging with camera trap images, even for experts [27]. The first attempts for online data processing with citizen scientists in near-real-time were conducted by a project in South Africa to fight wildlife poaching [28]. Captured images were immediately uploaded to a website. Volunteers examining the data can report a suspected poaching vehicle or human in the images and trigger a warning to local rangers in the nature reserve who thereby gain the opportunity to react quickly and prevent poaching activities.

Identification and Re-Identification of Individuals

While the automated classification of different species has been investigated over recent years, the field of identification of individual animals is still in its infancy. The identification and re-identification of individuals differ from the above-described task of species classification. For this task, not the species is recognized, but the unique individual. The identification of individuals with computer vision methods relies on visual biometric features that uniquely identify the individual. Since the biometric characteristics of different species vary, no overall solution covers all case studies. Early computer vision-assisted approaches for individual identification of marine mammals used unique body marks on the fins [29], or the fin's trailing edge was represented as integral curvatures [30]. The first ever automated estimation of a population was performed on African penguins (*Spheniscus demersus*) based on spot locations on their chest [31] compared against a database of known individuals. Nowadays, CNNs are a popular solution for individual identification and re-identification tasks. CNNs can extract features from animals with distinctive body marks. Previous studies applied CNNs to coat-, skin- or feather patterns of ringed seals (*Pusa hispida*) [32], whales [33], snow leopards (*Panthera uncia*) [34] and small birds [35]. In a study on the Great Barrier Reef, shell patterns of green turtles (*Chelonia mydas*) were extracted with a neural network system [36].

To the best of our knowledge, only supervised learning models have been used for individual identification of wildlife, treating each individual as a single class. Supervised, CNN-based solutions require large training datasets of labeled images, which are usually not available for wild animals, especially for automatically captured data.

A group from Shanghai Jiao Tong University, together with the World Wide Fund for Nature (WWF), generated and published a labeled dataset of 92 Amur tigers (*Panthera tigris altaica*) for training purposes. They trained an individual identification model on this dataset, for which each flank of a tiger was treated as an entity [37]. The patterned pelages on opposite flanks of felids are different and independent [38]. When the footage only shows opposite flanks of an animal separately in different captures, with no overlap of body parts seen, it is impossible to recognize whether the flanks belong to the same individual. Treating both flanks of the same individual as separate entities could lead to a biased estimation of the population size by a factor of 2. Further research tested Siamese convolutional neural networks with triplet loss [39], which are commonly used for person re-identification [40], for the re-identification of lions (*Panthera leo*), nyalas (*Tragelaphus angasii*) and ringed seals [41, 32]. But as with other deep learning approaches, labeled training data are required. Furthermore, CNN-based solutions bear the open set problem, which complicates identifying entities unknown to the population.

An alternative to CNN-based solutions is the pattern-matching scale-invariant feature transform (SIFT) algorithm [42]; with its scale, location, viewpoint and illumination invariant feature descriptor, it is well-suited for camera trap data [44]. Wild-ID [45] and HotSpotter [45, 25, 46, 47] are individual identification programs based on the SIFT algorithm for species with distinctive visual features. The SIFT approach is not applicable for species that lack unique fur or body markings.

The same concept used for human facial recognition can be applied to identify primates [48–50], pandas [51], bears [52] and pigs [53, 54]. For approaches concentrating on the face the collection of useable datasets in terms of quality, viewpoint and labeling is even more difficult than for fur-patterned species. Available datasets mostly stem from captive animals from zoos or farms.

The proposed solutions for individual identification described above all had at least one of the following: a labeled dataset, data collected under non-wild conditions, images manually photographed, closed populations where all individuals were known, or the solutions required human decision making or drawing bounding boxes.

In contrast, our study presents a pipeline that does not rely on labeled data or human interaction and covers the open set challenge. Our pipeline was tested with a dataset of videos automatically captured with camera traps in the wild. Thus, it covers the task of individual

identification and re-identification for an unlabeled dataset. Our pipeline benefits research by saving users valuable time estimating the number of individuals in a dataset.

3. PIPELINE FOR AUTOMATIC INDIVIDUAL IDENTIFICATION

Our objective was to provide researchers with a tool that automatizes individual identification from determining the animal's location in the video frames to feature extraction and matching. We develop a robust pipeline that unites the aforementioned analytical steps. The pipeline consists of newly developed components combined with existing components with proven functionality in prior case studies. Interim steps are implemented to substitute the otherwise required user input for specific components. The pipeline's main components cover the following tasks:

1. Image extraction: Extracting frames from video files and incorporating additional sequence-based information.
2. Object detection: Locating the animal within the image or classifying an image as empty.
3. Species classification: Selecting only images that include the species of interest.
4. Feature extraction: Detecting and describing features and measuring similarity to other images based on distance.
5. Clustering: Automatic matching of images to individuals based on their similarity.

The components of the pipeline are schematically outlined (Figure 4) and described in more detail below.

Image Extraction

The first component of the pipeline was the extraction of frames and additional information from video data compared with image data. We assumed that within one triggered video the same individual was seen throughout the frame sequence as leopards lead mainly solitary lives, and multiple images could be initially assigned to one individual ID in the database. Ideally, the animal moved during the video and images of various body poses from different viewpoints were obtained.

Object Detection

We embedded the above described MegaDetector [14] as an independent module to the pipeline for the task of object detection, which located the animal in the image and returned a bounding box. If no animal was found, the image was classified as empty.

Species Classification

Depending on the project/species of interest, a specific classification model must be chosen, e.g. the Zamba Cloud [17]. (For potential options, refer to the Related Work section). The present work focuses on individual identification, and the used dataset was already prefiltered for leopards by citizen scientists on the Zooniverse platform [23] the species classification component is therefore greyed out in Figure 4.

Feature Extraction

For feature detection and feature description, we employed the SIFT-based HotSpotter [45, 25, 46, 47]. SIFT-based algorithms do not require labeled training data. The HotSpotter outperformed its competitor Wild-ID in other studies [43]. The SIFT algorithm identified stable points in the image. It detected and described distinctive and characteristic features of the individual's fur-pattern (Figure 3) and turned them into feature vectors, which were mapped into a feature space. The feature vectors from frames from different videos were queried against other frames in the database, and a similarity score is calculated. The similarity score depended on the Euclidean distance of the mapped feature vectors in the vector space for each frame pairing. The analysis process required distinctive visual features and was only applicable to species with visually distinctive characteristics.

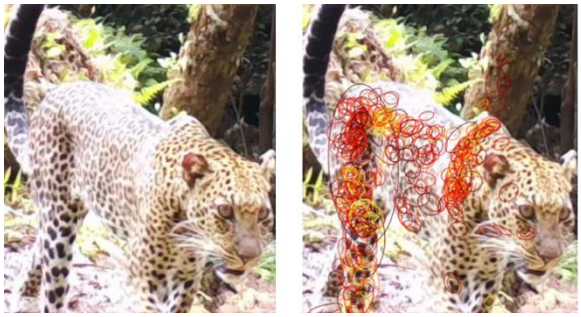


Figure 3. SIFT feature detection
Left: Raw frame from camera trap captured video. Right: Extracted SIFT features visualized with HotSpotter [45, 25, 46, 47].

Clustering

In the last step, the footage is assigned to respective individuals. This usually requires the user's decision and input. For our presented pipeline, the matching step was conducted automatically. We merged videos into clusters based on the user's predefined threshold for the similarity score. The clusters could be visualized in graphs, where nodes represented videos. Two nodes were connected with an edge, if frames from the videos matched. Each cluster represented one individual. We derived the width of the edges in the visualization (Figure 5) from the degree of similarity. A wide edge implied a high similarity between the animal shown in the frames of the videos. The distance between the nodes and clusters did not give

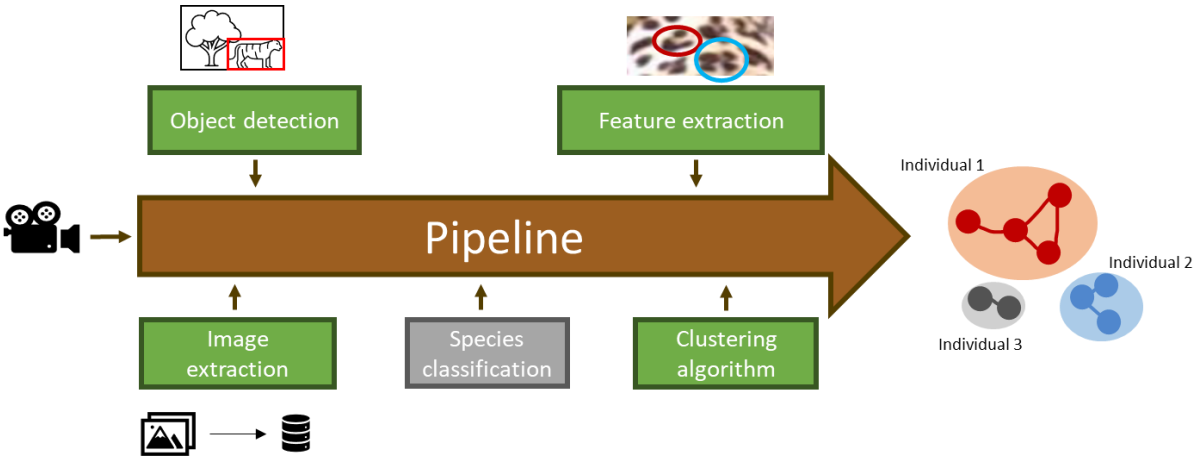


Figure 4. Schematic concept of the pipeline

Components of the pipeline include image extraction, object detection, species classification, feature extraction and matching, finalized by clustering the videos to represent individuals. The species classifier module is greyed out and was not used in this case study.

information on their similarity and were arranged to display a comprehensible representation. For each compared video pairing, three cases were possible:

A. Both nodes did not belong to a cluster yet.

B. One node was already part of a cluster, but the other one was not.

C. Both nodes already belonged to a cluster, but different clusters, causing a conflict.

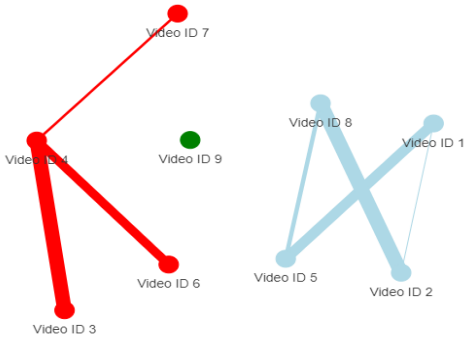


Figure 5. Schematic illustration of a cluster

Each node represents one video and each cluster an individual. The red graph showed that the animals seen in videos 3, 4, 6 and 7 were likely the same individual. Analogue for the blue graph. The animal in video 9 was not matched to any other animal in the video footage.

We handled the three cases as follows. In case A, a new cluster consisting of the two videos was created, while in case B, the free node was assigned to the existing cluster. Case C is the most complex of the cases. If the compared nodes were already assigned to different clusters, the nodes were rearranged and assigned to another cluster, so that the edges were based on the highest similarity scores. If the similarity of the new video pairing was higher than the similarity that binds the video into the present cluster, the video was released from the present cluster by deleting the edge and creating a new edge to the video with the higher similarity. Figure 5 illustrates a schematic example where the red subgraph shows that the animal seen in videos 3, 4, 6 and 7 were likely the same individual. The same accounts for the animal seen in the videos of the blue subgraph. While for video 9, printed in green, no match was found. Our pipeline outputted an HTML visualization of the clusters and a database comprising similarities and affiliations that ecologists could use for further observations.

The outlined components of the pipeline, including image extraction, object detection, species classification, feature extraction and clustering, are schematically outlined in Figure 4.

4. CASE STUDY

The data we used to demonstrate the pipeline in this study were provided by the PanAf [8]. The PanAf collected data at over 40 temporary and collaborative research sites with motion and infrared-equipped camera traps across Central and West Africa. Over 600,000 video clips were taken with a duration of one minute each. Forest habitats are complex areas to collect images and video data. Low light levels during the night further complicate the data collection and analysis, and the videos can be of low quality and only black and white. Snapshots taken from the videos of the animals can vary in distance and be blurred or relatively close up (Figure 1). The PanAf study was originally designed to capture data on chimpanzees and the camera locations were selected to suit their behavior, which makes the dataset especially challenging for other species than chimpanzees. To demonstrate the pipeline, a subset of the leopard dataset was used for which volunteers on Zooniverse labeled the individuals. The leopards' IDs were confirmed when citizen scientists that have been extensively involved and experienced in leopard identification unanimously agreed on the matching spot patterns after manual visual inspection [55]. The information on the individuals was only used for

validation purposes and not in the process itself. The leopard subset encompassed footage from 2011 to 2018 and totaled 210 videos from eight field sites representing 68 unique camera locations.

5. RESULTS

We demonstrated the pipeline for the individual identification and re-identification for an unlabeled dataset without manual interaction using part of the PanAf leopard dataset.

We automatically processed the 210 videos to validate our pipeline. A total of 116 matches were found, with 97 of those matches being correct, giving 83.6% success rate. The image in Figure 6 shows a correct match. In this example, the leopard's visible right hind limb had the most prominent features.

Even for complex footage at nighttime, with low quality and only parts of the animals captured, matching features could be extracted and matched (Figure 1). The most frequent reason for mismatches was the background for images taken at the same location since camera traps were fixed to a site and scenery (Figure 7). A comparison to other studies on individual identification is listed in Table 1. None of the existing approaches unites the ability to cope with data captured in the wild, unknown total populations, gain additional information from video format and at the same time does not require labeled data or manual inputs during the process.

Figure 6. Correct match of an individual

The top and bottom row show the same individual in different captures. Left: Raw images. Right: The same image, but with detected and matched features in the other image of the same individual.

Figure 7. Incorrect match of individuals

Reasons for the incorrect match are matched objects in the background of fixed camera sceneries.

Species	Method	Data captured in wild	No manual pre-processing	No labeled data required	Video	For unknown total populations	Top-1 accuracy
Manta rays / whales [56]	CNN	✓	✓	✓	✗	✗	64
Saimaa ringed seals [32]	Siamese network	✓	✗	✓	✗	✗	75
Jaguar/Ocelot [43]	HotSpotter / SIFT	✓	✓	(✓)*	✗	✗	77/76
Jaguar/Ocelot [43]	WildID / SIFT	✓	✓	(✓)*	✗	✗	68/63
Manta [57]	SIFT	✓	✓	✗	✗	✓	51
Amur tiger [37]	CNN	✗	✗	✓	✓	✗	89
The presented leopard pipeline	HotSpotter + pre-processing	✓	✓	✓	✓	✓	83

Table 1. Top-1 accuracy for individual identification programs

**Partly labeled data. New unknown images are mapped to a database of known individuals.*

6. DISCUSSION & OUTLOOK

In this study, we addressed the problem of animal re-identification from camera trap data. Our aim was to substitute manual user input and the need for labeled data. The core idea of the developed pipeline was to take advantage of video data and its consecutive frames for animals with solitary behavior. The pipeline's functionality was proven by identifying and re-identifying leopards from an unlabeled dataset collected by the PanAf.

For future work, a more detailed localization and extraction of the animal from the background rather than the current rectangular bounding boxes can address the challenge of matching the same objects in the background because of the fixed scenery in camera traps. The fixed scenery can also be used as an advantage for the extraction of the background. The detection of objects of interest may also be supported through the availability of video data by extracting optical flows [58]. CNN-based approaches for semantic segmentation could extract a mask for the animal, which excludes the background [59, 60].

For future studies, it may be interesting to collect additional information on the viewpoint and the visible flank of the animal for a better overview on known individuals. A process that automatically feeds a database with this information can further improve the monitoring of wildlife and prevent the incorrect matching of opposite sides of animals. With this information not only matching pairings can be detected, but also pairings that reliably show different individuals can be detected and marked.

Our pipeline will be used to support the PanAf identifying additional individuals in other regions. Another experiment for the future is to apply the pipeline to other species and valuate its suitability for cross-species applications.

7. ACKNOWLEDGMENTS

We thank the following members of the PanAf consortium and team for data or technical support with data processing: Karsten Dierks, Dervla Dowd, Henk Eshuis, Theo Freeman, John Hart, Thurston Cleveland Hicks, Jessica Junker, Vincent Lapeyre, Vera Leinert, Yasmin Moebius, Mizuki Murai, Emmanuelle Normand, Colleen Stephens and Virginie Vergnes. Further, we thank the following citizen scientists for their support: Tonnie Cummings, Carol Elkins, Lucia Hacker, Briana Harder, Karen Harvey, Laura K. Lynn, Heidi Pfund, Kristeena Sigler, Libby Smith, Jane Widness and Heike Wilken.

We thank the following government agencies for their support in conducting field research in their countries: Ministere des Eaux et Forets, Côte d'Ivoire; Ministère de l'Enseignement Supérieur et de la Recherche Scientifique, Côte d'Ivoire; Institut Congolais pour la Conservation de la Nature, DR-Congo; Ministere de la Recherche Scientifique, DR-Congo; Agence Nationale des Parcs Nationaux, Gabon; Centre National de la Recherche Scientifique (CENAREST), Gabon; Société Equatoriale d'Exploitation Forestière (SEEF), Gabon; Forestry Development Authority, Liberia; Direction des Eaux, Forêts et Chasses, Senegal. As well as the following NGOs in their countries: Taï Chimpanzee Project, Côte d'Ivoire; Wild Chimpanzee Foundation, Côte d'Ivoire, Liberia & Guinea; Lukuru Wildlife Research Foundation, DR-Congo; Loango Ape Project, Gabon; Fongoli Savanna Chimpanzee Project, Senegal.

The Pan African Program: The Cultured Chimpanzee is generously funded by the Max Planck Society, the Max Planck Society Innovation Fund, and the Heinz L. Krekeler Foundation.

8. REFERENCES

[1] "IUCN Red List Quadrennial Report 2017-2020," https://nc.iucnredlist.org/redlist/resources/files/1630480997-IUCN_RED_LIST_QUADRENNIAL_REPORT_2017-2020.pdf

[2] A. F. O'Connell, J. D. Nichols, K. U. Karanth, Eds. Camera traps in animal ecology: Methods and analyses. Springer, 2011.

[3] A. Caravaggi, P. B. Banks, A. C. Burton, C. M. V. Finlay, P. M. Haswell, M. W. Hayward, M. J. Rowcliffe, M. D. Wood, "A review of camera trapping for conservation behaviour research" Remote Sensing in Ecology and Conservation, vol. 3, no. 3, pp. 109–122, 2017, doi: 10.1002/rse2.48.

[4] S. Schneider, G. W. Taylor, S. Linquist, S. C. Kremer, "Past, present and future approaches using computer vision for animal re-identification from camera trap data" Methods in Ecology and Evolution, vol. 10, no. 4, pp. 461–470, 2019, doi: 10.1111/2041-210X.13133.

[5] P. Glover-Kapfer, O. R. Wearn, "Camera-Trapping WWF Guidelines" WWF Conservation Technology Series, 2017.

[6] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, C. Packer, "Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna" Scientific data, 2015, doi: 10.1038/sdata.2015.26.

[7] M. Stefańczyk, T. Bocheński, "Mixing deep learning with classical vision for object recognition" Journal of World Society for Computer Graphics (JWSCG), vol. 28, no. 1-2, pp. 147–154, 2020, doi: 10.24132/JWSCG.2020.28.18.

[8] Max Planck Institute for Evolutionary Anthropology. "Pan African Programme: The Cultured Chimpanzee | Where we work" http://panafrican.eva.mpg.de/english/where_we_work.php (accessed Nov. 22, 2022).

[9] P. Henschel, Ed., "Leopards in African Rainforests: Survey and Monitoring Techniques" Wildlife Conservation Society, 2003.

[10] T. N. Bailey, "The African Leopard: Ecology and Behavior of a Solitary Felid" Columbia University Press, 1993.

[11] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, T. E. Boult, "Toward Open Set Recognition" IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 7, pp. 1757–1772, 2013, doi: 10.1109/TPAMI.2012.256.

[12] P. Oza, V. M. Patel, "Deep CNN-based Multi-task Learning for Open-Set Recognition" 2019, early access: https://arxiv.org/pdf/1903.03161

[13] M. Willi, R. T. Pitman, A. W. Cardoso, C. Locke, A. Swanson, A. Boyer, M. Veldthuis, L. Fortson, "Identifying animal species in camera trap images using deep learning and citizen science" Methods in Ecology and Evolution, vol. 10, no. 1, pp. 80–91, 2019, doi: 10.1111/2041-210X.13099.

[14] S. Beery, D. Morris, S. Yang, "Efficient Pipeline for Camera Trap Image Review" Data Mining and AI for Conservation Workshop at KDD19, 2019, http:// arxiv.org/pdf/1907.06772

[15] M. A. Tabak, M. S. Norouzzadeh, D. W. Wolfson, S. J. Sweeney, K. C. Vercauteren, N. P. Snow, J. M. Halseth, P. A. Di Salvo, J. S. Lewis, M. D. White, B. Teton, J. C. Beasley, P. E. Schlichting, R. K. Boughton, B. Wight, E. S. Newkirk, J. S. Ivan, E. A. Odell, R. K. Brook, P. M. Lukacs, A. K. Moeller, E. G. Mandeville, J. Clune, R. S. Miller, "Machine learning to classify animal species in camera trap images: Applications in ecology" Methods in Ecology and Evolution, vol. 10, no. 4, pp. 585–590, 2019, doi: 10.1111/2041-210X.13120.

[16] G. Falzon, C. Lawson, K.-W. Cheung, K. Vernes, G. A. Ballard, P. J. S. Fleming, A. S. Glen, H. Milne, A. Mather-Zardain, P. D. Meek, "ClassifyMe: A Field-Scouting Software for the Identification of Wildlife in Camera Trap Images" Animals, vol. 10, no. 1, 2020, doi: 10.3390/ani10010058.

[17] "Zamba Cloud." https://www.zambacloud.com/#what-is-zamba-cloud (accessed Feb. 20, 2023).

[18] P. Follmann, B. Radig, "Detecting Animals in Infrared Images from Camera-Traps" Pattern Recognition and Image Analysis, vol. 28, no. 4, pp. 605–611, 2018, doi: 10.1134/S1054661818040107.

[19] C. Carl, F. Schönfeld, I. Profft, A. Klamm, D. Landgraf, "Automated detection of European wild mammal species in camera trap images with an existing and pre-trained computer vision model" European Journal of Wildlife Research, vol. 66, no. 4, 2020, doi: 10.1007/s10344-020-01404-y.

[20] J. Parham, C. Stewart, J. Crall, D. Rubenstein, J. Holmberg, T. Berger-Wolf, "An Animal Detection Pipeline for Identification" 2018 IEEE Winter Conference, pp. 1075–1083.

[21] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, J. Clune, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning" Proceedings of the National Academy of Sciences USA, vol. 115, no. 25, 2018, doi: 10.1073/pnas.1719367115.

[22] M. S. Palmer, S. E. Huebner, M. Willi, L. Fortson, C. Packer, "Citizen science, computing, and conservation: How can "Crowd AI" change the way we tackle large-scale ecological challenges?" Human Computation Journal, vol. 8, no. 2, pp. 54–75, 2021, doi: 10.15346/hc.v8i2.123.

[23] "Zooniverse." https://www.zooniverse.org/ (accessed Feb. 20, 2023).

[24] J. A. Ahumada, E. Fegraus, T. Birch, N. Flores, R. Kays, T. G. O'Brien, J. Palmer, S. Schuttler, J. Y. Zhao, W. Jetz, M. Kinnaird, S. Kulkarni, A. Lyet, D. Thau, M. Duong, R. Oliver, A. Dancer, "Wildlife Insights: A Platform to Maximize the Potential of Camera Trap and Other Passive Sensor Wildlife Data for the Planet" Environmental Conservation, vol. 47, no. 1, pp. 1–6, 2020, doi: 10.1017/S0376892919000298.

[25] T. Y. Berger-Wolf, D. I. Rubenstein, C. V. Stewart, J. A. Holmberg, J. Parham, S. Menon, J. Crall, J. van Oast, E. Kiciman, L. Joppa, "Wildbook: Crowdsourcing, computer vision, and data science for conservation" Data For Good Exchange 2017, 2017, http://arxiv.org/pdf/ 1710.08880v1

[26] Jason Parham, Jon Crall, Charles Stewart, Tanya Berger-Wolf, Dan Rubenstein, "Animal Population Censusing at Scale with Citizen Science and Photographic Identification" AAAI Spring Symposia, pp. 37–45, 2017.

[27] P. D. Meek, K. Vernes, G. Falzon, "On the Reliability of Expert Identification of Small-Medium Sized Mammals from Camera Trap Photos" Wildlife Biology in Practice, vol. 9, no. 2, 2013, doi: 10.2461/wbp.2013.9.4.

[28] "Wildlife Protection Solutions." https:// wildlifeprotectionsolutions.org/ (accessed Nov. 21, 2022).

[29] G. R. Hillman, B. Würsig, G. A. Gailey, N. Kehtarnavaz, A. Drobyshevsky, B. N. Araabi, H. D. Tagare, D. W. Weller, "Computer-assisted photo-identification of individual marine vertebrates: a multi-species system" Aquatic Mammals, vol. 29, no. 1, pp. 117–123, 2003, doi: 10.1578/016754203101023960.

[30] H. J. Weideman, Z. M. Jablons, J. Holmberg, K. Flynn, J. Calambokidis, R. B. Tyson, J. B. Allen, R. S. Wells, K. Hupman, K. Urian, C. V. Stewart, "Integral Curvature Representation and Matching Algorithms for Identification of Dolphins and Whales" 2017 IEEE International Conference on Computer Vision Workshops: ICCVW, pp. 2831–2839, 2017, doi: 10.1109/ICCVW.2017.334.

[31] R. B. Sherley, T. Burghardt, P. J. Barham, N. Campbell, I. C. Cuthill, "Spotting the difference: towards fully-automated population monitoring of African penguins Spheniscus demersus" Endangered Species Research, vol. 11, pp. 101–111, 2010, doi: 10.3354/esr00267.

[32] E. Nepovinnykh, T. Eerola, H. Kalviainen, "Siamese Network Based Pelage Pattern Matching for Ringed Seal Re-identification" 2020 IEEE Winter Applications of Computer Vision Workshops (WACVW), pp. 25–34, 2020, doi: 10.1109/WACVW50321.2020.9096935.

[33] R. Bogucki, M. Cygan, C. B. Khan, M. Klimek, J. K. Milczek, M. Mucha, "Applying deep learning to right whale photo identification" Conservation Biology, vol. 33, no.3, pp. 676-684, 2019, doi: 10.1111/cobi.13226.

[34] M. Hamilton, S. Raghunathan, A. Annavajhala, D. Kirsanov, E. de Leon, E. Barzilay, I. Matiach, J. Davison, M. Busch, M. Oprescu, R. Sur, R. Astala, T. Wen, C. Park, "Flexible and Scalable Deep Learning with MMLSpark" Journal of Machine Learning Research, vol. 82, 2017, https://arxiv.org/pdf/1804.04031

[35] A. C. Ferreira, L. R. Silva, F. Renna, H. B. Brandl, J. P. Renoult, D. R. Farine, R. Covas, C. Doutrelant, "Deep learning-based methods for individual recognition in small birds" Methods in Ecology and Evolution, vol. 11, no. 9, pp. 1072–1085, 2020, doi: 10.1111/2041-210X.13436.

[36] S. J. Carter, I. P. Bell, J. J. Miller, P. P. Gash, "Automated marine turtle photograph identification using artificial neural networks, with application to green turtles" Journal of Experimental Marine Biology and Ecology, vol.

452, pp. 105–110, 2014, doi: 10.1016/j.jembe.2013.12.010.

[37] S. Li, J. Li, H. Tang, R. Qian, W. Lin, "ATRW: A Benchmark for Amur Tiger Re-identification in the Wild" Proceedings of the 28th ACM International Conference on Multimedia, pp. 2590–2598, 2019. doi: 10.1145/3394171.3413569.

[38] K. S. Pereira, L. Gibson, D. Biggs, D. Samarasinghe, A. R. Braczkowski, "Individual Identification of Large Felids in Field Studies: Common Methods, Challenges, and Implications for Conservation Science" Frontiers in Ecology and Evolution, vol. 10, 2022, doi: 10.3389/fevo.2022.866403.

[39] X. Dong, J. Shen, "Triplet Loss in Siamese Network for Object Tracking" Proceedings of European Conference on Computer Vision (ECCV), pp. 472-488, 2018.

[40] D. Organisciak, C. Riachy, N. Aslam, H. P. Shum, "Triplet Loss with Channel Attention for Person Re-identification" Journal of World Society for Computer Graphics (JWSCG), vol. 27, no. 2, 2019, doi: 10.24132/JWSCG.2019.27.2.9.

[41] N. Dlamini, T. L. van Zyl, "Comparing Class-Aware and Pairwise Loss Functions for Deep Metric Learning in Wildlife Re-Identification" Sensors (Switzerland), vol. 21, no. 18, 2021, doi: 10.3390/s21186109.

[42] D. G. Lowe, "Object recognition from local scale-invariant features" The proceedings of the 7th IEEE International Conference on Computer Vision, 1999, vol.2, pp. 1150-1157, doi: 10.1109/ICCV.1999.790410.

[43] R. B. Nipko, B. E. Holcombe, M. J. Kelly, "Identifying Individual Jaguars and Ocelots via Pattern-Recognition Software: Comparing HotSpotter and Wild-ID" Wildlife Society Bulletin, vol. 44, no. 2, pp. 424–433, 2020, doi: 10.1002/wsb.1086.

[44] D. T. Bolger, T. A. Morrison, B. Vance, D. Lee, H. Farid, "A computer-assisted system for photographic mark-recapture analysis" Methods in Ecology and Evolution, vol. 3, no. 5, pp. 813–822, 2012, doi: 10.1111/j.2041-210X.2012.00212.x.

[45] T. Y. Berger-Wolf, Di Rubenstein, C. V. Stewart, J. Holmberg, J. Parham, J. Crall, "IBEIS: Image-based ecological information system: From pixels to science and conservation" Bloomberg Data for Good Exchange Conference, vol. 2, 2015.

[46] J. P. Crall, C. V. Stewart, T. Y. Berger-Wolf, D. I. Rubenstein, S. R. Sundaresan, "Hotspotter — patterned species instance recognition" 2013 IEEE Workshop on Applications of Computer Vision (WACV), pp. 230–237, 2013, doi: 10.1109/WACV.2013.6475023.

[47] J. Parham, C. Stewart, J. Crall, D. Rubenstein, J. Holmberg, T. Berger-Wolf, "An animal detection pipeline for identification" IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1075–1083, 2018.

[48] D. Schofield, A. Nagrani, A. Zisserman, M. Hayashi, T. Matsuzawa, D. Biro, S. Carvalho, "Chimpanzee face recognition from videos in the wild using deep learning" Science Advances, vol. 5, no. 9, 2019, doi: 10.1126/sciadv.aaw0736.

[49] D. Crouse, R. L. Jacobs, Z. Richardson, S. Klum, A. Jain, A. L. Baden, S. R. Tecot, "LemurFaceID: a face recognition system to facilitate individual identification of lemurs" BMC Zoology, vol. 2, no. 1, 2017, doi: 10.1186/s40850-016-0011-9.

[50] C.-A. Brust, T. Burghardt, M. Groenenberg, C. Kading, H. S. Kuhl, M. L. Manguette, J. Denzler, "Towards Automated Visual Monitoring of Individual Gorillas in the Wild" 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 2820–2830, 2017.

[51] P. Chen, P. Swarup, W. M. Matkowski, A. W. K. Kong, S. Han, Z. Zhang, H. Rong, "A study on giant panda recognition based on images of a large proportion of captive pandas" Ecology and Evolution, vol. 10, no. 7, pp. 3561-3573, 2020, doi: 10.1002/ece3.6152.

[52] M. Clapham, E. Miller, M. Nguyen, R. C. van Horn, "Multispecies facial detection for individual identification of wildlife: a case study across ursids" Mammalian Biology, 2022, doi: 10.1007/s42991-021-00168-5.

[53] K. Wang, C. Chen, Y. He, "Research on pig face recognition model based on keras convolutional neural network" IOP Conference Series: Earth and Environmental Science, vol. 474, no. 3, 2020, doi: 10.1088/1755-1315/474/3/032030.

[54] M. Marsot, J. Mei, X. Shan, L. Ye, P. Feng, X. Yan, C. Li, Y. Zhao, "An adaptive pig face recognition approach using Convolutional Neural Networks" Computers and Electronics in Agriculture, vol. 173, 2020, doi: 10.1016/j.compag.2020.105386.

[55] M. S. McCarthy, C. Stephens, P. Dieguez, L. Samuni, M.-L. Després-Einspenner, B. Harder, A. Landsmann, L. K. Lynn, N. Maldonado, Z. Ročkaiová, J. Widness, R. M. Wittig, C. Boesch, H. S. Kühl, M. Arandjelovic, "Chimpanzee identification and social network construction through an online citizen science platform" Ecology and Evolution, vol. 11, no.4, 2020, doi: 10.1002/ece3.7128.

[56] O. Moskvyak, F. Maire, A. O. Armstrong, F. Dayoub, M. Baktashmotlagh, "Robust Re-identification of Manta Rays from Natural Markings by Learning Pose Invariant Embeddings" Digital Image Computing: Techniques and Applications (DICTA), 2021, doi: 10.1109/DICTA52665.2021.9647359.

[57] C. Town, A. Marshall, N. Sethasathien, "Manta Matcher: automated photographic identification of manta rays using keypoint features" Ecology and evolution, vol. 3, no. 7, pp. 1902–1914, 2013, doi: 10.1002/ece3.587.

[58] Z. Tu, W. Xie, D. Zhang, R. Poppe, R. C. Veltkamp, B. Li, J. Yuan, "A survey of variational and CNN-based optical flow techniques" Signal Processing: Image Communication, vol. 72, pp. 9–24, 2019, doi: 10.1016/j.image.2018.12.002.

[59] S. Brahimi, N. Ben Aoun, C. Ben Amar, A. Benoit, P. Lambert, "Multiscale Fully Convolutional DenseNet for Semantic Segmentation" Journal of World Society for Computer Graphics (JWSCG), vol. 26, no. 2, 2018, doi: 10.24132/JWSCG.2018.26.2.5.

[60] A. Leipnitz, T. Strutz, O. Jokisch, "Performance Assessment of Convolutional Neural Networks for Semantic Image Segmentation" 27th International Conference on Computer Graphics, Visualization and Computer Vision 2019, doi: 10.24132/CSRN.2019.2901.1.4.

ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.31, No-1-2, 2023

ALIVE: Adaptive-Chromaticity for Interactive Low-light Image and Video Enhancement

Sumit Shekhar[1]
sumit.shekhar@hpi.uni-potsdam.de

Max Reimann[1]
max.reimann@hpi.uni-potsdam.de

Jobin Idiculla Wattaseril[1]
jobin.wattaseril@hpi.uni-potsdam.de

Amir Semmo[2]
amir.semmo@digitalmasterpieces.com

Jürgen Döllner[1]
doellner@uni-potsdam.de

Matthias Trapp[1]
matthias.trapp@hpi.uni-potsdam.de

[1]Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam,
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany.

[2]Digital Masterpieces GmbH,
August-Bebel-Str. 26, 14482 Potsdam, Germany.

ABSTRACT

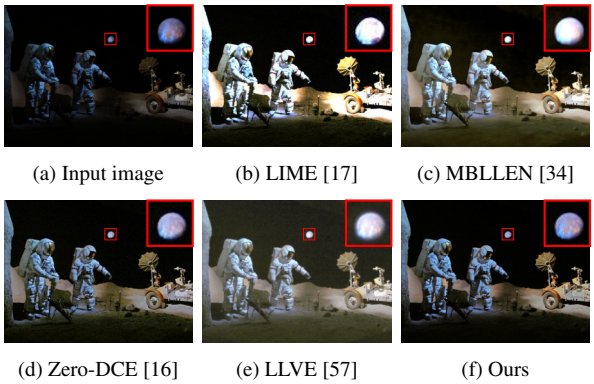
Image acquisition in low-light conditions suffers from poor quality and significant degradation in visual aesthetics. This affects the visual perception of the acquired image and the performance of computer vision and image processing algorithms applied after acquisition. Especially for videos, the additional temporal domain makes it more challenging, wherein quality is required to be preserved in a temporally coherent manner. We present a simple yet effective approach for low-light image and video enhancement. To this end, we introduce *Adaptive Chromaticity*, which refers to an adaptive computation of image chromaticity. The above adaptivity avoids the costly step of low-light image decomposition into illumination and reflectance, employed by many existing techniques. Subsequently, we achieve interactive performance, even for high resolution images. Moreover, all stages in our method consists of only point-based operations and high-pass or low-pass filtering, thereby ensuring that the amount of temporal incoherence is negligible when applied on a per-frame basis for videos. Our results on standard low-light image datasets show the efficacy of our method and its qualitative and quantitative superiority over several state-of-the-art approaches. We perform a user study to demonstrate the preference for our method in comparison to state-of-the-art approaches for videos captured in the wild.

Keywords

real-time, interactive, low-light, image, video, enhancement

1 INTRODUCTION

Due to unavoidable technical or environmental constraints, images and videos captured in poor lighting conditions suffer from severe degradation of visual quality. On most occasions, it is challenging for such visual media to be consumed for high-level tasks such as object detection or tracking due to deterioration or lack of information. Moreover, poor visual quality negatively impacts the overall aesthetics, and thus, the experience of end-users.



(a) Input image (b) LIME [17] (c) MBLLEN [34]

(d) Zero-DCE [16] (e) LLVE [57] (f) Ours

Figure 1: Comparison of LLIE results for three image-based (b to d) and one video-based (e) method. Our method (f) can brighten image while preserving details and avoiding artifacts in terms of over-exposedness, noise, and desaturation.

Numerous algorithms have been proposed for Low-light Image Enhancement (LLIE) (Fig. 1) and a few for video enhancement as well. A class of methods is based

Table 1: Comparing existing low-light image enhancement techniques in the context of interactivity. Here, the color green denotes the aspect which is favourable to interactive enhancement while the color red denotes otherwise.

	LIME [17]	SRIE [31]	MBLLEN [34]	RetinexNet [49]	Zero-DCE [16]	LLVE [57]	Ours
Provides enhancement editing at inference time?	Yes	Yes	No	No	No	No	Yes
Is the performance interactive?	No	No	Yes	Yes	Yes	Yes	Yes

on Retinex theory [25] that assumes the image to be a product of illumination and reflectance. Such Retinex-based approaches decompose the image into illumination and/or reflectance components, based on specific priors. However, finding effective priors is challenging and inaccuracies can result in artifacts and color deviations in the enhanced output. Further, the runtime for such a decomposition, employing a complex optimization process, is relatively long [32]. In comparison, deep-learning-based approaches are faster than conventional methods and learn the underlying prior using the given data distribution. However, they tend to suffer from limited generalization capability. The above could be due to limited/synthetic training data, ineffective network structures, or unrealistic assumptions [28]. Further, learning based approaches do not allow interactive editing of enhancement settings during test time and requires a complete re-training for this purpose. Therefore, we aim to develop a practical solution for LLIE, which adapts to different low-light conditions and also has low computational complexity for enabling interactive enhancement editing on commodity hardware (Tab. 1).

To achieve the above objective, we develop a method based on Retinex theory, the basis for various conventional and learning-based techniques. We avoid the compute-intensive decomposition step and propose an adaptive way to transition into baseline-reflectance (i.e., chromaticity) [4] via parameter tuning. We refer to it as *Adaptive Chromaticity (AC)*, which forms the basis for our approach. The adaptive transition into chromaticity can efficiently increase the output brightness while being robust against dark (or low-intensity) pixels. Moreover, it prevents amplification of sensor noises, which are common in low-light images. To further prevent noise amplification during enhancement, we decompose the input image into coarse and fine attributes, generally referred to as *base* and *detail* components respectively [2]. We generate multiple ACs for the *base* layer with varying levels of brightness followed by a multi-scale fusion step. Different levels of brightness prevents over/under-exposedness, while multi-scale fusion maintains spatial consistency. The *detail* layer is finally added to the result, thus preserving fine image details.

Unlike images, low-light video enhancement has received less attention. Application of image-based methods to videos on a per-frame basis is usually temporally incoherent and leads to flickering artifacts. Dark pixels, significantly contributing to noise amplification, is often the major source of temporal incoherence. Due to the ability of our method to robustly handle such pixels, the degree of incoherence is reduced significantly. Even the per-frame application of our image-based approach is superior to an existing video-specific approach. Our contributions are summarized as follows, we propose:

1. Adaptive Chromaticity (AC) to efficiently increase image brightness while preventing noise amplification.
2. An approach for interactive low-light image enhancement based on exposure fusion of multiple ACs.
3. A per-frame application of our image-based approach for videos that performs out-of-the-box without introducing significant temporal incoherence.

2 RELATED WORK

Low-Light Enhancement of Images:

One of the earliest algorithms for low-light image enhancement is based on Retinex theory. Jobson *et al.* [23, 22] propose center/surround Retinex at single-scale and multi-scale to achieve plausible results for dynamic range compression and color restoration. Various follow-up methods employ Retinex theory as their basis and propose complex optimization strategies to estimate reflectance and/or illumination for the purpose of low-light image enhancement [48, 14, 15, 17, 5, 31, 60, 13, 59, 41, 18]. Fu *et al.* [15] propose a weighted variational model for simultaneous reflectance and illumination estimation. Guo *et al.* [17] (LIME) perform refinement of an initial illumination map via a structure prior to obtain a well constructed illumination map thereby enabling enhancement. Li *et al.* [31] (SRIE) employ a fidelity term for gradients of the reflectance to reveal the structure details and also estimate a noise map out of their Retinex model. Ren *et al.* [41] propose a robust model to estimate reflectance and illumination maps simultaneously, with provision to suppress noise in the reflectance map. Most of the above techniques have long run-time involving CPU-based complex optimization solving for image decomposition. We also use the Retinex image-formation model as our premise. However, unlike existing techniques we do not perform the decomposition of image into reflectance and/or illumination layers, thus, achieving interactive performance on commodity hardware.

Another class of methods for low-light image enhancement is based on Histogram Equalization (HE), wherein

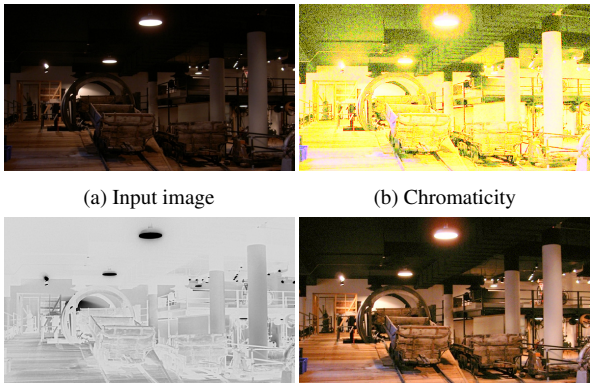
the histogram of the input image is stretched thereby improving its contrast [38]. Similar to Retinex-based approaches, various extension to the basic principle have been proposed [10, 1, 7, 26]. Celik and Tjahjadi [7] employ a variational approach for contrast enhancement using inter-pixel contextual information. Lee *et al*. [26] use a layered difference of 2D histograms and thus achieve better results than previous HE-based approaches. However, the primary focus of HE-based methods is contrast enhancement instead of physically-based illumination editing, thus having the potential risk of over- and/or under- exposed pixels.

Recently, deep learning has also been used substantially to address the problem of low-light image enhancement. Methods based on various learning strategies, such as supervised [33, 34, 49, 6, 40, 61, 51, 63], semi-supervised [54], unsupervised [16, 21, 27], and reinforcement learning [56] have been proposed. Lore *et al*. [33] present the first deep learning-based method in this context (LLNet) that employs stacked-sparse denoising autoencoder to lighten and denoise low-light images simultaneously. Lv *et al*. [34] (MBLLEN) propose an end-to-end multibranch network for simultaneous enhancement and denoising. Wei *et al*. [49] (RetinexNet) use Decom-Net for image decomposition followed by an Enhance-Net for illumination adjustment. The training is unsupervised for both the networks while the Enhance-Net also includes a joint denoising operation. Ren *et al*. [40] design an encoder-decoder network for global image enhancement and a separate recurrent neural network for further edge enhancement. Similar to Ren *et al*. [40], Zhu *et al*. [63] propose a method called EEMEFN, which consists of two stages: multi-exposure fusion and edge enhancement. Wang *et al*. [47] propose a network called DeepUPE to model image-to-image illumination and collect an expert-retouched dataset. Zhang *et al*. [61] propose a network called KinD based on Retinex theory and design a restoration module to counterbalance noise. Guo *et al*. [16] (Zero-DCE) estimates a set of best-fitting light-enhancement curves that iteratively enhances a given input image. The training is unsupervised and the method is efficient involving simple non-linear curve mapping. Chen *et al*. [8] collect a dataset named SID and train a U-Net [42] to estimate enhanced sRGB images from raw low-light images. Although learning-based methods can produce visually plausible results, they have limited generalization capability in comparison to conventional methods [28]. Moreover, unlike ours, most of the learning-based methods do not allow interactive editing of enhancement at inference time. For a new enhancement setting one has to retrain the network. Two methods which are closely related to our approach are that of Ying *et al*. [55] and Zheng *et al*. [62], both generate multiple images with different exposures followed by exposure fusion. Ying *et al*. employ a complex strategy with multiple steps to generate the exposure sequence followed by a computationally expensive optimization solving for fusion. The exposure sequence generation for Zheng *et al*. is relatively simpler than above, however, they make use of deep-learning to further enhance the sequence as an intermediate step. In comparison, our exposure sequence generation is straightforward and does not require any learning-based post-processing.

Apart from the above, existing techniques when applied on a per-frame basis, e.g., for videos, usually suffer from temporal incoherence. We prevent such inconsistency to a large degree by resorting to only point-based operations and high- or low- pass filtering.

Low-Light Enhancement of Videos:

In comparison to images, low-light video enhancement has received significantly less attention. One straightforward way to do so would be to stabilize a per-frame based application of low-light image enhancement technique using blind video consistent filtering approaches [3, 24, 43]. These techniques inherently make use of vision-based attributes such as optical flow [3, 24] or saliency masks [43] for temporal stabilization. However, computation of above vision-based attributes itself can be inaccurate/challenging for low light videos. Lv *et al*. [34] propose an extension for their learning based approach for images by replacing their 2D convolution layers with 3D ones and train it on synthetic video data. In order to collect real-world training data, Chen *et al*. [9] capture videos for static scenes with the corresponding long-exposure ground truths and ensure generalization for dynamic scenes by using a Siamese network. Jian and Zheng [20] develop a setup to capture bright and dark dynamic video pairs and subsequently train it using a modified 3D U-Net. However, with their sophisticated setup – consisting of two cameras, a relay lens and a beam splitter – the authors do not capture diverse scenes and objects as part of training data. Triantafyllidou *et al*. [45] propose a low-light video synthesis pipeline (SIDGAN) that maps "in the wild" videos into a corresponding low-light domain. The above approach employs a semi-supervised dual CycleGAN to produce dynamic video data (RAW-to-RGB) with intermediate domain mapping. In a recent work, Zhang *et al*. [57] (LLVE) enforce temporal stability for low-light video enhancement by predicting optical flow for a single image and synthesizing short range video sequences. However, their quality of enhancement is low in comparison to existing techniques (Sec. 4.4). We do not perform any temporal processing specific for videos, however our low-light image enhancement algorithm introduces only negligible temporal incoherence.



(a) Input image (b) Chromaticity

(c) Intensity Difference y (d) Adaptive Chromaticity

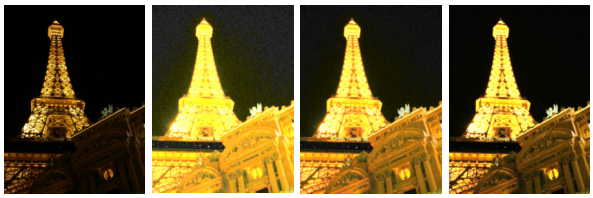
Figure 2: Given an input image (a), the noise in the chromaticity (b) is higher for low-intensity pixels with a larger intensity difference (c), which is significantly reduced for (d) adaptive chromaticity (with $\alpha = 0.3$ and $\gamma = 0.8$).

3 METHOD

According to the Retinex model, an image I can be expressed as the product of a *reflectance* layer $R \in \mathbb{R}^3$ and an *illumination* layer $L \in \mathbb{R}$ [25]: $I(x) = R(x) \times L(x)$, where the operator $\times$ denotes pixel-wise (x) multiplication. For the above equation to hold we assume only diffuse-reflection in the scene with monochromatic illumination. As a baseline, image "intensity" and "chromaticity" can be considered as the illumination and reflectance layer, respectively [4]. To compute image intensity one can employ different approaches, such as: norm or the maximum of the individual color channels. However, both does not yield desirable results for our purpose of perceptually plausible editing. To this end, we consider the *luma* (Y-channel in YCbCr color space) as our intensity operator $In(\cdot)$, since this satisfies the above objective. Chromaticity is correspondingly obtained by dividing the image with its intensity (Eqn. (1)). The above division operation is able to significantly reduce shading and shadows in the scene, which only affects the intensity, thus making the chromaticity relatively brighter than the input image. Moreover, it also acts as a normalizing factor for pixel color and saturates it, further making it appear perceptually bright. For an input image I with color channels r, g, and b in sRGB color space using 8-bit per channel (i.e., 24-bit color depth), we define intensity (following ITU-R BT.601) by the operator $In(\cdot)$ and chromaticity C as follows:

$$In(I) = 0.299r + 0.587g + 0.144b \text{ and } C = \frac{I}{In(I)}. \quad (1)$$

The brightening aspect of chromaticity is a preferable characteristic for low-light image enhancement. However, chromaticity suffers from undesirable artifacts in terms of *noise* and *color-shifts*, especially for low-intensity pixels (Fig. 2b).



(a) Input (b) y^2 (c) $\exp(y)$ (d) $\tan(y \cdot \frac{\pi}{2})$

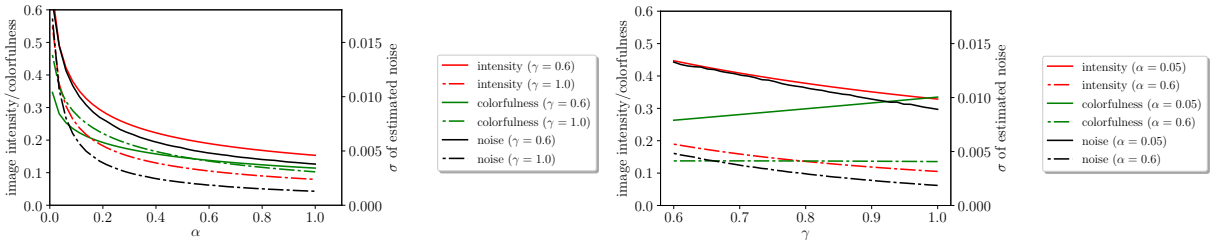
Figure 3: Our Single-Exposure (SE) output for $\alpha = 0.05$, and $\gamma = 0.7$ employing different adaptive functions $f(y)[= y^2, \exp(y), \tan(y \cdot \frac{\pi}{2})]$.

3.1 Adaptive Chromaticity

In order to preserve the brightening effect of chromaticity while avoiding artifacts, we introduce *Adaptive Chromaticity (AC)*. For identifying a low-intensity pixel, we compute the difference between pixel intensity, $In(\cdot)$, and the maximum intensity value $MaxIn$. For low-intensity pixels, this difference defined as $y = MaxIn - In(\cdot)$ would be comparatively larger. For example, for an intensity image encoded in the range of 0 to 1, $MaxIn = 1$ and for a low-intensity pixel p with $In(\cdot) = 0.05$ the difference $y(p) = 0.95$ is large. Similarly, for a high-intensity pixel q with $In(\cdot) = 0.8$ the difference $y(q) = 0.2$ is small (Fig. 2c). The above forms the basis for defining adaptive chromaticity (A_c), wherein we add an adaptive term, as a function of y, in the denominator while computing chromaticity (Eqn. (1)). To further increase the brightness and prevent color-shifts, we perform a non-linear scaling using *gamma correction*

$$A_c(I, \alpha, \gamma) = \left(\frac{I}{In(I) + \alpha f(y)} \right)^{\gamma}. \quad (2)$$

Here, $f(y)$ is a function in terms of y, α is a control parameter, and γ is a parameter for gamma correction. The adaptive function $f(y)$ should be chosen such that its value is close to zero when y is small and is substantially high for large values of y. Thus, by tuning the control parameter α, we can smoothly transition between the bright chromaticity (when $\alpha \to 0$) and a complete dark image (when $\alpha \to \infty$). The intuition behind the adaptive denominator in Eqn. (2) is that we divide by a larger value for low-intensity pixels as compared to high-intensity pixels, thereby, reducing undesirable artifacts. For adaptivity, a function f should be chosen that satisfies the above property and is efficient to compute. Among possible variants, y^2 and $\exp(y)$ produces desirable results. However, $f(y) = \tan(y \cdot \frac{\pi}{2})$ works significantly better in terms of noise reduction and also gives plausible results, see Fig. 3. The AC brightens an image while significantly reducing chromaticity-related artifacts (Fig. 2d) and forms the basis for our low-light image and video enhancement methodology.



(a) Impact of varying α on the resultant AC.

(b) Impact of varying γ on the resultant AC.

Figure 4: Changes in the characteristics of resultant Adaptive Chromaticity in terms of intensity, colorfulness and noise while varying α (Fig. 4a) and γ (Fig. 4b). Intensity is computed using Eqn. (1). Colorfulness represents the perceptual amount of saturation following [19]. Image noise is calculated using skimage estimate_sigma [11] based on a wavelet-based estimator [12] of the gaussian noise standard deviation σ. Metrics are computed and averaged over the LIME dataset [17].

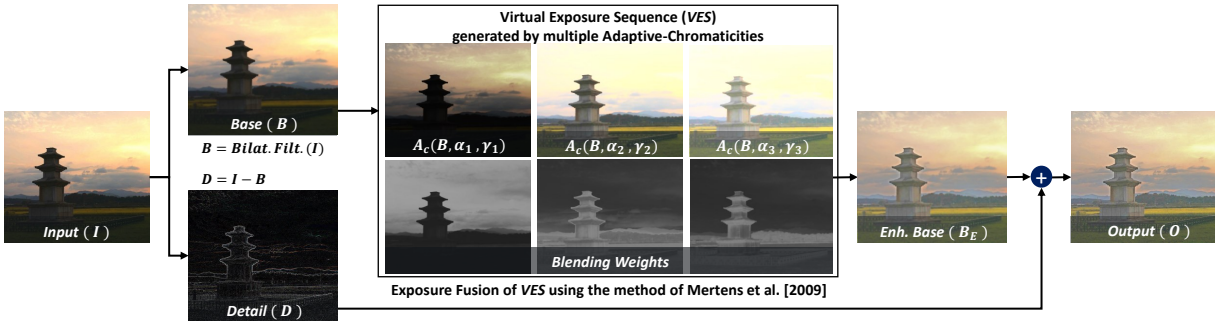


Figure 5: Flowchart of our low-light image enhancement algorithm. To prevent noise amplification we decompose the input image into *Base* and *Detail* layers. Subsequently, multiple Adaptive Chromaticities (ACs) are generated (Sec. 3.1) for the Base layer to create a Virtual Exposure Sequence (VES) (Sec. 3.2). Following to that, these images are blended guided by quality measures of contrast, saturation, and well-exposedness (Sec. 3.2). The above is performed in a multi-resolution fashion, as proposed by Mertens *et al.* [35]. Finally, the Detail layer is added to the enhanced Base layer to obtain the final output.

Parameter Analysis:

We analyse the changes in the characteristics of resultant AC in terms of image *intensity*, *colorfulness* and *noise* while varying the parameters α and γ in Fig. 4. Decreasing α leads to a quadratic increase in all three metrics (Fig. 4a). Moreover, note the significant decline in noise for higher values of alpha (> 0.8). On the other hand, decreasing γ linearly increases noise and intensity, while at the same time desaturates the image (Fig. 4b). The desaturating nature of γ plays a counterbalancing role to the effect of α in terms of *colorfulness* thereby preventing color-shifts. It is thus evident, that both α and γ needs to be adjusted to brighten the image while retaining the original saturation level, and also highlights that denoising is an essential requirement during low-light enhancement.

3.2 Our Approach for LLIE

To further reduce noise amplification during enhancement, we decompose the input image into *Base* (B) and *Detail* (D) components [2], and only enhance *Base*, as depicted in our full LLIE-algorithm flowchart in Fig. 5. We assume that most of the noise due to low-light conditions is captured in the high-frequency *Detail* layer. Thus, enhancing only *Base* layer will lead to negligible noise amplification. For base-detail decomposition we make use of Bilateral Filter [44], however, in principle, one can use any edge-preserving filter for this purpose. We use

$$B = BilatFilt(I,\ \sigma_s,\ \sigma_t) \quad D = I - B \tag{3}$$

where $\sigma_s = 1.0$ (spatial width) and $\sigma_t = 0.5$ (tonal range) works fine with most images (or video-frames). Following the above decomposition, the *Base* layer is enhanced via AC using a single-exposure (SE) or multiple-exposure (ME) setting. In either case, subsequently the *Detail* layer is added to the enhanced *Base* to obtain the final result (Fig. 6). For single-exposure, a single AC of base layer is assigned as its enhanced version (Fig. 6e). Multi-exposure enhancement involves computing multiple ACs of the base layer and is proposed as a two-step process consisting of Virtual Exposure Sequence (VES) *generation* and *fusion*.

VES Generation:

The overall exposedness of an image is increased by lowering α and/or γ values in Eqn. (2). However, the brightening effect of either of these parameters α or γ is slightly different. For lower values of α, increase in brightness comes at the cost of color-shifts (Fig. 7a, Fig. 7d). On the other hand, for lower γ values, an

(a) Input image (b) Chromaticity (c) Base (d) Detail (e) Single Exp. (SE) (f) Multi Exp. (ME)

Figure 6: For a low-light image (a) the corresponding chromaticity (b) has artifacts in terms of color-shifts and noise. To overcome noise amplification, we decompose the image into *Base* (c) and *Detail* (d) layers using a bilateral filter ($\sigma_s = 1.0, \sigma_t = 0.5$). Further, chromaticity-based artifacts are reduced by employing Adaptive Chromaticity (AC) and for the single-exposure approach, an enhanced image is obtained as the sum of AC ($\alpha = 0.1, \gamma = 0.8$) of Base layer and Detail layer (e). To further preserve details during enhancement, we use a multi-exposure fusion technique (3 exposure levels – $\alpha_1 = 0.03, \alpha_2 = 0.1, \alpha_3 = 2.0$ and $\gamma_1 = 0.7, \gamma_2 = 0.8, \gamma_3 = 0.5$ – and 4 pyramid levels) to obtain a high-quality output (f).

increase in brightness is accompanied with desaturation (Figs. 7d to 7f). For both α and γ, lower values leads to increase in noise (Fig. 7d) (see Sec. 3.1). Increasing the exposedness by tuning either α or γ is a point-based operation and does not respect the relative contrast within the image. The above leads to the problem, wherein already visible regions in the low-light image are over-exposed while increasing the brightness. It is similar to challenges in High Dynamic Range (HDR) photography, which aims to preserve all the details within an HDR scene.

We do not assume an HDR version of the image at our disposal, however we can generate different levels of brightness by varying the values of α and γ respectively. Thus, we generate a *virtual exposure sequence* for the given base layer by computing multiple ACs. For the base layer B, an exposure sequence $\{E_k | \, k = 1 \ldots N\}$ is obtained based on the parameter series $\{(\alpha_k, \gamma_k) \, | \, k = 1 \ldots N\}$, with

$$E_k = A_c(B, \alpha_k, \gamma_k). \tag{4}$$

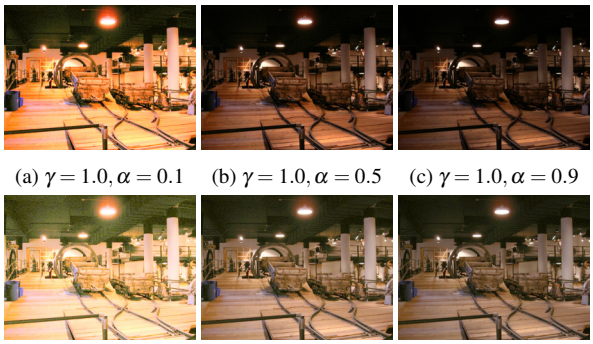
Subsequently, an HDR image can be generated using the above sequence of images and further tone-mapping can preserve details in both bright and dark regions while enhancing it [39].

VES Fusion:

For efficiency, we avoid the step of computing an HDR image, and directly fuse the multiple exposures into a high-quality, low dynamic range image using the exposure-fusion technique of Mertens *et al.* [35]. The well-exposedness of an image in the exposure sequence is determined based on quality measures of *contrast* (c_k), *saturation* (s_k), and *well-exposedness* (e_k) on a per-pixel basis.The three quality measures are combined into a joint weighting function

$$w_k = {c_k}^{\upsilon_c} \cdot {s_k}^{\upsilon_s} \cdot {e_k}^{\upsilon_e}, \tag{5}$$

where the above product can be seen as logical conjunction and the parameters υ_c, υ_s, and υ_e control the influence of individual quality measures. Finally, the obtained sequence of weight maps are normalized such that they sum up to one at each pixel location, thereby ensuring consistent results, as follows:



(a) $\gamma = 1.0, \alpha = 0.1$ (b) $\gamma = 1.0, \alpha = 0.5$ (c) $\gamma = 1.0, \alpha = 0.9$

(d) $\gamma = 0.5, \alpha = 0.1$ (e) $\gamma = 0.5, \alpha = 0.5$ (f) $\gamma = 0.5, \alpha = 0.9$

Figure 7: Virtual Exposure Sequence (VES) for the input image in Fig. 2: as a sequence of ACs generated by varying values of α and γ.

$$\widehat{w_k} = \frac{w_k}{\sum_{k=1}^{N} w_k}. \tag{6}$$

Once the weight maps are computed, a Laplacian pyramid $L(E_k)$ of each image and a Gaussian pyramid of each normalized weight map $G(\widehat{w_k})$ are generated. At each pyramid level l, the images are fused on per-pixel and per-color channel basis as

$$L(B_E)_l = \sum_{k=1}^{N} G(\widehat{w_k})_l L(E_k)_l. \tag{7}$$

The enhanced base layer, B_E, is obtained by collapsing the computed Laplacian pyramid $L(B_E)$. Following the above, we sum the detail layer (D) and the enhanced base layer (B_E) to obtain the final output O where,

$$O = B_E + \eta D, \tag{8}$$

and $\eta > 1$ is used to amplify the details in the final output [37]. However, large values of η (> 4.0) leads to halo-artifacts and unnatural looks. For most images

$\eta = 2.0$ gives visually plausible results. Note that the operations defined in Eqn. (1) till Eqn. (8) are all point-based where we have omitted the pixel-location x for the sake of clarity. All the steps in our method are efficiently summarized in Algo. 1.

4 RESULTS

4.1 Parameter Settings

The enhancement of the base layer for our Multi-Exposure (ME) version consists of two steps, for which the parameter settings are discussed as follows.

VES Generation:

Ideally, to capture fine details at different exposure levels, multiple images are required for the exposure sequence. However, the processing time will increase according to the number of images. Empirically, we determine three exposure levels ($N = 3$) as sufficient to preserve details at different levels of brightness. Further, we empirically determine $\gamma \in [0.6, 1.0]$ and $\alpha \in [0.01, 3.0]$ to result in well-exposed and less-noisy outputs. For most of the images, $\gamma_1 = 0.7, \alpha_1 = 0.03$ (high-exposure level), $\gamma_2 = 0.8, \alpha_2 = 0.1$ (mid-exposure level), and $\gamma_3 = 0.5, \alpha_3 = 2.0$ (low-exposure level) yield desirable results. For all the results in the paper, unless stated otherwise, we use the above parameter settings.

VES Fusion:

For exposure fusion, we set the weighting exponents for the quality measures to $\upsilon_c = \upsilon_s = \upsilon_e = 1$, as suggested by Mertens *et al*. [35]. During fusion, higher number of pyramid-levels facilitate the preservation of fine details. However, processing time increases with the number of levels, which is more pronounced for high-resolution images. Empirically, we determine four pyramid levels ($M = 4$) as a good trade-off between performance and quality.

4.2 Qualitative and Quantitative Results

We compare our results with state-of-the-art image-based methods: two conventional methods (SRIE [31] and LIME [17]), two supervised-learning based methods (MBBLEN [34] and RetinexNet [49]), a unsupervised-learning based method (Zero-DCE [16]), and a video-based method (LLVE [57]). The results are produced from publicly available source codes with respective parameter settings.

Images:

We test the above methods on images taken from the following datasets: LIME [17] (10 images), DICM [26] (44 images), NPE [48] (72 images), and VV [46] (24 images). For quantitative evaluation, we employ the Natural Image Quality Evaluator (NIQE) [36] metric to compare the performance of different methods on the above datasets. We choose this metric, as it is provides a completely blind quality measure for images and is based on only deviations from statistical regularities in natural images. Tab. 2 shows that overall we perform better than compared approaches except for Zero-DCE. We present qualitative comparison for enhanced image outputs in Fig. 8. The results of LIME(Fig. 8(b)) tends to be over-exposed, MBLLEN provides satisfactory brightening (Fig. 8(d)) however tends to over-smooth image details, the output of RetinexNet (Fig. 8(e)) seems to look unnatural, and for LLVE the results (Fig. 8(g)) appear to be hazy and desaturated. Our results are visually comparable to Zero-DCE and SRIE. However, in contrast to Zero-DCE, which requires a re-training of the complete network for a different degree of enhancement, our approach allows for interactive enhancement manipulation. Further, the slow optimization solving in SRIE makes it orders of magnitude slower than our approach Tab. 3. Moreover, the outcome of our user study, which includes a broad range of images (Fig. 9), indicates that overall our method is preferred over them.

For subjective evaluation of our method in the context of images, we perform a user study similar to Zhang *et al*. [57] comparing different techniques. We employ 9 different images (2 from LIME [17], 2 from DICM [26], 2 from NPE [48], and 3 from VV [46] datasets respectively) and compare 6 other techniques (5 image-based and 1 video-based) against our method. Thereby constituting 54 blind A/B tests which are presented in a random fashion to each participant. In total, 13 persons (7 female and 6 male) within the ages of 22 to 38 years participated in the study. We asked the participants to focus on the following aspects during comparison:

Exposure: As compared to the input, the enhanced image should be well-exposed, neither under- nor over-exposed.

Noise (and flickering): The enhanced image should have less noise (and flickering – in case of videos). However, the denoising should not be excessive as to remove details.

Color: The colors in the enhanced image should appear natural and it should not look over- or under-saturated.

For every low-light image, the participant is shown two enhanced versions of the image simultaneously (one of them is ours) and is asked to pick the version of their choice based on the above criteria. For the majority of

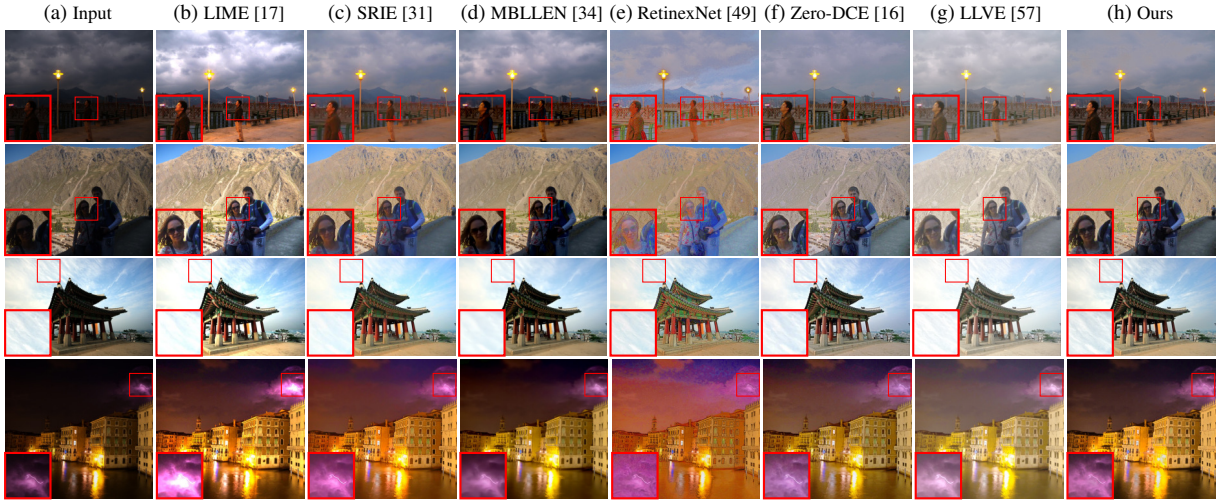


Figure 8: Low-light image enhancement results. Input images are taken from LIME [17], DICM [26], and VV [46] datasets.

Table 2: NIQE [36] (↓) values for images in LIME [17], DICM [26], NPE [48], and VV datasets. The best value is shown in red and the next best in blue.

Method	DICM	LIME	NPE	VV	Avg.
LIME	2.99	3.67	3.02	2.99	3.05
SRIE	3.27	4.29	3.45	3.25	3.42
MBLLEN	3.16	3.69	3.15	3.31	3.21
RetinexNet	3.59	3.63	3.62	**2.62**	3.45
LLVE	3.10	3.65	**2.98**	2.86	3.04
Zero-DCE	**2.48**	**3.10**	**2.92**	2.87	**2.79**
Ours	**2.84**	**3.22**	3.00	**2.66**	**2.92**

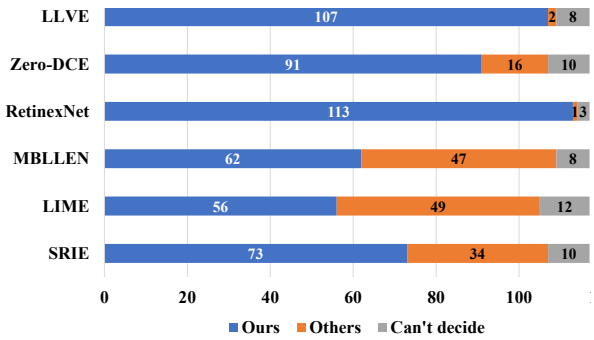


Figure 9: Statistics of user study results on low-light image enhancement. For 13 participants and 9 different images, we compare each existing method against ours through a total of 117 randomized A/B tests.

cases participants prefer our method against the existing approaches, see Fig. 9.

Videos:

To evaluate video-enhancement results, we perform a subjective user study similar to that of images explained in the previous paragraph. As test data, we make use of the challenging low-light videos provided by Li *et al.* in their survey LLIV [28]. In total, 13 persons (3 female, and 10 male) within the ages of 19 to 42 years participated in the study. Note that the above group of participants did not participate in the images-based user study to avoid any inherent bias between both the studies. The experiment consisted of 7 different low-light videos enhanced by ours and 6 other (5 image-based and 1 video-based) approaches. Two enhanced videos are shown to a participant simultaneously (one of them is ours), thereby constituting 42 blind A/B tests which are shown in a randomized order to each participant. Fig. 10 shows that our method surpasses all other methods including LLVE by a large margin.

4.3 Face Detection in the Dark

We investigate the performance of low-light enhancement methods for increasing the face-detection accuracy on low-light images. Specifically, following the settings presented in Li *et al.* [28], we use 500 randomly sampled images from the DARK FACE dataset [53] to measure performance of the state-of-the-art Dual Shot Face Detector (DSFD) [30] trained on the WIDER FACE dataset [52]. We use the author's DSFD implementation [29] with a non-maximum suppression threshold of 0.3 and evaluate using the dark face UG2 challenge evaluation tool [50]. Fig. 11 depicts the precision-recall curves as well as average precision (AP) under a 0.5 IoU threshold. The results show that all low-light enhancement methods achieve a substantial improvement in precision and recall over the unprocessed images (baseline result). For our method, the ME setting does not increase detection rates significantly. Moreover, we observe that shifting faces into a brightness and contrast range that the classifier

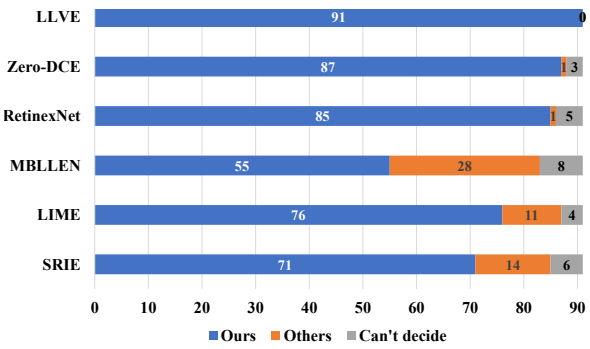


Figure 10: Statistics of the user study results on low-light video enhancement. For 13 participants and 7 different videos from LLIV [28] dataset, we compare each existing method against ours through a total of 91 randomized A/B tests.

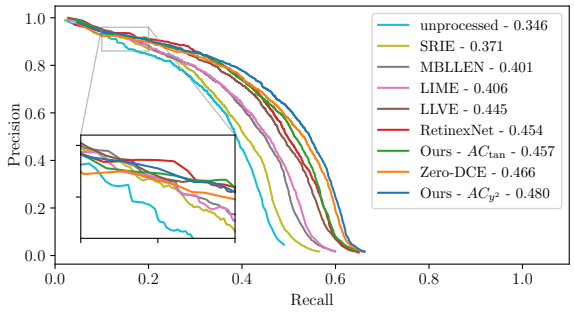


Figure 11: Precision-recall curves for face detection using DSFD [30] on dark-face images [53] enhanced using different LLIE methods. Average precision (AP) of each method is indicated in the legend, where "unprocessed" denotes the baseline AP on images in [53]. Our method uses adaptive chromaticity (AC) without exposure fusion, we compare two variants for $f(y)$, namely $f(y) = y^2$ with $\alpha = 0.25, \gamma = 0.6$ and $f(y) = \tan(y\frac{\pi}{2})$ with $\alpha = 0.25, \gamma = 0.3$.

has been trained on is crucial for accuracy improvement irrespective of overall image aesthetics. Thus we employ only Adaptive Chromaticity (AC) for this purpose. We investigate the performance of A_c (Eqn. (2)) for two different versions of $f(y)$, and find that while $f(y) = \tan(y\frac{\pi}{2})$ achieves visually more pleasing results, $f(y) = y^2$ outperforms all other LLIE methods on face detection. Overall our results show that AC adjustment is a simple and efficient pre-processing method for boosting detection accuracy on low-light images which outperforms more sophisticated techniques.

4.4 Run-time Performance Evaluation

All our experiments were performed on an consumer PC using Microsoft Windows 10 as operating system, with a 2.2 GHz (Intel i7) CPU, 16 GB of RAM, and a Nvidia GTX 1050 Ti graphics card with 4 GB VRAM. Our full algorithm, implemented with C++ and CUDA (v10.0), runs at real-time for VGA resolution images (Tab. 3) and at interactive frame rates on HD and FHD resolution images. Unlike ours, most of the existing

Table 3: Run-time performance of various methods in milliseconds. The top three run-time performance values for each resolution are shown in red, blue, and brown colors respectively. Note, that all learning-based methods except LIME and SRIE make use of the GPU. We were not able to run certain methods at higher resolutions due to Out-of-Memory (OOM) exceptions.

Method	VGA 640 × 480	HD 1280 × 720	FHD 1920 × 1080	QHD 2560 × 1440
LIME	580	1940	6450	**10180**
SRIE	11820	49830	OOM	OOM
MBLLEN	430	1300	3010	OOM
RetinexNet	1030	3710	7590	17540
LLVE	110	310	700	OOM
Zero-DCE	**4.69**	**11.77**	**25.75**	OOM
Ours SE	**4.87**	**12.91**	**28.59**	**49.49**
Ours ME	**59.58**	**180**	**410**	**740**

techniques are either not able to handle QHD resolution or are significantly slower for the given hardware configuration. Excluding the SE setting, our ME version performs better than all the other methods except Zero-DCE [16]. While AC forms the basis of our approach, more than 90% of the processing time is spent on multi-pyramid based exposure fusion. For the SE setting, the result obtained has artifacts in the form of over-exposedness and color-shifts, however, provides a reasonable approximation for the enhanced image. Thus, the SE version, our fast variant, can potentially serve as a preview of the enhanced output and allow for further interactive parameter editing.

5 DISCUSSION

Most of the existing methods, including ours, face three major challenges for LLIE. First is the trade-off between under- and over-exposedness. In order to expose the low-lit regions within an image, one might over-expose existing well-exposed parts. We approached the above to a large extent via adaptive computation of chromaticity and further by making use of an exposure sequence and multi-pyramid based blending. As a generic approach, one can compute the degree of exposure for different image regions, as an exposure mask, in a pre-processing step and use it for further processing. Second is the introduction and amplification of noise while enhancing images. To prevent the above, we first decompose the image into *base* and *detail* layers. However, a more sophisticated denoising scheme specifically tailored for low-light noise might perform better for this purpose. Thirdly, the enhancement process can result in changes in perceived color. In our approach, such changes are limited due to the counter-balancing effect of α and γ on the perceived colorfulness.

(a) Input (b) Ours (c) Ours + Denoising (d) MBLLEN (e) SRIE

Figure 12: Our result can further be improved by a post-processing denoising operation. Here, we compare our denoised-output (denoising done using FFDNET [58]) with that of MBLLEN [34] and SRIE [31].

Limitations: In order to tackle the issue of noise most of the existing techniques either employ denoising priors in their objective formulation [31], perform denoising as a post-processing operation [17], or introduce synthetic noise during training [34, 57]. We do not include any explicit denoising step in our methodology and still perform better both qualitatively and quantitatively. However, among the possible challenges in low-light image enhancement we are less effective in terms of noise-removal. The above is reflected to a certain degree during the user study where we observed that on certain occasions participants preferred the method of Lv *et al.* [34] and Li *et al.* [31] due to their less-noisy results. We conjecture that this preference can be shifted in our favor by performing a post-processing denoising operation. Note, that our denoised output in Fig. 12c has better quality and does not suffer from artifacts such as over-exposure (as in Fig. 12d) or color-shifts (as in Fig. 12e).

6 CONCLUSIONS & FUTURE WORK

This paper presents a simple yet effective technique to enhance low-light images and videos. The key to our approach is Adaptive Chromaticity that allows to efficiently increase the image brightness. Our SE version runs at real-time frame rates and can be used for a fast enhancement preview. To further improve results, we generate a virtual exposure sequence by computing multiple adaptive chromaticities for the *base* layer followed by a multi-pyramid based fusion. Our ME version runs at interactive frame rates, even for high resolution images. Experimental results validate the advancement of our approach in comparison to various state-of-the-art alternatives. For the above, we perform both quantitative and qualitative evaluation including subjective user studies. As part of future work we plan to include a denoising step in our algorithm and potentially use the multi-scale nature of exposure-fusion for this purpose. For videos, we plan to use the neighboring frames to improve the denoising as well as enhancement quality.

7 ACKNOWLEDGMENTS

We thank the participants who took part in the user study. This work was partially funded by the German Federal Ministry of Education and Research

Algorithm 1: Our Low-light Image Enhancement Algorithm

Input: Input image I, Bilateral Filter parametrs σ_s, σ_t, Adaptivity parameters $\alpha_1, \ldots, \alpha_N$, Gamma correction parameters $\gamma_1, \ldots, \gamma_N$, Exposure fusion parameters $\sigma, \upsilon_c, \upsilon_s, \upsilon_e$, Exposure levels – N, Pyramid levels – M, Additive parameter η

Output: Enhanced output image O

$B \leftarrow$ `BilateralFilter`(I, σ_s, σ_t) `// Base Layer`
$D \leftarrow I - B$ `// Detail Layer`
$wtSum \leftarrow 0$
for $k \leq 1$ *to* N **do**
  $E_k \leftarrow A_c(B, \alpha_k, \gamma_k)$ `// Generate exposure series`
  $w_k \leftarrow$ `ComputeWeights`$(E_k, \sigma, \upsilon_c, \upsilon_s, \upsilon_e)$ `// Eq. 5`
  $wtSum \leftarrow wtSum + w_k$
$outerSum \leftarrow 0$
for $k \leq 1$ *to* N **do**
  innerSum $\leftarrow 0$
  $\widehat{w_k} \leftarrow w_k / wtSum$
  $tmp1 \leftarrow E_k$
  $G(\widehat{w_k})_l \leftarrow \widehat{w_k}$
  for $l \leq 1$ *to* M **do**
    $tmp2 \leftarrow$ `GaussianFilter`$(tmp1, \sigma = l)$ `// "l" is the Gaussian Filter kernel width`
    $L(E_k)_l \leftarrow tmp1 - tmp2$ `// Laplacian pyramid of Base exposure levels`
    $G(\widehat{w_k})_l \leftarrow$ `GaussianFilter`$(G(\widehat{w_k})_l, \sigma = l)$
    $innerSum \leftarrow innerSum + G(\widehat{w_k})_l \cdot L(E_k)_l$
    $tmp1 \leftarrow tmp2$
  $innerSum \leftarrow innerSum + G(\widehat{w_k})_l \cdot tmp2$
  $outerSum \leftarrow outerSum + innerSum$
$B_E \leftarrow outerSum$ `// Enhanced Base Layer`
$O \leftarrow B_E + \eta D$ `// Enhanced Output Image`

(BMBF) through grants 01IS18092 ("mdViPro") and 01IS19006 ("KI-LAB-ITSE") and the Research School on "Service-Oriented Systems Engineering" of the Hasso Plattner Institute.

REFERENCES

[1] M. Abdullah-Al-Wadud et al. "A Dynamic Histogram Equalization for Image Contrast Enhancement". In: *IEEE Transactions on Consumer Electronics* 53.2 (2007), pp. 593–600. DOI: 10.1109/TCE.2007.381734.

[2] Soonmin Bae, Sylvain Paris, and Frédo Durand. "Two-Scale Tone Management for Photographic Look". In: *ACM SIGGRAPH 2006 Papers*. SIGGRAPH '06. 2006, pp. 637–645. DOI: 10.1145/1179352.1141935.

[3] Nicolas Bonneel et al. "Blind Video Temporal Consistency". In: *ACM Trans. Graph.* 34.6 (2015). ISSN: 0730-0301. DOI: 10.1145/2816795.2818107.

[4] Nicolas Bonneel et al. "Intrinsic Decompositions for Image Editing". In: *Computer Graphics Forum* 36.2 (May 2017), pp. 593–609. ISSN: 0167-7055. DOI: 10.1111/cgf.13149.

[5] Bolun Cai et al. "A Joint Intrinsic-Extrinsic Prior Model for Retinex". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 4020–4029. DOI: 10.1109/ICCV.2017.431.

[6] Jianrui Cai, Shuhang Gu, and Lei Zhang. "Learning a Deep Single Image Contrast Enhancer from Multi-Exposure Images". In: *IEEE Transactions on Image Processing* 27.4 (2018), pp. 2049–2062. DOI: 10.1109/TIP.2018.2794218.

[7] Turgay Celik and Tardi Tjahjadi. "Contextual and Variational Contrast Enhancement". In: *IEEE Transactions on Image Processing* 20.12 (2011), pp. 3431–3441. DOI: 10.1109/TIP.2011.2157513.

[8] Chen Chen et al. "Learning to See in the Dark". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3291–3300. DOI: 10.1109/CVPR.2018.00347.

[9] Chen Chen et al. "Seeing Motion in the Dark". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 3184–3193. DOI: 10.1109/ICCV.2019.00328.

[10] H.D. Cheng and X.J. Shi. "A simple and effective histogram equalization approach to image enhancement". In: *Digital Signal Processing* 14.2 (2004), pp. 158–170. ISSN: 1051-2004. DOI: https://doi.org/10.1016/j.dsp.2003.07.002.

[11] skimage v0.19.0 docs. *estimate_sigma*. 2022. URL: https://scikit-image.org/docs/stable/api/skimage.restoration.html#skimage.restoration.estimate_sigma (visited on 01/27/2022).

[12] David L Donoho and Iain M Johnstone. "Ideal spatial adaptation by wavelet shrinkage". In: *Biometrika* 81.3 (1994), pp. 425–455. ISSN: 0006-3444. DOI: 10.1093/biomet/81.3.425.

[13] Gang Fu, Lian Duan, and Chunxia Xiao. "A Hybrid L2 -Lp Variational Model For Single Low-Light Image Enhancement With Bright Channel Prior". In: *2019 IEEE International Conference on Image Processing (ICIP)*. 2019, pp. 1925–1929. DOI: 10.1109/ICIP.2019.8803197.

[14] Xueyang Fu et al. "A Probabilistic Method for Image Enhancement With Simultaneous Illumination and Reflectance Estimation". In: *IEEE Transactions on Image Processing* 24.12 (2015), pp. 4965–4977. DOI: 10.1109/TIP.2015.2474701.

[15] Xueyang Fu et al. "A Weighted Variational Model for Simultaneous Reflectance and Illumination Estimation". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2782–2790. DOI: 10.1109/CVPR.2016.304.

[16] Chunle Guo et al. "Zero-DCE: Zero-Reference Deep Curve Estimation for Low-Light Image Enhancement". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 1777–1786. DOI: 10.1109/CVPR42600.2020.00185.

[17] Xiaojie Guo, Yu Li, and Haibin Ling. "LIME: Low-Light Image Enhancement via Illumination Map Estimation". In: *IEEE Transactions on Image Processing* 26.2 (2017), pp. 982–993. DOI: 10.1109/TIP.2016.2639450.

[18] Shijie Hao et al. "Low-Light Image Enhancement With Semi-Decoupled Decomposition". In: *IEEE Transactions on Multimedia* 22.12 (2020), pp. 3025–3038. DOI: 10.1109/TMM.2020.2969790.

[19] David Hasler and Sabine E. Suesstrunk. "Measuring colorfulness in natural images". In: *Human Vision and Electronic Imaging VIII*. Vol. 5007. International Society for Optics and Photonics. 2003, pp. 87 –95. DOI: 10.1117/12.477378.

[20] Haiyang Jiang and Yinqiang Zheng. "Learning to See Moving Objects in the Dark". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 7323–7332. DOI: 10.1109/ICCV.2019.00742.

[21] Yifan Jiang et al. "EnlightenGAN: Deep Light Enhancement Without Paired Supervision". In: *IEEE Trans. Image Process.* 30 (2021), pp. 2340–2349. DOI: 10.1109/TIP.2021.3051462.

[22] D.J. Jobson, Z. Rahman, and G.A. Woodell. "A multiscale retinex for bridging the gap between color images and the human observation of scenes". In: *IEEE Transactions on Image Processing* 6.7 (1997), pp. 965–976. DOI: 10.1109/83.597272.

[23] D.J. Jobson, Z. Rahman, and G.A. Woodell. "Properties and performance of a center/surround retinex". In: *IEEE Transactions on Image Processing* 6.3 (1997), pp. 451–462. DOI: 10.1109/83.557356.

[24] Wei-Sheng Lai et al. "Learning Blind Video Temporal Consistency". In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. 2018, pp. 179–195. DOI: 10.1007/978-3-030-01267-0_11.

[25] Edwin H Land and John J McCann. "Lightness and retinex theory". In: *Journal of the Optical Society of America* 61.1 (1971), pp. 1–11. DOI: 10.1364/JOSA.61.000001.

[26] Chulwoo Lee, Chul Lee, and Chang-Su Kim. "Contrast Enhancement Based on Layered Difference Representation of 2D Histograms". In: *IEEE Transactions on Image Processing* 22.12 (2013), pp. 5372–5384. DOI: 10.1109/TIP.2013.2284059.

[27] Hunsang Lee, Kwanghoon Sohn, and Dongbo Min. "Unsupervised Low-Light Image Enhancement Using Bright Channel Prior". In: *IEEE Signal Processing Letters* 27 (2020), pp. 251–255. DOI: 10.1109/LSP.2020.2965824.

[28] Chongyi Li et al. "Low-Light Image and Video Enhancement Using Deep Learning: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), pp. 1–1. DOI: 10.1109/TPAMI.2021.3126387.

[29] Jian Li and Yabiao Wang. *Tencent/FaceDetection-DSFD*. 2019. URL: https://github.com/Tencent/FaceDetection-DSFD.

[30] Jian Li et al. "DSFD: dual shot face detector". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5060–5069. DOI: 10.1109/CVPR.2019.00520.

[31] Mading Li et al. "SRIE: Structure-Revealing Low-Light Image Enhancement Via Robust Retinex Model". In: *IEEE Transactions on Image Processing* 27.6 (2018), pp. 2828–2841. DOI: 10.1109/TIP.2018.2810539.

[32] Jiaying Liu et al. "Benchmarking Low-Light Image Enhancement and Beyond". In: *International Journal of Computer Vision* 129.4 (2021), pp. 1153–1184. ISSN: 1573-1405. DOI: 10.1007/s11263-020-01418-8.

[33] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. "LLNet: A deep autoencoder approach to natural low-light image enhancement". In: *Pattern Recognition* 61 (2017), pp. 650–662. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2016.06.008.

[34] Feifan Lv et al. "MBLLEN: Low-Light Image/Video Enhancement Using CNNs". In: *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 2018, p. 220. URL: http://bmvc2018.org/contents/papers/0700.pdf.

[35] T. Mertens, J. Kautz, and F. Van Reeth. "Exposure Fusion: A Simple and Practical Alternative to High Dynamic Range Photography". In: *Computer Graphics Forum* 28.1 (2009), pp. 161–171. DOI: https://doi.org/10.1111/j.1467-8659.2008.01171.x.

[36] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. "Making a Completely Blind Image Quality Analyzer". In: *IEEE Signal Processing Letters* 20.3 (2013), pp. 209–212. DOI: 10.1109/LSP.2012.2227726.

[37] Franck Neycenssac. "Contrast Enhancement Using the Laplacian-of-a-Gaussian Filter". In: *CVGIP: Graph. Models Image Process.* 55.6 (1993), pp. 447–463. DOI: 10.1006/cgip.1993.1034.

[38] Stephen M. Pizer et al. "Adaptive histogram equalization and its variations". In: *Computer Vision, Graphics, and Image Processing* 39.3 (1987), pp. 355–368. ISSN: 0734-189X. DOI: https://doi.org/10.1016/S0734-189X(87)80186-X.

[39] Erik Reinhard et al. *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.

[40] Wenqi Ren et al. "Low-Light Image Enhancement via a Deep Hybrid Network". In: *IEEE Transactions on Image Processing* 28.9 (2019), pp. 4364–4375. DOI: 10.1109/TIP.2019.2910412.

[41] Xutong Ren et al. "LR3M: Robust Low-Light Enhancement via Low-Rank Regularized Retinex Model". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 5862–5876. DOI: 10.1109/TIP.2020.2984098.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. 2015, pp. 234–241. ISBN: 978-3-319-24574-4. DOI: 10.1007/978-3-319-24574-4_28.

[43] Sumit Shekhar et al. "Consistent Filtering of Videos and Dense Light-Fields Without Optic-Flow". In: *Vision, Modeling and Visualization*. 2019. DOI: 10.2312/vmv.20191326.

[44] C. Tomasi and R. Manduchi. "Bilateral filtering for gray and color images". In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, pp. 839–846. DOI: 10.1109/ICCV.1998.710815.

[45] Danai Triantafyllidou et al. "Low Light Video Enhancement Using Synthetic Data Produced with an Intermediate Domain Mapping". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. 2020, pp. 103–119. ISBN: 978-3-030-58601-0. DOI: 10.1007/978-3-030-58601-0_7.

[46] Vasileios Vonikakis. *Busting image enhancement and tone-mapping algorithms: A collection of the most challenging cases*. 2022. URL: https://sites.google.com/site/vonikakis/datasets (visited on 01/27/2022).

[47] Ruixing Wang et al. "Underexposed Photo Enhancement Using Deep Illumination Estimation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 6842–6850. DOI: 10.1109/CVPR.2019.00701.

[48] Shuhang Wang et al. "Naturalness Preserved Enhancement Algorithm for Non-Uniform Illumination Images". In: *IEEE Transactions on Image Processing* 22.9 (2013), pp. 3538–3548. DOI: 10.1109/TIP.2013.2261309.

[49] Chen Wei et al. "RetinexNet: Deep Retinex Decomposition for Low-Light Enhancement". In: *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 2018, p. 155. URL: http://bmvc2018.org/contents/papers/0451.pdf.

[50] Dejia Xu. *Dark Face Eval Tool*. 2019. URL: https://github.com/Ir1d/DARKFACE_eval_tools.

[51] Ke Xu et al. "Learning to Restore Low-Light Images via Decomposition-and-Enhancement". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2278–2287. DOI: 10.1109/CVPR42600.2020.00235.

[52] Shuo Yang et al. "WIDER FACE: A Face Detection Benchmark". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 5525–5533. DOI: 10.1109/CVPR.2016.596.

[53] Wenhan Yang et al. "Advancing Image Understanding in Poor Visibility Environments: A Collective Benchmark Study". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 5737–5752. DOI: 10.1109/TIP.2020.2981922.

[54] Wenhan Yang et al. "From Fidelity to Perceptual Quality: A Semi-Supervised Approach for Low-Light Image Enhancement". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 3060–3069. DOI: 10.1109/CVPR42600.2020.00313.

[55] Zhenqiang Ying, Ge Li, and Wen Gao. "A bio-inspired multi-exposure fusion framework for low-light image enhancement". In: *arXiv preprint arXiv:1711.00591* (2017). URL: https://arxiv.org/pdf/1711.00591.pdf.

[56] Runsheng Yu et al. "DeepExposure: Learning to Expose Photos with Asynchronously Reinforced Adversarial Learning". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/a5e0ff62be0b08456fc7f1e88812af3d-Paper.pdf.

[57] Fan Zhang et al. "LLVE: Learning Temporal Consistency for Low Light Video Enhancement From Single Images". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 4967–

4976. DOI: 10.1109/CVPR46437.2021.00493.

[58] Kai Zhang, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising". In: *IEEE Transactions on Image Processing* 27.9 (2018), pp. 4608–4622. DOI: 10.1109/TIP.2018.2839891.

[59] Qing Zhang, Yongwei Nie, and Wei-Shi Zheng. "Dual Illumination Estimation for Robust Exposure Correction". In: *Computer Graphics Forum* 38.7 (2019), pp. 243–252. DOI: 10.1111/cgf.13833.

[60] Qing Zhang et al. "High-Quality Exposure Correction of Underexposed Photos". In: *Proceedings of the 26th ACM International Conference on Multimedia*. MM '18. 2018, pp. 582–590. ISBN: 9781450356657. DOI: 10.1145/3240508.3240595.

[61] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. "Kindling the Darkness: A Practical Low-Light Image Enhancer". In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France, 2019, pp. 1632–1640. DOI: 10.1145/3343031.3350926.

[62] Chaobing Zheng et al. "Single image brightening via multi-scale exposure fusion with hybrid learning". In: *IEEE Transactions on Circuits and Systems for Video Technology* 31.4 (2020), pp. 1425–1435. URL: https://arxiv.org/pdf/2007.02042.pdf.

[63] Minfeng Zhu et al. "EEMEFN: Low-Light Image Enhancement via Edge-Enhanced Multi-Exposure Fusion Network". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.07 (2020), pp. 13106–13113. DOI: 10.1609/aaai.v34i07.7013.

ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.31, No-1-2, 2023

Visual Exploration of Repetitive Patterns on Ancient Peruvian Pottery

Stefan Lengauer[1], Lin Shao[1,2], Magdalena Mayerhofer[1], Reinhold Preiner[1], Stephan Karl[3], Elisabeth Trinkl[3], Ivan Sipiran[4], Benjamin Bustos[4], and Tobias Schreck[1]

[1]Graz University of Technology, Institute CGV, Inffeldgasse 16c, 8010 Graz, Austria
[2]Fraunhofer Austria Center for Data Driven Design, Inffeldgasse 16c, 8010 Graz, Austria
[3]University of Graz, Institute of Classics, Universitätsplatz 3/II, 8010 Graz, Austria
[4]University of Chile, Dept. of Computer Science, Beauchef 851, Santiago, Chile

ABSTRACT

The analysis and understanding of artifact properties and their relationships is a key objective in the archaeological analysis of cultural heritage objects. There are many aspects of concern, including both shape properties of the objects as a whole and appearances stemming from paintings and ornamentations on the object surfaces. To date, experts consider those mostly holistically on a per-object basis. We present an approach for the interactive visual exploration and correlation of shape- and ornament-based properties of a large collection of ancient vessels. Our approach allows us to group objects by said properties, and to relate them in side-by-side and bipartite graph displays. To this end, we define an encompassing set of feature descriptors, which are leveraged to cluster the objects by user-selected properties. Case studies show that a comparative overview of all objects effectively supports the discovery of interesting co-occurrences of shapes and ornaments. This way, our tool opens new possibilities for the domain analysis of cultural heritage object collections by data-driven visual exploration.

Keywords

Visual exploration, Pattern analysis, Pattern descriptors

1 INTRODUCTION

The analysis of ancient pottery is an essential task for the understanding of ancient cultures and lifestyles. Of particular interest are the lavish surface decorations exhibited by the majority of pottery artifacts. These decorations – so called *vase paintings* – comprise both repetitive ornaments and motifs depicting mythological scenes. They provide important information for an artifact's attribution to a specific epoch, culture, workshop of even painter [ES16].

The concurrent exploration a large collection allows us to reveal clusters of objects with common traits, if these objects are appropriately arranged in a structured manner, based on relevant properties. Potential properties comprise intrinsic traits like shape, material, capacity and such, such as well as derived traits like culture, dating, etc. However, also more complex traits describing an object's vase painting, e.g., variability, distribution and positioning on the surface or colorization, can be extracted automatically using customized data processing techniques.

Within the scope of this paper, we focus on two of the most important properties: (i) object shape and (ii) vase painting. We present a novel visualization method that groups similar elements along these properties and presents the resulting groups in separate cluster visualizations (Fig. 1). These are connected in a bipartite graph, revealing relationships and co-occurrences between different shapes and paintings. The thickness of graph edges between shape and painting clusters reveal inter-cluster correlations in the collection. This visualization is the core component of an overarching interactive exploration system, supporting various degrees of visual granularity – from a broad overview down to closeup. To this end, we conceptualize and implement tailored views for the individual levels with customized object previews for different properties (e.g., positional glyphs in Fig. 1, bottom, indicating the positioning of patterns on the object surface).

The contribution of our paper is the interactive visualization concept which we evaluate using prototype implementation together with a real-world dataset of ancient pottery objects. Moreover, we present a novel feature descriptor which is able to capture the arrangement of repetitive patterns (e.g., regarding regularity) in a quantifiable manner.

In the following, we report related visualization techniques (Sec. 2), before defining the domain analysis task (Sec. 3) relevant for experts. In Sec. 4 we discuss the proposed concept in detail, before we present the major insights gained with this tool in Sec. 5. We conclude the paper with a discussion (Sec. 6) including feedback from archaeologists.

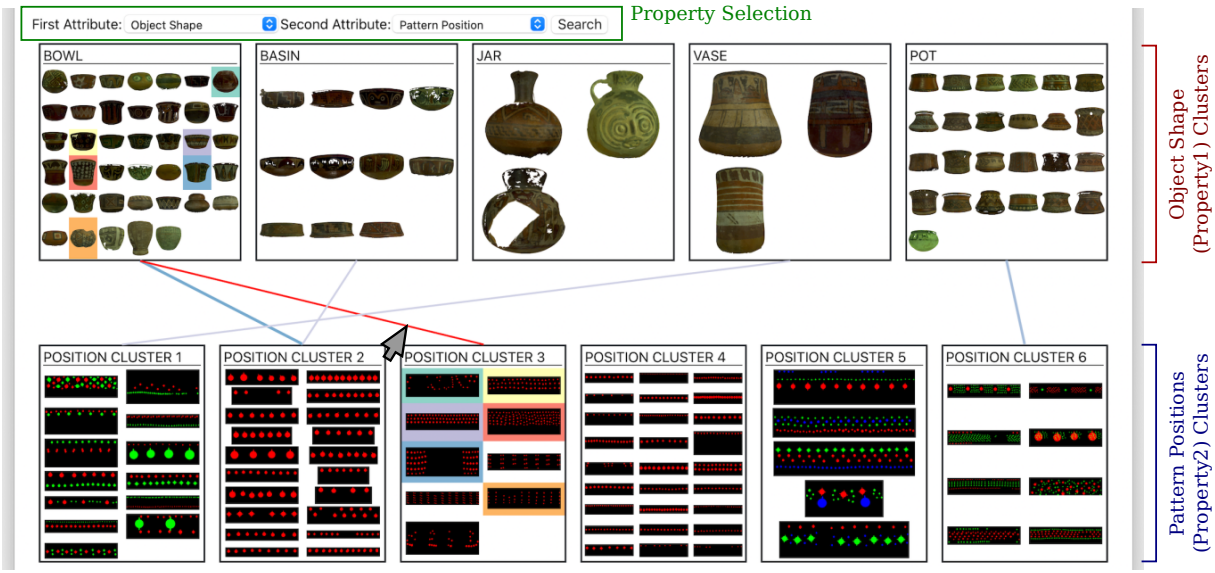


Figure 1: A collection is explored along two selected properties, i.e., object shape and pattern position, which are clustered individually (top and bottom row). Visual links in-between reveal correlations inter-cluster correlations.

2 RELATED WORK

One goal of Information Visualization (InfoVis) techniques is to convert abstract information into visual representations, and thus gain knowledge about internal structures of a dataset. In cases where corresponding data elements have inherent relationships among each other, graph-based visualization techniques are commonly used. A graph visualization is often encoded by a set of nodes and edges and allows the visual analysis of structures and grouping of nodes and/or edges (Compound Graph Visualization) [HSS15]. Further application areas and examples of graph-based visualizations are given in surveys [vLKS*11; TKE12].

Visualization techniques have also become important tools for the research of cultural heritage (CH) objects. Although many scientific visualization techniques in the CH domain focus on the realistic rendering of 3D objects, there is a growing number of interactive visual systems for analyzing CH data [WFS*19]. In recent works, systems and interface designs were introduced that utilize InfoVis designs and Visual Analytics approaches for representing multidimensional and temporal information of CH collections. For instance, the PolyCube framework by Windhager *et al.* [WSL*20] uses space-time cube representations to visualize multidimensional, time-dependent properties of collections. By connecting different visualization techniques, like map, set, and network visualizations, they revealed spatial, categorical, and relational collection aspects. Simon *et al.* [SIBdS16] introduce Peripleo, an open source tool to explore the geographic, temporal and thematic composition of distributed digital collections. Lengauer *et al.* [LKK*20] present an interactive visual exploration system for artifact collections, dubbed Linked Views Visual Exploration System (LVVES), that supports task-oriented analysis and exploration along temporal, spatial, and shape modalities.

To visualize sets of images, the overall layout for arranging the images is often a crucial part. In this regard, Brivio *et al.* [BTC10] propose a Voronoi-based layout to visualize photographic campaigns in CH. In Glinka *et al.* [GPD17] the authors show the potential of details-on-demand techniques for the exploration of large CH collections including images, keywords and textual data. They employ a zoomable timeline visualization to link a "distant-viewing" and "close-viewing" mode for the exploration.

In Mauri *et al.* [MPCC13] a graph of actors and projects is used to explore collaborations between architects, while Tortora *et al.* [DPT*12] use graph views to support archaeologist in finding new correlations from ontology-driven metadata. An extensive survey on visualization techniques for CH collection data is given in [WFS*19].

As opposed to existing approaches, our proposed design should be particularly useful in revealing correlations between different object properties. To this end, we use a bipartite graph layout, connecting different similarity clusterings – a design which has, to our knowledge, not been used before. Moreover, we provide two details-on-demand feature: a side-by-side view for comparing two clusters from different traits, and a closeup view for additional information on the lowest level of visual granularity.

3 DOMAIN ANALYSIS TASKS

The prototype system is designed to be used by domain experts having well-defined research questions. W.r.t.

the analysis of repetitive patterns on ancient pottery those include (but are not limited to): (1) How regular are the ornament patterns within a pattern class? (2) Do similar ornament patterns exhibit similar spatial arrangements? (3) Are ornament patterns correlated with the vessel shape? (4) Are properties of ornament pattern or other shape properties generally correlated with each other?

To date, such questions are mostly answered using pairwise visual comparisons of artifacts. Based on this established workflow, we define the following domain analysis tasks, for which we support a domain expert with customized visualizations: (**T1**) Select two properties and discover inter-trait correlations, (**T2**) Show the detailed correlations between two selected clusters, and (**T3**) Show the properties of a single record.

4 CONCEPT

In the following, our proposed design is discussed in depth. After a broad overview of the idea (Sec. 4.1), we present the dataset used in our experiments (Sec. 4.2). In Sec. 4.3, we give a detailed formal description of our descriptors designed for describing pattern arrangements, while Sec. 4.4 discusses the employed shape features for describing pattern shapes. Sec. 4.5 concludes the section with a description of the applied clustering as well as the similarity computation between clusters.

4.1 Overview

As the starting point for the visual exploration process, we provide the user with a bipartite cluster view, for which the user has to select two object properties via respective drop-down dialogues (Fig. 1, green). The selected properties are used to cluster the objects of the given collection separately, e.g., by 'vessel shape' and 'pattern positioning' (Fig. 1, red and blue). Based on the visual links between the two sets of clusters, a user can derive the presence and strength of possible correlations, e.g., between BOWL shapes and POSITION CLUSTER 2 and 3 w.r.t. Fig. 1, and can further investigate which objects of a pair of clusters is responsible for a correlation by hovering the mouse over the respective inter-cluster link. Clicking on such a link switches the visualization to a side-by-side view of the respective clusters (Fig. 2) and clicking an item in any of the clusters switches to a closeup view (Fig. 3) of the object in question. Return buttons allow one to undo such a transition and allow for a continuous exploration process. Design details on these different views and information on how they support the different domain analysis tasks (**T1**–**T3**) are given in Sec. 4.6.

4.2 Dataset: Peruvian vessels

The dataset we use for our experiments stems from the 2021 SHREC track on "Retrieval of cultural heritage objects" by Sipiran *et al.* [SLL*21], containing almost 1,000 3D models of ancient pottery artifacts. The real artifacts are kept in the Josefina Ramos de Cox museum in Lima, Perú, where they were digitized as part of a research project[1]. The collection comprises objects from several pre-Columbian cultures, like Chancay, Lurin or Nazca, featuring varied geometry and surface decoration. In a later documentation effort, the boundaries of ornament elements were annotated and all occurring surface patterns were grouped for a subset of the collection, exhibiting well preserved and lavish vase paintings. This annotation by Lengauer *et al.* [LSP*21] is publicly available[2] and contains detailed outlines of all surface patterns, together with pattern-wise properties like orientation, scale, position on the surface, and n-foldness, giving an encompassing data basis for an attribute-driven exploration. In total, the dataset comprises 2,529 *pattern entities* from 82 textured models, which are grouped into 102 distinct similarity classes (referred to by *pattern archetypes*).

The dataset also comprises a varied collection of intrinsic and derived traits, such as vessel shape, pattern variability, colorization, etc., which we need to describe quantitatively in order to cluster the objects by them. The properties are defined at three different levels of detail: (1) *Per-object* properties were provided by experts via a categorization of the vessel shape (i.e., 'bowl', 'basin', 'jar', 'vase' and 'pot'). Surface colorization is also described on a per-object level through the computation of color histograms. (2) *Per-archetype* properties comprise several carefully designed attributes describing the distribution of entities across the surface and the relations among themselves (Sec. 4.3). On a (3) *per-entity* level, we use an abstract description of a pattern entity's shape through established shape features (Sec. 4.4).

4.3 Quantifying Pattern Arrangements

We design a set of custom properties pertaining to the distribution, regularity, overall variability and other important traits of a pattern archetype. All properties are given as scalar, normalized to the range $[0,1)$, so that they can be combined into a feature vector. Specifically, we define and compute the following measures:

Occurrence Frequency. This value describes a pattern archetype's quantity of entities and is given by

$$\tilde{n}_p = \left(\frac{n_p - n_{p_{min}}}{n_p} \right)^{\alpha}, \qquad (1)$$

[1] Project 02-2018-FONDECYT-BM-IADT-AV (Concytec-Perú): Restoration and conservation of archeological pieces using deep learning and convolutional auto-encoder on graphs.

[2] https://datasets.cgv.tugraz.at/pattern-benchmark/ [Accessed 2023-04-26]

with n_p as the absolute number of pattern entities, $n_{p_{min}} = 2$ as the minimal number of entities for an archetype, and $\alpha = 3$ denoting an empirically determined non-linear scaling factor. Note that $\tilde{n}_p$ asymptotically approaches 1 for $n_p \to \infty$ as there is no theoretical limit for the number of entities.

Fold Symmetry. The patterns' n-fold symmetry (the number of symmetry planes) is already provided by the annotated dataset and is, similar to the **Occurrence Frequency**, normalized to $[0,1)$.

Scale Variability. This property measures the variation of a pattern's size among its entities. To this end, a Normal distribution $\mathcal{N}(\mu,\sigma^2)$ is fitted to the entity scales $S = \{s_i\}_{i\in I}$, with $I = [1\,..\,n_p]$ as the index set of a pattern archetype's entities. S is also provided by the given annotation. Similar to the **Occurrence Frequency**, the variance σ^2 is non-linearly normalized to $[0,1)$ by $\tilde{n}_\sigma = \sigma^2/(\sigma^2+1)$.

Regularity. This property aims to capture the amount of 'structure' exhibited by the patterns' distribution on the surface. More specifically, this metric ranks the distribution from completely random (i.e., no perceivable structure at all, like in Fig. 3) to perfectly aligned in a grid-like manner (e.g., green pattern in Fig. 4, top). To this end, we detect the presence of rows and column arrangements in the set of unordered pattern entities. They are determined based solely on the patterns' center points $C = \{\langle x_i, y_i\rangle\}_{i\in I} \subset \mathbb{R}^2$. Rows and columns are obtained independently by fitting *Gaussian Mixture Models* (GMM) with an optimal number n of components to the x and y components of C, respectively. n is determined via the *Bayesian Information Criterion* (BIC) [BK10], measuring how well a particular GMM models the given data. To this end, we evaluate the BIC for $n \in [1\,..\,\lfloor |C|/n_{min}\rfloor]$, with $n_{min} = 3$ as we require a row or column to feature at least three pattern entities (fewer elements do not reflect a domain expert's notion of rows or columns). If the BIC exceeds an empirically defined threshold, it is assumed no rows or columns are present in the pattern layout.

We assume that the regularity of a distribution increases with its number of rows and columns (normalized quantity $\tilde{m}$) and decreases with the pattern entities offsets from their respective rows and columns (normalized error $\tilde{\varepsilon}$). Based on these assumptions, we define the three tiers of regularity of a distribution: *(i)* 'has rows and columns', *(ii)* 'has either rows or columns', and *(iii)* 'has no identifiable arrangement'. Representatives of these tiers are normalized within the intervals $[0,\frac{1}{3})$, $[\frac{1}{3},\frac{2}{3})$ and $[\frac{2}{3},1)$ respectively. Let $Y = \{y_k\}_{k\in K} \subset \mathbb{R}_+$ be the y-positions of rows K and $X = \{x_l\}_{l\in L} \subset \mathbb{R}_+$ be the x-positions of columns L.

The normalized regularity $\tilde{r}$ is given by

$$\tilde{r} = \frac{1}{3}\begin{cases} \frac{1}{1+\frac{\tilde{\varepsilon}}{\tilde{m}}} + 2 & |X|>0 \text{ and } |Y|>0 \\ \frac{1}{1+\frac{\tilde{\varepsilon}}{\tilde{m}}} + 1 & |X|>0 \text{ or } |Y|>0 \\ \frac{n_p}{n_p+10} & \text{otherwise.} \end{cases} \tag{2}$$

The normalized quantity $\tilde{m}$ we define as

$$\tilde{m} = \begin{cases} \frac{|X|+|Y|}{|X|+|Y|+10} & |X|>0 \text{ and } |Y|>0 \\ \frac{|X|}{|X|+1} & |X|>0 \\ \frac{|Y|}{|Y|+1} & \text{otherwise,} \end{cases} \tag{3}$$

and the normalized error $\tilde{\varepsilon}$ as

$$\tilde{\varepsilon} = \frac{1}{|I|}\sum_{i\in I}\begin{cases} \left\|\langle x_i,y_i\rangle - \langle \hat{x}_i,\hat{y}\rangle\right\| & |X|>0 \text{ and } |Y|>0 \\ \|x_i-\hat{x}_i\| & |X|>0 \\ \|x_i-\hat{y}_i\| & \text{otherwise,} \end{cases} \tag{4}$$

with $\hat{x}_i = \arg\min_{x_l\in X}\|x_i - x_l\|$ and $\hat{y}_i = \arg\min_{y_k\in Y}\|y_i - y_k\|$ as a pattern entities offset from its designated column and row respectively.

Alternatingness. Pattern entities exhibit an orientation – or for an n-fold symmetry larger than one, even multiple equivalent orientations – provided by the annotation. With the 'alternatingness' property, we aim to quantify how a pattern's orientation deviates on average from its predecessor if an archetype's pattern entities exhibit any kind of sequence. The rationale behind this is that patterns which appear in an alternating fashion (e.g., always rotated by 180 degrees from one entity to the next) stand out from those which have a uniform orientation or those which have a completely random orientation.

Let $I_k, I_l \subset I$ denote the index sets of patterns belonging to the k-th row and l-th column, respectively. For a pattern archetype with n-fold symmetry n_f and $O = \{o_i\}_{i\in I} \subset \mathbb{R}_+$ as the orientations of its pattern entities, we define the normalized alternatingness as

$$\tilde{o} = \begin{cases} 0 & \substack{n_f=\infty \\ \text{or } |X|+|Y|=0} \\ \frac{1}{\Delta o_{max}}\left(\sum_{k\in K}\frac{\Delta_k}{|I_k|} + \sum_{l\in L}\frac{\Delta_l}{|I_l|}\right) & \text{otherwise.} \end{cases} \tag{5}$$

Here, $\Delta o = 2\pi/n_f$ denotes the maximum orientation difference between consecutive patterns with $\Delta o_{max} = \Delta o/2$. The row-wise and column-wise differences are given by $\Delta_k = \sum_{j\in[1..|I_k|]} \angle_{min}(o_{I_k[j]}, o_{I_k[j+1 \mod |I_k|]})$ and $\Delta_l = \sum_{j\in[1..|I_l|]} \angle_{min}(o_{I_l[j]}, o_{I_l[j+1 \mod |I_l|]})$, respectively, where $\angle_{min}(a,b) = \min_{i,j\in[1..n_f]}\{|a + i\Delta o - b + j\Delta o|\}$ defines the minimum angle between two ambiguous pattern orientations

The attributes for the given dataset, as well as the source code used to extract them from the annotated dataset, is available on the website hosting the pattern annotations.

4.4 Quantifying Pattern Entities' Shapes

A description of a pattern's shape is computed from the polygon describing its silhouette. To obtain a fixed-length numerical representation of this input, we employ the *Shape Context* feature descriptor by Belongie *et al.* [BMP02], as it is invariant w.r.t. all affine transformations. The underlying idea of this approach is to, first of all, extract a small sample of a contour with a roughly uniform sampling. For each of these sample points, a histogram describing the directivity and distance to all other points is computed. This description of a point by means of a histogram allows to determine the similarity between two points using the χ^2 metric. The similarity of two input shapes with the same number of sample points can then be inferred from the assignment costs of assigning all point pairs in an optimal fashion. This well-established minimization problem is referred to as square assignment problem and can be solved using, e.g., the Hungarian method [PS98]. As this kind of assignment is computationally expensive, we take a very small sample size of 20 sample points, which however is sufficient for the task at hand.

4.5 Clustering

Different clustering algorithms are applied, depending on the type of property (categorical, abstract) and level of detail (per-object, per-archetype, per-entity). For the object's shape class and the fold symmetry, no clustering algorithm is necessary as these properties are already grouped into five and four classes, respectively. For the pattern shape (Sec. 4.4) we employ a hierarchical clustering, since it has the advantage that we can provide our own distance function, which is necessary for our used feature descriptor. For all other properties (Sec. 4.3) Lloyd's K-means clustering [Llo82] is used to obtain six similarity clusters.

4.5.1 Inter-cluster Similarities

For the bipartite cluster view (Fig. 1) we also require a measure of the similarity between clusters obtained from different properties, since we want to display how strong the selection of objects between cluster-pairs varies. For the similarity between clusters, which are given as a set of objects we adopt the well known Jaccard index [Jac01], such that a small cluster with a high overlap with a large cluster still has a high similarity, to account for unevenly sized clusters. Let c_a, c_b be two clusters (sets of objects) obtained by clustering along property a and b, respectively. We define the similarity between c_a and c_b by

$$sim(c_a, c_b) = \frac{1}{2}\left(\frac{c_a \cap c_b}{c_a} + \frac{c_a \cap c_b}{c_b}\right). \qquad (6)$$

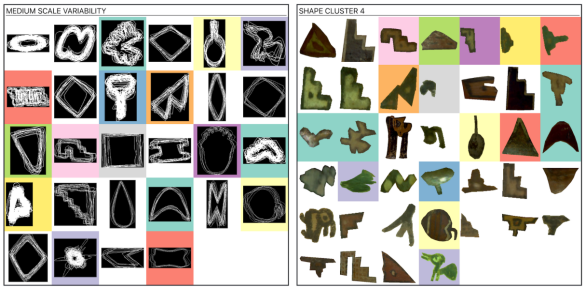


Figure 2: The side-by-side cluster view shows two clusters obtained by clustering along different object traits. The highlighting in different colors marks objects which appear in both clusters, with the same color indicating common objects.

4.6 Visual Design

Our prototype visualization system comprises three appropriate views, supporting different levels of visual granularity as well as dedicated visualizations for some of the pattern attributes. That is, the views, in descending order of visual granularity, are the following.

Bipartite Clusters Graph. This view (Fig. 1), which is the centerpiece of our exploration system, supports the discovery of correlations between traits. Hence, it is tailored to fulfill the requirements of task **T1**. From two drop-down menus at the top, users are able to select the two properties they want to compare. Based on this selection, two groups of clusters are presented in a row-wise manner. The clusters are displayed as containers, framing their belonging objects which are visualized (depending on the selected attribute) as glyphs, which are arranged in a grid-like layout. Depending on the number of objects within the cluster, the glyphs are scaled such that the space within the container is optimally used. Note, however, that the sequence in which the objects appear within a cluster is not deterministic in our current implementation. Future work might include a sorting according to inner-class similarity or other attributes. Containers also bear a label, which is a class name, e.g., 'bowl', in the case of categorical attributes like vessel shape, or an abstract term, e.g., 'position cluster n', for derived properties. In-between the two rows of clusters, we display visual links whose color saturation and thickness are relative to the cluster similarities (Sec. 4.5.1).

Additional information regarding cluster similarity is revealed through interaction. More specifically, upon hovering over a link, all objects (common across the clusters connected by the link) are highlighted. Since the appearance of the object can differ, pair-wise affiliations are established through a qualitative color coding [Bre]. From this view, a user can also transit to the side-by-side clusters view

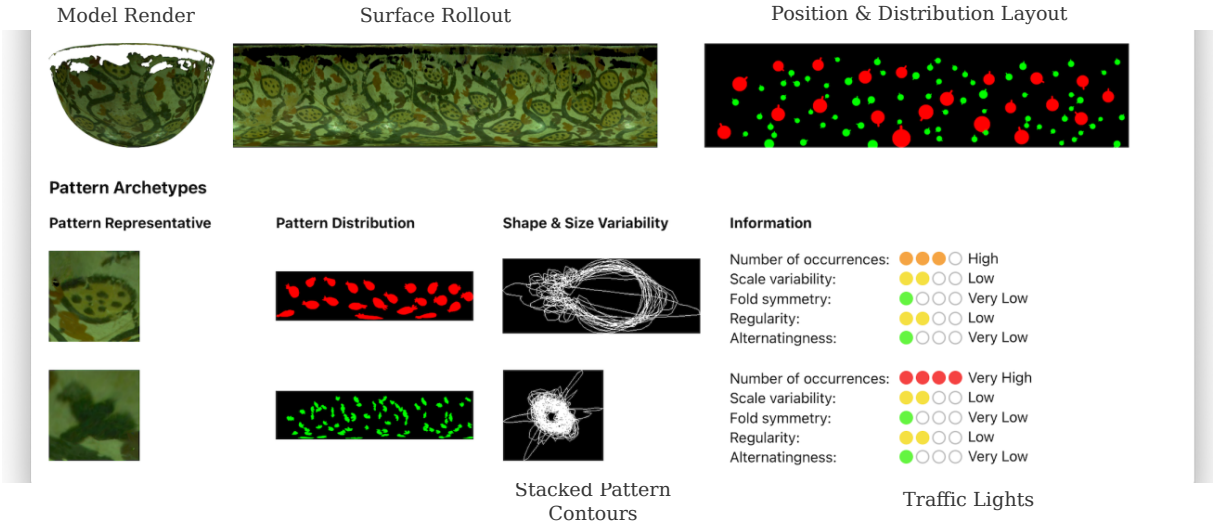


Figure 3: Closeup view of a bowl object with the various object and pattern attributes being visualized with our custom designs.

by clicking a link, or to the closeup view by clicking any of the objects within a container.

Side-by-side Clusters. The side-by-side cluster view (Fig. 2), addressing task **T2**, shows two clusters from different traits side-by-side. The container representation from the bipartite cluster graph is reused and the connection between the items is similarly established through the same color coding. From this view, a user can either go back to the bipartite clusters graph or move forward to the closeup view by clicking one of the items.

Closeup. The lowest level of visual granularity, representing just a single object, is given by the closeup view (Fig. 3). This view illustrates all the information and properties, with their respective visualizations, available for a specific object as required by task **T3**. From here, a user can return to the side-by-side cluster view or the bipartite clusters graph.

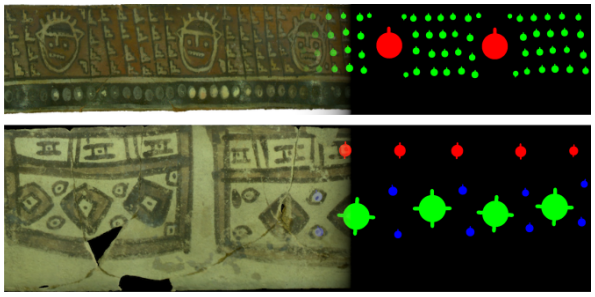


Figure 4: The projected model surfaces of two objects (left) blending over into the pattern position & distribution layout (right).

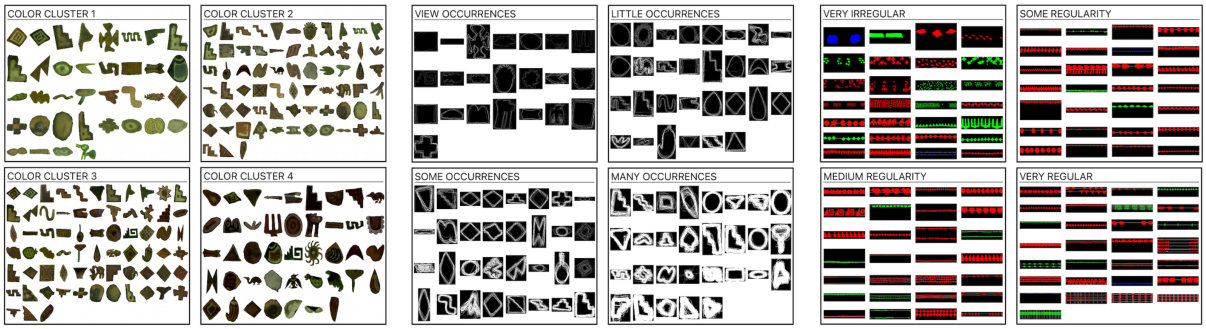
For some of the derived pattern attributes, dedicated previews are devised for displaying them in the clusters. Those comprise the following.

Model Rendering. For each 3D model, one static rendering is generated, which is used as a thumbnail image in the close-up view.

Surface Rollout. One visualization that is already given by the used dataset is a surface rollout (Fig. 4, left). Such a representation is able to visualize a model's surface as a whole and can be obtained by fitting a proxy geometry to the 3D model, which is subsequently projected, cut open, and flattened. The rollouts used in our experiments are based on a variant of the cylindrical unwrapping by Karras *et al.* [KPP96].

Position & Distribution Layout. For the regularity property we implement a glyph showing the positions, scales, and orientations of pattern entities but omitting any distracting textual information (Fig. 4, right). Starting from an empty image, with equal size to the rollout, all pattern entities are drawn as circular markers with the radii indicating their respective sizes. Color coding is used to group the patterns by their archetypes and the individual orientations are visualized by an outwards pointing straight line. Note that, in the case of an n-fold symmetry larger than one, multiple such lines are drawn for each of the equivalent orientations.

Stacked Pattern Contours. To visualize the variety of shapes belonging to one and the same pattern archetype, we conceptualize a stacked outline image comprising the silhouettes of all its pattern entities (Fig. 2). To this end, we leverage the polylines marking the outline of a pattern entity, given within the annotation. For a meaningful overlay, we rotate them in the inverse direction of their given orientation property and scale them relative to their inverse scale property. All these registered polylines



(a) Color (b) Occurrence Frequency (c) Regularity

Figure 5: Clusters resulting from clustering by the properties color (a), occurrence frequency (b) and regularity (c).

are then combined in an additive manner on an empty image.

Traffic Lights. Other (abstract) pattern properties are visualized with a traffic light analogy, after sorting them globally into four bins ranging from 'very low' to 'very high' (Fig. 3, lower right half).

5 RESULTS

Our proposed visualization concepts are implemented in an interactive prototype, allowing us to evaluate usability and effectiveness aspects. In the following we briefly describe the implementation (Sec. 5.1), before we discuss some of the findings obtained with the system (Sec. 5.2).

5.1 Implementation

For the prototype, we use a web-based implementation relying on React[3] for the visualization frontend and a backend written in Kotlin[4]. Data processing as clustering and image processing, is conducted using Python scripts, relying on the SciPy[5] and OpenCV[6] libraries. Extracted feature descriptors, pattern attributes, and cluster similarities are cached in a MySQL database.

5.2 Findings in the Dataset

In the following, we present some of the datasets' intrinsic property structures and correlations, revealed by our visualization. Firstly, the kind of clusters obtained by clustering along a single object property. In this regard, we have selected the three varied properties: color, occurrence frequency, and regularity (Sec. 4.3), which are illustrated in single cluster views in Fig. 5a, 5b, and 5c, respectively. All of them comprise – depending on the property in question – different object representatives (Sec. 4.6).

I.e., in the first example – the color clusters (Fig. 5a) – the patterns are represented in their original state, segmented from the unrolled model surface. Four different color clusters are visible with the first one, comprising mostly greenish and yellowish patterns, while the fourth features all the darker patterns. Cluster two and three have mostly light-brown samples. The second cluster view (Fig. 5b) shows the patterns clustered by their occurrence frequency. Here, the stacked pattern contours are employed to visualize the variety and frequency of pattern shapes belonging to a common pattern archetype. From this image, it can be seen that some pattern classes are strongly correlated with the occurrence frequency. More specifically, it appears that cross-like shapes generally have very few occurrences, while staircase-shaped patterns have very many entities. Other shapes like rectangles, tears, circles, etc. are somewhat in-between. For the third cluster view (Fig. 5c), showing the regularity property, we use the position & distribution layout. This helps us to easily spot that the first cluster comprises mostly completely random pattern distributions, while the fourth cluster features clearly features several checkerboard arrangements and other very regular layouts. The examples from cluster two and three exhibit at least either row or column structures.

In another case example, we look into inter-property correlations which are revealed by the Bipartite Clusters Graph. Six of the correlations found in the Peruvian pottery dataset have been selected and are given in Fig. 6 with the side-by-side cluster view. It can be seen (Fig. 6a) that the patterns on the bowl are strongly correlated with the fourth color cluster. I.e., the patterns on bowls generally exhibit darker hues. The bowl shape seemingly also entails a high scale variability (Fig. 6b). Two significant correlations are also established for the pot objects. Firstly, the patterns on this shape are mostly of the light-brownish hue found in color cluster three (Fig. 6c). Secondly, the pot shape also strongly correlates with shape cluster four, which is comprised to a large extent of patterns with a staircase or pyramid-

[3] https://reactjs.org [Accessed 2023-04-26]
[4] https://kotlinlang.org [Accessed 2023-04-26]
[5] https://scipy.org [Accessed 2023-04-26]
[6] https://opencv.org [Accessed 2023-04-26]

(a) Vessel Shape ∼ Color

(b) Vessel Shape ∼ Scale Variability

(c) Vessel Shape ∼ Color

(d) Vessel Shape ∼ Pattern Shape

(e) Color ∼ Occurrence Frequency

(f) Color ∼ Scale Variability

Figure 6: Side-by-side cluster views showing the strongly correlated clusters between the properties (a) vessel shape and color, (b) vessel shape and scale variability, (c) vessel shape and color, (d) vessel shape and pattern shape, (e) color and occurrence frequency, as well as (f) color and scale variability.

like outline (Fig. 6d). Those pattern types, in particular, seem to be characteristic for the pot shape.

Comparing the properties color and occurrence frequency also yields a strong interdependency between the color cluster three and the patterns with a very high occurrence frequency (Fig. 6e). The last correlation is between the color and the scale variability properties (Fig. 6f), as it appears that patterns with the least scale variability belong to the second color cluster.

6 DISCUSSION

The examples presented in Sec. 5 show that we are – even without being familiar with the explored domain – able to easily spot several correlations and clusters in the data, which are not revealed by simply looking at the 3D models or images. Besides exploring the data ourselves, we conduct a collaborative walkthrough and subsequent discussion and feedback round with two actual domain experts from the field of archaeology. The exploration workflow was very well received, and it was established that the presented visualization and exploration techniques would also be a valuable tool for other branches of archaeology. One thing that was particularly surprising was the amount of intrinsic object information that could be revealed with basic feature extraction and data processing techniques. Examples for that are the occurrence frequency or scale variability, which makes it easy to identify clusters within the data.

7 CONCLUSION

We introduce an approach for the visual comparison of different properties for sets of ancient pottery objects. Links and color-highlighting allow for identifying relationships between groups in terms of co-occurrence of objects. Our approach supports domain experts to understand vessel shape and ornament elements, and eventually to draw conclusions and help with the interpretation of digital vessel objects. We showed the

principal applicability of our concept on a small-sized annotated dataset. The proposed design, however, is also suitable for larger object collections. Future work includes the extension of the shape and ornament features to use, and the inclusion of metadata and textual descriptions of the objects. Data analysis methods as frequent pattern mining, could be a valuable addition to help domain experts search for interesting patterns in large amounts of vessel objects.

ACKNOWLEDGEMENTS

This work was partially co-funded by the Austrian Science Fund FWF and the State of Styria, Austria within the project *Crossmodal Search and Visual Exploration of 3D Cultural Heritage Objects* (CrossSAVE-CH, P31317-NBL).

REFERENCES

[BK10] BHAT, HARISH S and KUMAR, NITESH. "On the derivation of the bayesian information criterion". *School of Natural Sciences, University of California* 99 (2010) 4.

[BMP02] BELONGIE, SERGE, MALIK, JITENDRA, and PUZICHA, JAN. "Shape matching and object recognition using shape contexts". *IEEE transactions on pattern analysis and machine intelligence* 24.4 (2002), 509–522 5.

[Bre] BREWER, CYNTHIA A. *ColorBrewer*. URL: http://www.ColorBrewer.org (visited on 06/16/2022) 5.

[BTC10] BRIVIO, PAOLO, TARINI, MARCO, and CIGNONI, PAOLO. "Browsing Large Image Datasets through Voronoi Diagrams". *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), 1261–1270. DOI: 10.1109/TVCG.2010.136 2.

[DPT*12] DEUFEMIA, V., PAOLINO, L., TORTORA, G., et al. "Investigative Analysis across Documents and Drawings: Visual Analytics for Archaeologists". *Proceedings of the International Working Conference on Advanced Visual Interfaces*. AVI '12. Capri Island, Italy: Association for Computing Machinery, 2012, 539–546. DOI: 10.1145/2254556.2254658 2.

[ES16] ESCHBACH, NORBERT and SCHMIDT, STEFAN. *Töpfer Maler Werkstatt: Zuschreibungen in der griechischen Vasenmalerei und die Organisation antiker Keramikproduktion*. Beihefte zum Corpus vasorum antiquorum ; Bd. 7. CH Beck, 2016. ISBN: 3406669409 1.

[GPD17] GLINKA, KATRIN, PIETSCH, CHRISTOPHER, and DÖRK, MARIAN. "Past Visions and Reconciling Views: Visualizing Time, Texture and Themes in Cultural Collections". *Digit. Humanit. Q.* 11 (2017) 2.

[HSS15] HADLAK, STEFFEN, SCHUMANN, HEIDRUN, and SCHULZ, HANS-JÖRG. "A Survey of Multi-faceted Graph Visualization". *Eurographics Conference on Visualization (EuroVis) - STARs*. The Eurographics Association, 2015. DOI: 10.2312/eurovisstar.20151109 2.

[Jac01] JACCARD, PAUL. "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines". *Bull Soc Vaudoise Sci Nat* 37 (1901), 241–272 5.

[KPP96] KARRAS, GE, PATIAS, P, and PETSA, E. "Digital monoplotting and photo-unwrapping of developable surfaces in architectural photogrammetry". *International Archives of photogrammetry and Remote Sensing* 31 (1996), 290–294 6.

[LKK*20] LENGAUER, STEFAN, KOMAR, ALEXANDER, KARL, STEPHAN, et al. "Visual Exploration of Cultural Heritage Collections with Linked Spatiotemporal, Shape and Metadata Views". *Vision, Modeling, and Visualization*. The Eurographics Association, 2020. DOI: 10.2312/vmv.20201196 2.

[Llo82] LLOYD, STUART. "Least squares quantization in PCM". *IEEE transactions on information theory* 28.2 (1982), 129–137 5.

[LSP*21] LENGAUER, STEFAN, SIPIRAN, IVAN, PREINER, REINHOLD, et al. "A Benchmark Dataset for Repetitive Pattern Recognition on Textured 3D Surfaces". *Computer Graphics Forum* (2021). ISSN: 1467-8659. DOI: 10.1111/cgf.14352 3.

[MPCC13] MAURI, MICHELE, PINI, AZZURRA, CIMINIERI, DANIELE, and CIUCCARELLI, PAOLO. "Weaving Data, Slicing Views: A Design Approach to Creating Visual Access for Digital Archival Collections". *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI*. Trento, Italy: Association for Computing Machinery, 2013. DOI: 10.1145/2499149.2499159 2.

[PS98] PAPADIMITRIOU, CHRISTOS H and STEIGLITZ, KENNETH. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998 5.

[SIBdS16] SIMON, RAINER, ISAKSEN, LEIF, BARKER, ELTON T. E., and de SOTO CAÑAMARES, PAU. "Peripleo: a tool for exploring heterogenous data through the dimensions of space and time". *Code4Lib Journal* (2016) 2.

[SLL*21] SIPIRAN, IVAN, LAZO, PATRICK, LOPEZ, CRISTIAN, et al. "SHREC 2021: Retrieval of cultural heritage objects". *Computers & Graphics* 100 (2021), 1–20. DOI: https://doi.org/10.1016/j.cag.2021.07.010 3.

[TKE12] TARAWANEH, RAGA'AD M., KELLER, PATRIC, and EBERT, ACHIM. "A General Introduction To Graph Visualization Techniques". *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering - Proceedings of IRTG 1131 Workshop 2011*. Vol. 27. OpenAccess Series in Informatics (OASIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, 151–164. DOI: 10.4230/OASIcs.VLUDS.2011.151 2.

[vLKS*11] Von LANDESBERGER, TATIANA, KUIJPER, ARJAN, SCHRECK, TOBIAS, et al. "Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges". *Comput. Graph. Forum* 30.6 (2011), 1719–1749. DOI: 10.1111/j.1467-8659.2011.01898.x 2.

[WFS*19] WINDHAGER, FLORIAN, FEDERICO, PAOLO, SCHREDER, GÜNTHER, et al. "Visualization of Cultural Heritage Collection Data: State of the Art and Future Challenges". *IEEE Transactions on Visualization and Computer Graphics* 25.6 (2019), 2311–2330. DOI: 10.1109/TVCG.2018.2830759 2.

[WSL*20] WINDHAGER, FLORIAN, SALISU, SAMINU, LEITE, ROGER A., et al. "Many Views Are Not Enough: Designing for Synoptic Insights in Cultural Collections". *IEEE Computer Graphics and Applications* 40.3 (2020), 58–71. DOI: 10.1109/MCG.2020.2985368 2.

ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.31, No-1-2, 2023

JengASL: A Gamified Approach to Sign Language Learning in VR

Alex Shaw, Burkhard C. Wünsche, Kevin Mariono, Aiden Ranveer, Manhui Xiao, Ryo Hajika, Yitong Liu
School of Computer Science, University of Auckland, New Zealand
lsha074@aucklanduni.ac.nz, burkhard@cs.auckland.ac.nz, gmar591@aucklanduni.ac.nz,
raid127@aucklanduni.ac.nz, mxia321@aucklanduni.ac.nz, ryo.hajika@auckland.ac.nz,
yliu959@aucklanduni.ac.nz

ABSTRACT

Learning sign language has many advantages ranging from being able to communicate with millions of hearing impaired people, to improving cognitive function and communication skills. Sign language is recognised as an official language in 74 countries, including Germany, Japan, and the UK. Despite that only a small percentage of people attempt to learn sign language.

In this research we investigate how virtual reality and gamification can be used to make learning sign language more enjoyable and motivating. We present JengASL, a gamified approach using 3D hand models, gesture recognition, and interactive gameplay in Virtual Reality to teach American Sign Language. We evaluate this system with a pilot study using eight participants and found that while it is less effective for sign memorisation than traditional 2D image-based learning methods, learning is more, but not significantly more, enjoyable and motivating.

Keywords

Virtual Reality, Sign Language, Teaching, Gesture Recognition, Gamification

1 INTRODUCTION

Learning sign language has numerous advantages ranging from being able to communicate with millions of people suffering from hearing difficulties [MV14] (an estimated 15-20% in America and New Zealand [BLC14, SFB+13]) to improving cognitive function and reasoning, memory, attention span, creativity, and communication skills [CCRV98, MLS06, CCP07].

Despite that sign language is not widely used and most people willing to learn it do so using books, videos, or through community-hosted events [MV14]. However, such teaching media might not always be available and/or provide little or no formative feedback. Furthermore, signs can be viewed differently depending on the angle at which they are observed, and 2D teaching materials cannot provide students with changes in depth and perspective. Virtual Reality overcomes some of these difficulties. By implementing 3D representations of hand gestures in a VR environment such as in [AVCA06], the user will be able to see different perspectives by tilting their head or walking to a different position.

The learning process is often inhibited by a lack of ongoing motivation. Learning a language takes time and effort. Gamification and Serious Games offer significant potential for improving motivation and persistence by combining the learning process with a more enjoyable gameplay activity, that also allows for more avenues of feedback and self-assessment through skill performance influencing performance in the gameplay tasks. Gamification has the potential to improve both user engagement and learning performance [ORCV17, SWL18].

Our research question is thus: Can we improve sign language learning using gamification in a Virtual Reality Environment?

2 RELATED WORK

2.1 Teaching Strategies

Teaching methods have been classified broadly into direct instruction, peer-teaching, and interactive teaching [MR17]. Interactive teaching is based on the idea that students need practical application to fully comprehend study material, motivating students to participate in teaching content and maintain concentration for longer. It also helps teachers to assess how well the student is actually learning the material. Feedback, in particular constructive feedback, is another factor in providing effective interactive learning [Sen18]. Good constructive feedback should be systematic, relevant and encouraging in order to achieve successful teaching [Ova91], and has been associated with increased student confidence and motivation [CR08]. Numerous papers and several systematic reviews have demonstrated the potential of VR, including gamification, in education [KLRWP17].

While numerous games exist for teaching sign language, few of them use VR or evaluate the effectiveness

of the approach. "CopyCat" is an educational adventure game to help deaf children improve their language and memory abilities [ZBP+11]. The player interacts with the game's main character using sign language and can use a virtual tutor to learn the correct signs. The research focus of the paper is on the sign recognition system developed as part of the research and no evaluation of the game is provided.

"Sign my World" is a mobile video game for teaching Australian Sign Language (Auslan). The game uses a 2D cartoon like interface and interactive cards which associate words with videos of the Auslan sign for that word [KPN12].

Bouzid et al. developed a memory game where users have to match cards containing words and sign writing notation, which are interpreted by a 3D avatar using gestures [BKEJ16]. The authors performed a user study with 9 participants and report that based on video analysis the majority of users were engaged and enjoyed the game.

2.2 Gesture Recognition for Sign Language Applications

A crucial aspect of any system teaching sign language is the recognition of the sign language gestures in order to assess users' performance. Accurate gesture recognition can also enable a system to provide constructive feedback that targets a specific part of the user's movements as suggested in SignTutor [AAA+09]. Previously developed technologies for gesture recognition can be split into two categories:

Glove-Based gesture recognition involves the user wearing a glove with markers or sensors. The CyberGlove is one such tool, which measures the angles of hand joints and the position of the hand, which can then be used to train a neural network for recognising gestures [WS99, PMS+09, SLC15].

Camera-Based gesture recognition methods use cameras or other optical sensors to gain data from the user by computing gestures using a visual representation of the hands. This includes first determining what needs to be examined, e.g., the hands, and then tracking their movements to determine the gesture being signed [LWLD11, FHA+22].

2.2.1 Object Segmentation

The first step of many tracking methods is foreground/background segmentation. Holden et al. use image sequences from a single colour camera to recognize Australian Sign Language (Auslan) using skin colour detection and active contour models [HLO05]. Object segmentation can be effected by variations in surrounding colours and lighting [HLO05]. SignTutor requires users to wear different coloured gloves to counteract changes in background and lighting conditions and helping with segmentation if fingers/hands overlap [AAA+09]. Keskin et al. used two camera images and non-skin-coloured gloves as markers to separate the hands from complex backgrounds [KEA03].

Depth cameras are often less sensitive to changes in lighting and background. Mo and Neumann used the Canesta camera to estimate the pose of the hands with the assumption that it is the closest object to the camera within the depth threshold, based on finger boundaries that are also calculated from depth values [MN06]. The method failed with non-frontal poses and poses that cannot be modelled due to noise or positioning. Li and Jarvis tried to remove some of the noise that comes with depth mapping using Median filtering and segmenting the hand using a depth histogram method [LJ09]. Histogram binning is also used in SignTutor to determine hand regions, although rather than using depth data it uses the HSV colourspace [AAA+09].

The Leap Motion sensor computes the 3D positions of hands within a certain range of the sensor, but instead of a depth map it dynamically computes a set of hand points (palm, finger positions, hand orientation). This was used by Chuan et al. to recognize 26 letters of the American Sign language alphabet by extracting the position and length the fingers as well as the pose of the palm [CRG14].

2.2.2 Gesture Classification

Gesture classification is usually achieved using Machine Learning. Keskin et al. use a Hidden Markov Models (HMMs) and a 3D Kalman filter to reduce noise [KEA03]. SignTutor uses two Kalman filters, one on each hand to reduce segmentation noise and predict hand trajectories [AAA+09]. Chuan et al. used a k-Nearest Neighbour and Support Vector Machine for alphabet recognition, however, gestures that looked similar were often misclassified, possibly due to mislabelled data in the Leap Motion sensor [CRG14].

Chai et al. used an interesting approach to build a translation application for Chinese Sign Language using Microsoft Kinect [CLL+13]. In their algorithm, the movement trajectory of each word is first aligned to the same sampling point. A match is then performed with existing libraries to determine the gesture. Since movements were tracked in this algorithm, it is possible to determine dynamic gestures and account for varying hand motion speeds between different signers.

More recent work has used deep learning. For example, Kothadiya et al. use two deep learning models for Indian Sign Language to achieve 97% recognition accuracy over 11 signs [KBS+22]. Al-Qurishi et al. present a review of deep learning-based approaches for sign language recognition [AQKS21]. The authors conclude that the presented models are relatively effective for a

range of tasks, but none currently possess the necessary generalization potential for commercial deployment.

The majority of papers we found focused on sign recognition [ABA21]. SignTutor does perform teaching and assesses users accuracy, but is limited by using a 2D display and not having gamification, which means that users must be intrinsically motivated to use the application [AAA+09]. Hence our design will focus on the gamification of sign language learning.

3 DESIGN

Based on the above literature, constructive feedback and interaction plays a large part in whether or not students can effectively learn. JengASL's Design can be split into several components as illustrated in Figure 1, providing both teaching and feedback via Gesture Recognition.

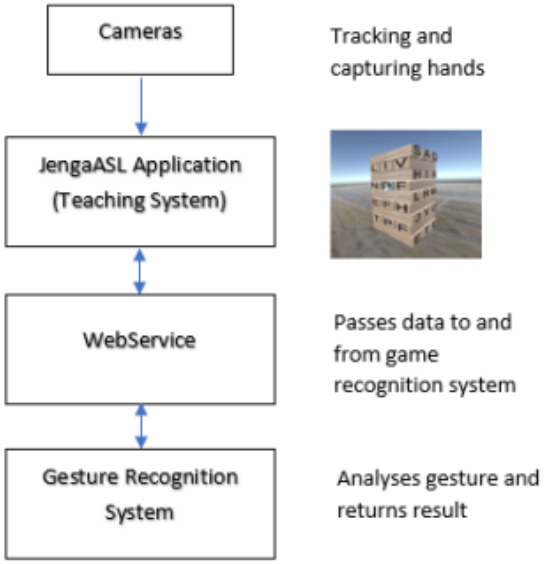


Figure 1: JengASL system architecture overview.

Cameras are accessed from within the JengaASL application to record the users hand gestures. Photos are then parsed in-game into a format suitable for gesture recognition, and data is passed to the gesture recognition system via a web-service.

3.1 JengASL

In order to make learning more interesting and interactive, we have integrated gamification into our VR teaching system. Our game consists of wooden blocks that are stacked as in the well known game "Jenga", each with letters attached to them. Jenga was chosen as a popular game that is both simple and can be played alone or with multiple players. A point system is used to keep track of how well the user is performing, and the game ends once the tower falls.

The user interacts with the system by first selecting a block with an associated letter, as seen in Figure 2. The user then sees an indication of the associated sign and is prompted to replicate it for the camera.

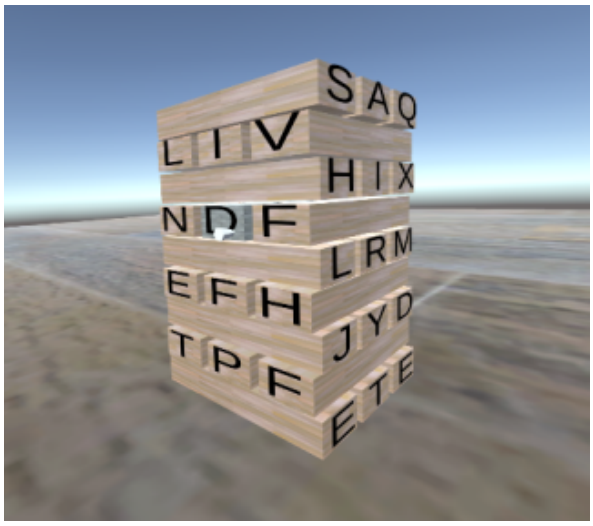


Figure 2: Selecting a block in the game environment.

One of our key contributions to ASL learning is the ability of users to view the gesture at different angles, helping them to learn the gesture as well as assisting them in being able to recognise the gesture in real life applications. When users choose a block they want to move, models of the hand gesture corresponding to the letter on the block are displayed in front of the user. The user is then able to walk around to look at the hand gesture from different view points. To make it easier for the user to see from both the perspective of the signer and signee without having to walk all the way around, both the front and back views of the gesture are shown (Figure 3).

Figure 3: In-game sign demonstration.

In order to effectively teach users, we use active teaching to allow them to practice what they learn. In order to move a block in JengASL, the user must perform the sign language gesture corresponding to the letter on the block. The gesture is captured using a web cam and immediate feedback is given by use of the gesture recognition system, which returns letter/likelihood pairs. For example, the following returned data - '(A, 0.8), (B,0.3), (C,0.5)...' - would indicate that the gesture had an 80% probability of being the letter 'A' and 0.3% probability of being the letter 'B', and so on. The letter with the highest likelihood is shown to the user and they are given an option to try again if it is not the one they intended. In order to motivate users to achieve greater accuracy in their gesturing, a point system is used. Each time a block is removed from the tower, points P are added, with an amount determined by the following equation:

$$\begin{aligned} AttemptsLeft &= MaxAttempts - Attempts \\ P &= 100 * |difficulty - accuracy| * AttemptsLeft \end{aligned} \tag{1}$$

Accuracy is obtained from the gesture recognition system as the percentage likelihood of the gesture being the specified letter. Due to it being highly unlikely that any user will ever achieve 100% recognition accuracy for a gesture, we implement a percentage accuracy difficulty level as a cap, which can be modified. In this case, the user only has to achieve, e.g., a 80% gesture accuracy to achieve full points, and higher difficulties would have a higher cap. Doing so can also encourage players to set goals and improve their gesture accuracy, and discourage them from settling for an incorrect gesture to remove the block. Similarly, to motivate the user to become more accurate, the more times the user decides to perform the selected gesture, the fewer points they will be awarded.

An important aspect of game design is to reward the user for doing well, and provide consequences if they do not. In our case the user is rewarded by gaining more points, and penalties are applied by reducing the smoothness with which blocks are removed from the tower. "Jitter" is added to blocks' movement as random vertical shaking. The jitter motion is inversely correlated with gesture accuracy. The height of a block within the tower is taken into consideration as shaking in at the bottom of the tower has a higher chance of toppling it over. The jitter factor is a modifiable constant that depends on the difficulty of the game. If too much jitter occurs (causing instability), the tower will topple over and the game will end. We use a physics system for the jitter, such that if a block is removed with jitter but the tower does not fall, the jitter will have still nudged other blocks in the tower, potentially reducing its stability. This both replicates the real-world version of Jenga and incentivises accuracy on *every* sign. Even if a given sign is not inaccurate enough to knock the tower over, consistently making mistakes will dramatically increase the risk to loose the game. Using probabilities rather than yes/no decisions also adds excitement to the game since, as with real Jenga, the player can never be sure what will happen.

3.1.1 Gesture Recognition

In order to provide correct feedback, we must be able to recognise the accuracy of the users' gestures. Ideally, we would like to provide precise feedback about which finger positions are incorrect. However, this proved difficult with existing technologies such as leap motion, which struggled with capturing motions where fingers overlap. We instead opted for using webcam input and machine learning (CNNs). While we used a very simple model and small training data set, recent publications show that the technology is advancing rapidly and capable of providing increasingly accurate recognition [AQKS21].

In order to reduce size and memory overheads associated with integrating a large trained CNN model into a game engine, we decided to implement a web-service which will be queried in game when the user performs a gesture. The web-service runs the model and returns the result to the game client, which will then parse it into a format suitable for use within the game.

4 IMPLEMENTATION

4.1 JengASL

Our game is implemented using Unity to host the JengASL application. Unity provides in-built Virtual Reality support, and also comes with many assets and game objects which can be used with our game, allowing us to reduce time spent on building the VR environment. The game was built based on the publicly available JengaVR [Ngi17], which implemented the physics required for the blocks to interact with each other or fall, however it had to be extended to include menus, gaze-interaction, webcam access, and the point system. Since we use an older head-mounted display without eye-tracking, gaze interaction uses the user's viewing direction obtained by the HMDs orientation.

A webcam was used in our work to capture users' gesture information by clicking a button in game. Block selection was done using gaze interaction to increase immersion, allowing users to have both hands free as they do their gestures. Communication between the game and the gesture recognition system is done through a TCP connection. During our study the web service for gesture recognition was hosted locally to mimise latency.

Learning Method	Mean	Std. Dev.
Traditional	89.06	20.20
JengASL	66.30	25.68

Table 1: Correctness Rate of the traditional and JengASL learning method (in %)

4.2 Gesture Recognition

For gesture recognition, we used a pre-trained VGG 16 model from Python's Keras library, with 16 total layers including input and output, and configured to classify ASL alphabets.

The dataset chosen for training provided 3000 images for each letter of the ASL alphabet. In this work we chose to use the 8 letters that the model was able to recognise with the highest accuracy.

5 EVALUATION

We conducted a small pilot study with 8 users aged 18–33 to test the effectiveness of our VR learning system, collecting both qualitative and quantitative data. 7 out of 8 users had no experience with ASL, and 6 had no experience with VR. Each user tested both the VR system, as well as the traditional method of looking at ASL gesture images. Demographic information of the users were collected at the beginning of each trial, before they were invited to complete two learning sessions of 8 gestures:

1. Session 1: 2D images of 8 different gestures representing characters different from session 2 (duration: 3 minutes)

2. Session 2: JengASL with 8 different gestures representing characters different from session 1 (duration: 5 minutes)

The additional time given to the VR game was to accommodate for in-game loading times and minor delays associated with interacting via the hardware.

The rate of gesture retention of participants was measured by asking users to perform each of the gestures that they learnt in the preceding session, and recording the correctness.

Data for qualitative analysis was collected using the Intrinsic Motivation Inventory (IMI) questionnaire [RD06, CSD22], with 3 sub-scales - Interest/Enjoyment, Effort/Importance, and Pressure/Tension. Users were asked 17 questions from these sub-scales with a rating from 1 to 7, and for all subscales the average score for its questions was recorded.

6 RESULTS

Table 6 gives the retention rates for both conditions. All users attempted 8 gestures for the traditional method and an average of 7 for the VR method, with the lowest being 4. The performance of users in VR had a moderately strong correlation with their performance in the Traditional method tests (Pearson's r = 0.47). From our data, JengASL performs worse than the traditional method, with a lower mean correctness rate. The differences in correctness is shown to be statistically significant using a two-tailed paired t-test ($p < 0.05$).

Table 2 shows participant responses to the IMI questionnaire subscales. Pressure/tension showed a strong positive correlation to correctness for the VR environment (r = 0.60), and only a weak correlation for traditional (r = 0.28). Effort/importance showed a weak negative correlation for traditional (r=-0.18) but a moderate positive correlation for VR (r = 0.40). Interestingly however, in traditional and VR methods there is a moderate positive correlation between effort/importance and correctness (r = 0.41 and r = 0.50 respectively). In both cases, interest/enjoyment had a moderate negative correlation with gesture correctness.

We can see that on average, users enjoyed the VR method more than the traditional method, made a similar effort for learning, and felt more pressure since a wrong sign could mean loosing the game. However, a two-tailed paired t-test showed no statistically significant difference between the two methods for all subscales.

7 DISCUSSION

JengASL was able to increase user interest and effort, although not at a statistically significant level. The increase in enjoyment was unsurprising, however at a lesser degree as expected. This may have been because although a gamified VR learning environment is a fresh and interesting idea, enjoyment is dependent on the game. Our gesture recognition system required a controlled environment (implemented in this case by placing a black screen behind the user's hand during recognition), and may have made it more awkward or difficult for users to play the game. The amount of jitter implemented in the game also needed to be optimised through user feedback.

Our system failed to increase the retention rate of gestures compared to the baseline. The most likely reason for this is the time restraint implemented during our evaluation: while looking at an image (traditional method), the user is likely to spend the whole time focusing on the image and attempting to memorise the gestures. However, in the VR system gestures are displayed when a block is clicked, and users have only until it disappears to look at and copy the gesture, leading to less time in total to memorise. This means that it

	Traditional Learning		Learning using JengASL	
Sub-Scale	Mean	Trad. Std. Dev.	Mean	Std. Dev.
Interest/Enjoyment	4.09	1.61	4.70	1.17
Effort/Importance	3.3	1.2	3.325	0.93
Pressure/Tension	4.45	1.14	5.325	1.08

Table 2: IMI Results with scores on a 5-level Likert scale from 1 (strongly disagree) to 5 (strongly agree) for traditional learning and learning using our VR tool JengASL.

could take longer to learn gestures using JengASL compared to traditional methods. Note however that taking more time to memorise does not mean our system is less effective, as increased enjoyment means users are likely to be more motivated to put in more time to play the game, or even more motivated to begin learning in the first place.

7.1 Limitations

Our study suffers from order effects since condition 1 (2D images) was always performed first. The reason for this was that we believe that combining three new experiences at once (sign language, game, and VR) might be too demanding. Since we used different sets of characters for each condition, we believe that learning characters for condition 1 should have limited effects on learning characters for condition 2. Some effects might still exits such as getting exhausted or bored after completing condition 2. This might be an additional explanation for the lower retention rates with the VR conditions.

The fact that we used different character sets for each condition might create another problem. The CNN for sign recognition had different accuracies for different characters and we hence chose the 8 characters with the highest accuracy. This meant that for the 2D image condition we had to choose randomly 8 from the remaining characters. The characters used in one condition might be more difficult to learn or more difficult to form using hand gestures, which would effect retention rates and recognition accuracy. For example, the letter “C” is relatively easy to learn and form since it involves making a “C” shape with the hand.

Other limitations are the small size of the user study (n=8) and self-selection bias since participants volunteered and we might have only got students with an interest in research, sign language, or games.

7.2 Design Considerations

Despite the negative results in terms of performance, this work has revealed several key considerations and barriers to be considered when designing tools such as this one for teaching gesture based skills.

One key consideration is the method by which the user interacts with the non-skill elements of the virtual environment. In particular, if the skill requires both hands as in the case of New Zealand Sign Language. In JengASL, users were required to use a controller to identify the Jenga block to be removed from the tower. This was motivated by previous research [ADWW19] and our own observations that gesture-based input with the available technologies was not accurate enough. While it is certainly possible for the user to use the controller to interact with the environment then put it down to perform gestures with both hands, this is an interruption to their activity and a barrier to usability. For this reason, even had training data availability not been a concern, JengASL would still be better suited for teaching American Sign Language (a one-handed language), than New Zealand Sign Language.

Another barrier to use is the availability of training data. While there is training data widely available for the subset of sign language signs that is the alphabet (for many languages), acquiring a dataset that is representative of the larger vocabulary of the language is difficult, not to mention the practical concerns around training a model with high accuracy for a large number of signs.

We also raise the key consideration of feedback and assessment with respect to precision as something to consider when developing a training tool. In JengASL, we provide feedback on accuracy in the form of score and the jitter mechanic. This is important to avoid sloppy use of the taught skill - particularly in the case where the skill in question is for communication and may have many very similar signs.

A final barrier to use for some sign languages is the presence of non-hand gestures in signs. In the example of NZSL, some signs can also include motion (e.g. the sign for “H”), touching the head (e.g. the sign for “Deaf”), and mouthing the associated word. While this is not the case for all sign languages, it is a limiting factor to what signs can be taught with a purely gesture-based system.

We did not have any problems with cybersickness and refer readers to design considerations listed by Shaw et al. [SWL+15] and Yin et al. [YBH+21] (section 3.6) in order to reduce cybersickness and make VR experiences safe.

8 CONCLUSIONS

In this paper, we have presented a tool that integrates sign language teaching into an immersive VR game.

One of the main contributions of our work is to allow learners of sign language to see various gestures from different angles, for easier learning of the different signs. While our system was not able to increase the retention rate of users' sign knowledge, our model does show potential in being more enjoyable, and thus more motivating than learning gestures using traditional methods.

At present there are several significant barriers to the use of VR tools and gesture recognition models for teaching sign language. Some of these can be mitigated with appropriate training data for the gesture recognition system, but the nature of signs in some sign languages and the large possible vocabulary makes such tools currently only suitable for supplementary learning and practice. Further research and development is necessary before these tools are suitable for standalone teaching.

Costs are also a factor for widespread adaption of VR training tools. Our solution uses a simple web cam (20 US$) and a head-mounted display connected to a desktop computer (we used an old Oculus Rift 2, second-hand about 200 US$).

8.1 Future Work

In future work we would like to develop/use more accurate neural networks for sign recognition and test them for different sign languages (ASL, NZSL, Auslan).

We would like to improve the teaching quality of the tool by enabling users to see hand gestures for longer and by providing more informative feedback. For example, a virtual model could be overlayed on the user's hand and the user then has to modify his/her hand gesture to precisely match the model.

Learning is most effective if the material is challenging, but not too difficult. Ideally we would like to measure cognitive load during the learning process [ABC+22] and then either adjust difficulty or provide feedback or visual hints in order to match task difficulty with the learners capabilities. Concepts used in intelligent tutoring systems would also be useful to increase learning [CLW18].

Finally, we would like to make a more extensive user study using randomly assigned characters for each condition with more participants, a longer training phase, and testing both short-term and long-term retention.

In using Jenga, we have taken a game that is traditionally played in a multiplayer form and used it as a solo teaching tool. It would be interesting to investigate how competition and competitiveness factor into motivation, enjoyment, and skill retention in a multiplayer gamified learning environment.

9 REFERENCES

[AAA+09] Oya Aran, Ismail Ari, Lale Akarun, Bülent Sankur, Alexandre Benoit, Alice Caplier, Pavel Campr, Ana Huerta Carrillo, and François-Xavier Fanard. Signtutor: An interactive system for sign language tutoring. IEEE MultiMedia, 16(1):81–93, 2009.

[ABA21] Ibrahim Adepoju Adeyanju, Oluwaseyi Olawale Bello, and Mutiu Adesina Adegboye. Machine learning methods for sign language recognition: A critical review and analysis. Intelligent Systems with Applications, 12:200056, 2021.

[ABC+22] Mohammad Ahmadi, Huidong Bai, Alex Chatburn, Burkhard C. Wünsche, and Mark Billinghurst. PlayMeBack - Cognitive Load Measurement using Different Physiological Cues in a VR Game. In Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology (VRST 22). New York, NY, USA, Article 70, pp. 1–2. ACM, 2022.

[ADWW19] Benjamin J. H. Andersen, Arran T. A. Davis, Gerald Weber, and Burkhard C. Wünsche. Immersion or Diversion: Does Virtual Reality Make Data Visualisation More Effective? International Conference on Electronics, Information, and Communication (ICEIC '19), Auckland, New Zealand, pp. 1–7, 2019.

[AQKS21] Muhammad Al-Qurishi, Thariq Khalid, and Riad Souissi. Deep learning for sign language recognition: Current techniques, benchmarks, and open issues. IEEE Access, 9:126917–126951, 2021.

[AVCA06] Nicoletta Adamo-Villani, Edward Carpenter, and Laura Arns. An immersive virtual environment for learning sign language mathematics. In ACM SIGGRAPH 2006 Educators Program, pp. 20–es. ACM, New York, NY, USA, 2006.

[BKEJ16] Yosra Bouzid, Mohamed Ali Khenissi, Fathi Essalmi, and Mohamed Jemni. Using educational games for sign language learning - a signwriting learning game: Case study. Educational Technology & Society, 19(1):129–141, 2016.

[BLC14] Debra L. Blackwell, Jacqueline W. Lucas, and Tainya C. Clarke. Summary health statistics for us adults: national health interview survey, 2012. Vital and health statistics. Series 10, Data from the National Health Survey, (260):1–160, 2014.

[CCP07] Allegra Cattani, John Clibbens, and Timothy J. Perfect. Visual memory for shapes in deaf signers and nonsigners and in hearing signers and nonsigners: Atypical lateralization and enhancement. Neuropsychology, 21(1):114–121, 2007.

[CCRV98] Olga Capirci, Allegra Cattani, Paolo Maria Rossini, and Virginia Volterra. Teaching Sign Language to Hearing Children as a Possible Factor in Cognitive Enhancement. The Journal of Deaf Studies and Deaf Education, 3(2):135–142, 1998.

[CLL+13] Xiujuan Chai, Guang Li, Yushun Lin, Zhihao Xu, Yili Tang, Xilin Chen, and Ming Zhou. Sign language recognition and translation with kinect. In IEEE Conf. on AFGR, volume 655, page 4, 2013.

[CLW18] Tyne Crow, Andrew Luxton-Reilly, and Burkhard C. Wünsche. Intelligent tutoring systems for programming education: a systematic review. In Proceedings of the 20th Australasian Computing Education Conference (ACE '18). New York, NY, USA. pp. 33–52. ACM, 2018.

[CR08] Mary P. Clynes and Sara E. C. Raftery. Feedback: an essential element of student learning in clinical practice. Nurse Education in practice, 8(6):405–411, 2008.

[CRG14] Ching-Hua Chuan, Eric Regina, and Caroline Guardino. American sign language recognition using leap motion sensor. In Proc. of the 13th International Conference on Machine Learning and Applications (ICMLA '14), pp. 541–544. IEEE, 2014.

[CSD22] CSDT. Intrinsic motivation inventory (IMI), 2022. http://selfdeterminationtheory.org/intrinsic-motivation-inventory/.

[FHA+22] Fahmid Al Farid, Noramiza Hashim, Junaidi Abdullah, Md Roman Bhuiyan, Wan Noor Shahida Mohd Isa, Jia Uddin, Mohammad Ahsanul Haque, and Mohd Nizam Husen. A Structured and Methodological Review on Vision-Based Hand Gesture Recognition System. Journal of Imaging, 8(6):153, 2022.

[HLO05] Eun-Jung Holden, Gareth Lee, and Robyn Owens. Australian sign language recognition. Machine Vision and Applications, 16(5):312, 2005.

[KBS+22] Deep Kothadiya, Chintan Bhatt, Krenil Sapariya, Kevin Patel, Ana-Belén Gil-González, and Juan M. Corchado. Deepsign: Sign language detection and recognition using deep learning. Electronics, 11(11), 2022.

[KEA03] Cem Keskin, Ayse Erkan, and Lale Akarun. Real time hand tracking and 3D gesture recognition for interactive interfaces using HMM. ICANN/ICONIPP, 2003:26–29, 2003.

[KLRWP17] Sam Kavanagh, Andrew Luxton-Reilly, Burkhard C. Wünsche, and Beryl Plimmer. A systematic review of virtual reality in education. Themes in Science and Technology Education, 10(2):85–119, 2017.

[KPN12] Jessica Korte, Leigh Ellen Potter, and Sue Nielsen. Designing a mobile video game to help young deaf children learn Auslan. In Proc. of the 26th Annual BCS Interaction Specialist Group Conference on People and Computers (BCS–HCI '12), pp. 345–350, 2012.

[LJ09] Zhi Li and Ray Jarvis. Real time hand gesture recognition using a range camera. In Australasian Conference on Robotics and Automation, pp. 21–27, 2009.

[LWLD11] Rui Liu, Burkhard C. Wünsche, Christof Lutteroth, and Patrice Delmas, A Framework for Webcam-based Hand Rehabilitation Exercises, Proc. of the International Conference on Computer Vision Theory and Applications (VISAPP '11), Vilamoura, Algarve, Portugal, pp. 1–6, 2011.

[MLS06] Diane C. Millar, Janice C. Light, and Ralf W. Schlosser. The impact of augmentative and alternative communication intervention on the speech production of individuals with developmental disabilities: A research review. Journal of Speech, Language, and Hearing Research, 49(2):248–264, 2006.

[MN06] Zhenyao Mo and Ulrich Neumann. Real-time hand pose recognition using low-resolution depth images. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06), volume 2, pp. 1499–1505. IEEE, 2006.

[MR17] Daniel Muijs and David Reynolds. Effective teaching: Evidence and practice. Sage, 2017.

[MV14] Rachel McKee and Mickey Vale. The vitality of New Zealand sign language project: Report on a survey of the Deaf/NZSL community. Technical report, Victoria University of Wellington, September 2014.

[Ngi17] Ngiani. Jenga–VR, 2017. https://github.com/ngiani/Jenga-VR.

[ORCV17] Margarita Ortiz-Rojas, Katherine Chiluiza, and Martin Valcke. Gamification and learning performance: A systematic review of the literature. In 11th European Conference on Game-Based Learning (ECGBL), pp. 515–522, 2017.

[Ova91] Martha N. Ovando. Constructive feedback: A key to successful teaching. Management, 5(3), 1991.

[PMS+09] Farid Parvini, Dennis McLeod, Cyrus Shahabi, Bahareh Navai, Baharak Zali, and Shahram Ghandeharizadeh. An approach to glove-based gesture recognition. In International Conference on Human-Computer Interaction, pages 236–245. Springer, 2009.

[RD06] Richard M. Ryan and Edward L. Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. American Psychologist, 55:68–78, 2006.

[Sen18] S. Senthamarai. Interactive teaching strategies. Journal of Applied and Advanced Research, 3(S1):36–38, 2018.

[SFB+13] Gretchen Stevens, Seth Flaxman, Emma Brunskill, Maya Mascarenhas, Colin D. Mathers, and Mariel Finucane. Global and regional hearing impairment prevalence: an analysis of 42 studies in 29 countries. The European Journal of Public Health, 23(1):146–152, 2013.

[SLC15] Neelesh Sarawate, Ming Chan Leu, and Özcelik Cemil. A real-time american sign language word recognition system based on neural networks and a probabilistic model. Turkish Journal of Electrical Engineering & Computer Sciences, 23(Sup. 1):2017–2123, 2015.

[SWL18] Ayoung Suh, Christian Wagner, and Lili Liu. Enhancing user engagement through gamification. Journal of Computer Information Systems, 58(3):204–213, 2018.

[SWL+15] Alex Shaw, Burkhard C. Wünsche, Christof Lutteroth, Stefan Marks, and Rodolphe Callies. Challenges in virtual reality exergame design. Proceedings of the 16th Australasian User Interface Conference (AUIC '15): Volume 162, pp. 61–68, 2015.

[WS99] John Weissmann and Ralf Salomon. Gesture recognition for virtual reality applications using data gloves and neural networks. In Proc. of the International Joint Conference on Neural Networks (IJCNN '99), volume 3, pp. 2043–2046. IEEE, 1999.

[YBH+21] Betty Yin, Samuel Bailey, Emma Hu, Milinda Jayarekera, Alex Shaw, and Burkhard C. Wünsche. Tour de Tune 2 - Auditory-Game-Motor Synchronisation with Music Tempo in an Immersive Virtual Reality Exergame. Proceedings of the 2021 Australasian Computer Science Week Multiconference (ACSW '21). New York, NY, USA, pp. 1–10. ACM, 2021.

[ZBP+11] Zahoor Zafrulla, Helene Brashear, Peter Presti, Harley Hamilton, and Thad Starner. Copycat: An american sign language game for deaf children. In Proc. of the IEEE International Conference on Automatic Face & Gesture Recognition (FG 2011), pp. 647–647, 2011.

ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.31, No-1-2, 2023

Reconstruction from Multi-view Sketches: an Inverse Rendering Approach

Joan Colom[0000-0001-7577-0657]
Universitat Politècnica de València
Camino de Vera
46022, Valencia, Spain
joacoco@inf.upv.es

Hideo Saito[0000-0002-2421-9862]
Keio University
Hiyoshi Kohoku-ku
223-8522, Yokohama, Japan
hs@keio.jp

ABSTRACT

Reconstruction from real images has evolved very differently from reconstruction from sketches. Even though both present similarities, the latter aims to surpass the subjectivity that drawings present, increasing the task's uncertainty and complexity. In this work, we draw inspiration from reconstruction over real multi-view images and adapt it to work over sketches. We leverage inverse rendering as a refinement process for 3D colored meshes while proposing modifications for the domain of drawings. Compared to previous methods for sketches, our proposal recovers not only shape but color, offering an optimization system that does not require prior training. Through the results, we evaluate how different quality factors in sketches affect the reconstruction and report how our proposal adapts to them compared to directly applying existing inverse rendering systems for real images.

Keywords

Computer Vision, Graphics, Optimization, Image Processing, 3D Reconstruction, Inverse Rendering.

1. INTRODUCTION

The reconstruction from realistic multi-view images has improved remarkably in recent years, capable of recovering 3D shapes and materials [Bos20a, Goe20a, Kim16a, Li22a, Mil20a, Mun21a, Wan21a, Zha21a, Zha21b]. However, reconstruction from sketches has received less attention, having a slower development.

Multi-view reconstruction, from any source, aims to obtain a three-dimensional description of an object depicted in multiple two-dimensional representations through different views. Consequently, both tasks have a similar nature. However, reconstruction from realistic images often is approached as *inverse rendering*. This area aims to revert the rendering process, recovering unknown scene parameters from observations [Kat17a, Lai20a, Li18a, Liu19a, Mun21a, Nie04a]. A rendering pipeline is assumed and consistent, approximating the laws of light transport. Therefore, the ambiguity of the task is reduced with the number of meaningful references.

In contrast, sketches are not the result of laws of physics; instead, they are *subjective* views of reality [Xia22a]. If different people draw the same object, the results will be very different. In the same way, sketches are not interpreted equally by everyone. This higher ill-posed nature distinguishes the task over sketches from the task over realistic images. While the latter aims to invert a well-known process and recover the lost scene information, the former aims to overcome subjective interference to build the most plausible object.

However, this is challenging as any or no object may fit all the sketched views. Some previous approaches present interactive alternatives involving the user [Li16a, Li18b], leaning on the user's decisions to deal with uncertainty. Other works have tackled automatic reconstruction, either from single [Gao22a, Gry20a, Wan18a, Wan20a, Zha21c] or multi-view sketches [Lun17a, Han20a]. However, they have focused on shape, not considering color.

Color is crucial to define the visual identity of an object. When recovering color and shape, the closest references are found in reconstruction from realistic images. This paper proposes using inverse rendering to optimize textured models from sketches and flat-colored drawings by modifying the ideas of Goel et al. [Goe20a] for real images[1]. Our contributions are:

- We adapt inverse rendering techniques for real images to the ill-posed task of reconstruction over sketches and flat-colored drawings, proposing a split loss to ease the joint optimization.
- We propose a sampling method to partially recover mesh colors after a remeshing step.
- We study how different quality factors and inconsistencies in the source sketches affect the reconstruction.

[1]Code at github.com/JoanCoCo/SketchReconstruction.

The rest of this paper is structured as follows. First, Section 2 presents recent approaches related to our work. Next, Sections 3 and 4 introduce the proposed solution and the results obtained. Finally, Section 5 details the conclusions drawn from the results.

2. RECENT SOLUTIONS

Many works tackle reconstruction, either from real images or sketches. This section focuses on works close to ours: reconstruction from sketches as images and multi-view inverse rendering reconstruction.

Reconstruction from sketches

There are two works closest to ours in this area. Firstly, Han et al. [Han20a] proposed extracting attenuation maps from multi-view sketches using a CGAN. These allowed optimizing a voxel grid by a Direct Shape Optimization algorithm, obtaining a voxel representation of the target using the view poses.

Secondly, Lun et al. [Lun17a] used a similar scheme with different ideas. Through a CNN, from sketches, they generated the depth, normal, and foreground probability maps from 12 fixed views. Using them, partial point clouds were obtained and fused. Later, the global cloud was converted into a mesh and refined through contour fitting. In this case, the network had to be trained for a fixed set of input viewpoints.

Both approaches used generative models for producing intermediate spatial descriptions to optimize a 3D representation. Therefore, they required training to obtain a usable system, needing paired sketches and 3D shape information.

Even though some hand-drawn datasets exist, such as [Gry19a] and [Xia22a], they do not provide enough samples for training. Therefore, previous works relied on synthetic data [Han20a, Lun17a, Wan20a, Zha21c], overlooking the subjectivity of hand-drawn samples and resulting in low generalization. Instead of a trainable system, we propose a refinement scheme using inverse rendering.

Inverse rendering

Inverse rendering methods revert the rendering process to estimate scene parameters from images, usually thanks to differentiable rendering. Therefore, images are used as feedback, not requiring a ground truth 3D shape. However, each reconstruction requires "training" to optimize the desired scene parameters.

Kim et al. [Kim16a] already proposed a refinement approach for reconstruction based on multi-view images. With an initial Structure-from-Motion estimation [Sch16a], a differentiable rendering was used to optimize mesh, albedo, and lighting. NeRF [Mil20a] introduced an MLP characterizing the space given the position and viewing direction, which was optimized per scene through volume rendering. This allowed novel view synthesis and scene inspection, inspiring many works [Bos20a, Wan21a, Zha21a, Zha21b]. Finally, NVDiffRec [Mun21a] gathered similar ideas, proposing a mesh optimization from scratch. Given multi-view images, masks, and viewpoints, they used differentiable rendering with deferred shading to jointly optimize a deformable tetrahedral grid, materials, and environment maps, outputting a textured 3D mesh.

These approaches offered systems directly usable with any set of images given the required inputs, optimizing a 3D representation of the provided data. However, the rendering pipeline was usually simplified, looking for a good balance between a realistic appearance and efficiency [Bos20a, Kim16a, Mun21a, Zha21b].

Goel et al. presented a refinement approach using differentiable path tracing to avoid the limitations of local lighting [Goe20a]. Starting from an initial mesh, they applied two alternating refinement steps repeated cyclicly: a material step focused on BRDF parameters and a geometry step updating the vertex's positions. Upon convergence, subdivision increased the mesh resolution, followed by remeshing to fix artifacts.

As reported in [Col22a], these techniques are generic enough to be applied over sketches. However, they were designed for realistic images, not considering the sketches' uncertainty and subjectivity. Therefore, they present components unsuitable for drawings, showing worse performance than in their intended domain.

The following section proposes modifying the inverse rendering optimization techniques to work over sketches and flat-colored drawings. We build on the foundations of Goel et al. [Goe20a].

3. PROPOSED SOLUTION

We modify the refinement scheme by Goel et al. [Goe20a] to make it suitable for sketches and flat-colored drawings. Our changes fall in four areas:

- Sketches do not require realistic materials or complex lighting [Col22a]. Therefore, we replace the BRDF materials with a single purely diffuse material and fix the environment map to entirely white. Consequently, the input is reduced to multi-view plain image drawings, masks isolating the target, and the view poses of each image.
- Simultaneous optimization of mesh and materials is possible, as shown by [Kim16a, Mun21a]. We replace the alternating scheme in [Goe20a] with simultaneous optimization, followed by a *long-tail* refinement of the colors.
- Goel et al. reset the material after remeshing, discarding the estimated color. We propose a resampling scheme to recover the color partially.
- We guide the refinement process using split losses for shape and color, as well as regularizations.

In contrast, common elements with Goel et al.'s work are the refinement over an initial mesh, using mesh

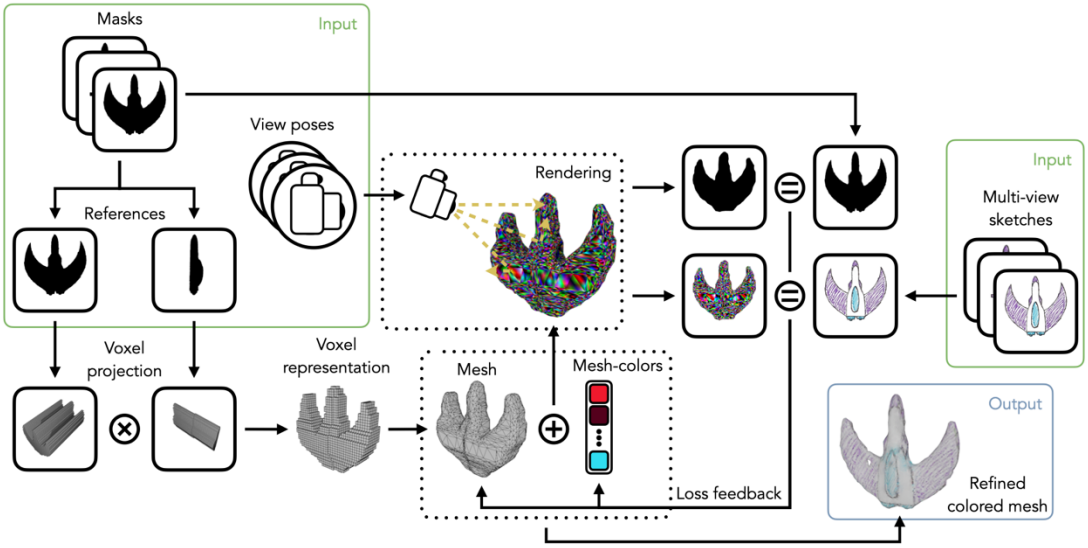


Figure 1. Summary of the basic proposal. Given multi-view sketches, masks, and view poses, an initial

colors [Yuk08a], and using remeshing for solving artifacts. The following sections present the solution in more detail, and Figure 1 summarizes it.

Optimization scheme

Given the sketches, masks, and viewpoints, an initial mesh is optimized to represent the sketched object. Goel et al. experimented with mesh initialization techniques, such as voxel carving or COLMAP [Sch16a], reporting the best performance with the latter. However, sketches and flat-colored drawings contain few key points, rendering COLMAP inadequate for our task. Instead, to obtain an initialization of the general shape, we use a simple visual hull estimation based on parallel projections from a given subset of sketches into a voxelated occupancy space. The total shape is the intersection of all the projections, and a mesh is obtained through marching cubes, remeshing, and simplification.

Mesh colors are used instead of textures to avoid the discontinuities of texture coordinates optimization [Mun21a], storing the sampled colors of the mesh triangles in a vector [Yuk08a]. Given the triangle ID t, resolution R, and barycentric coordinates (i, j) of a sample with $0 \leq i \leq R$ and $0 \leq j \leq R - i$, the index c of the sample in the color vector is computed with Equation 1. We initialize the colors randomly.

$$c = \frac{(R+1)(R+2)}{2}t + \frac{2R - i + 3}{2}i + j \qquad (1)$$

The refinement involves optimizing the mesh vertex positions and the color vector. These parameters compose a single object 3D scene that, when rendered from the given viewpoints, is converted into images. These can be compared with the references using loss functions. Finally, the differentiable rendering allows using gradient descent optimization to update the parameters jointly. Although disjoint coarse-to-fine optimization can allow finner detail [Goe20a], the detail requirement in sketches is generally low. Therefore, simultaneous optimization with fixed mesh resolution can lead to reasonable results in fewer steps.

Losses

Losses must account for the task's properties and the result's desirable features. We aim to obtain a 3D triangular mesh that resembles the sketched object. However, sketches are not consistent descriptions but interpretations of the world. Consequently, we do not aim to find a replica but a reasonable approximation.

The losses must inform the shape and colors of the target. Instead of capturing both with one loss [Goe20a], we set dedicated losses for better tailoring:

- Color loss. Color details in sketches are inconsistent. Sketch lines have two uses: conveying surface color detail or representing geometric features. Both produce color feedback, but only the former corresponds to true color information. Moreover, the second type is inconsistent between views. We use a Laplacian pyramid loss [Boj17a] for comparison at different resolution levels to deal with these issues. Coarser levels inform the general color. Meanwhile, finner levels reinforce consistent lines, while inconsistent lines are overtaken by coarse color feedback. Equation 2 presents it where I_P and I_T are the rendered and reference images; K is the number of levels; G is the Gaussian filter; $|I_P|$ and $D(I_P)$ are the number of pixels and channels of I_P; and I^l represents the image scaled down by l^{-1}.

$$L_C(I_P, I_T) = \sum_{i=1}^{|I_P|D(I_P)} \frac{\left|I_{P_i}^{K+1} - I_{T_i}^{K+1}\right|}{|I_P|D(I_P)} + \sum_{l=1}^{K} \sum_{i=1}^{|I_P|D(I_P)} \frac{\left|I_{P_i}^{l} - G\left(I_{P_i}^{l}\right) - I_{T_i}^{l} + G\left(I_{T_i}^{l}\right)\right|}{|I_P|D(I_P)} \quad (2)$$

- Silhouette loss. Due to the color inconsistencies and lack of shading, the silhouette is the primary source of shape information. We can capture it using the mean squared error of the masks. Even though the outline can present inconsistencies, this loss balances the feedback, averaging the references. Equation 3 models the loss, being M_P and M_T the rendered and ground truth masks.

$$L_M(M_P, M_T) = \frac{1}{|M_P|} \sum_{i=1}^{|M_P|} \left(M_{P_i} - M_{T_i}\right)^2 \quad (3)$$

However, to guide the reconstructions toward desirable properties, regularizations are also needed:

- Shape regularization. It favors smooth meshes and avoids degenerations. Equation 4 formulates it as an adaptation of the curvature flow smoothing [Oht01a] for uniformity on uneven distributions, where V is the set of vertices; F obtains the set of pairs of vertices that form a face with the input, and α_v is the angle at vertex v in a given triangle.

$$L_{CF}(V) = \sum_{v_i \in V} \left\| \sum_{v_j, v_k \in F(v_i)} (v_j - v_i)\, cot\, \alpha_{v_k} + (v_k - v_i)\, cot\, \alpha_{v_j} \right\|^2 \quad (4)$$

- Normal regularization. It favors meshes that induce automatic smooth normals by penalizing their variation inside a neighborhood. Equation 5 defines it based on Laplacian regularization [Mun21a, Oht01a], where N and n_v are the one-ring neighborhood and normal of a vertex.

$$L_N(V) = \sum_{v_i \in V} \left\| \frac{1}{|N(v_i)|} \sum_{v_j \in N(v_i)} n_{v_j} - n_{v_i} \right\|^2 \quad (5)$$

- Color smoothness regularization. It favors color uniformity inside the triangles. Equation 6 defines it, where T is the triangles set, C is a color vector, and $C_{R(s)}^{t}$ is the color of a random sample in t.

$$L_{CS}(T, C) = \sum_{t_i \in T} \left\| \frac{1}{4} \sum_{s=2}^{5} \left| C_{R(1)}^{t_i} - C_{R(s)}^{t_i} \right| \right\|^2 \quad (6)$$

- Spring regularization. Inspired by [Hop93a], it aims for a minimum solution, disfavoring overgrowing and balancing edge sizes. Equation 7 defines this function.

$$L_S(T) = \sum_{\substack{v_1, v_2, v_3 \in t_i \\ t_i \in T}} \|v_1 - v_2\| + \|v_2 - v_3\| + \|v_3 - v_1\| \quad (7)$$

Finally, decay of the normal regularization is applied to further avoid overgrowing, while progressive strengthening of the shape regularization is used to penalize significant changes towards the end. The total loss is expressed by Equations 8, 9, and 10, where MI is the maximum number of iterations and i is the current one. The weights of the losses were set empirically to make the main losses dominant while keeping the regularization magnitudes balanced.

$$L = 40L_M + 10L_C + 0.02d_{CF}(i)L_{CF} + 0.01d_N(i)L_N + 0.0002L_{CS} + 0.0025L_S \quad (8)$$

$$d_{CF}(i) = 1.0 - \max\left(0.01, \left(1.0 - 1.5\frac{i}{MI}\right)\right) \quad (9)$$

$$d_N(i) = \max\left(0.05, \min\left(0.4, \left|\log\left(\frac{i}{MI} + 10^{-4}\right)^5\right|\right)\right) \quad (10)$$

Remeshing and resampling

The measures for avoiding overgrowing still leave room for slight degeneration. Periodic screened Poisson reconstruction is applied to fix them, followed by simplification to keep the number of faces constant. However, this interferes with color estimation, as mesh colors are linked to the triangle IDs. When remeshing, a new mesh is generated, redefining vertices and faces. As a result, triangles in the same spatial position have different IDs, shuffling the surface colors. Therefore, part of the progress is lost. Instead of reinitializing the color as in [Goe20a], we propose a sampling method to recover lost progress.

By storing a copy of the mesh before remeshing, the colors of the new mesh can be updated sampling it. From now on, the input and output meshes to the remeshing are named m^i and m^o, respectively. Similarly, the color vectors of the input and resulting meshes are $\vec{c}^i$ and $\vec{c}^o$. With this, the resampling procedure consists in:

- For each triangle t_k^o in m^o, compute the world coordinates s_n^o of every sample inside it [Yuk08a].
- For each triangle t_k^i in m^i, compute its center $\bar{t}_k^i$ by averaging the positions of its vertices.
- The distance matrix from each sample to each center is computed. This allows finding the closest triangle of m^i to each sample in m^o, obtaining $\mathbb{C} = \left\{ \left(s_n^o, t_k^i\right) | s_n^o \in m^o, t_k^i = \underset{t_h^i \in m^i}{\operatorname{argmin}} \left\| s_n^o - \bar{t}_h^i \right\| \right\}$.
- For every pair in $\mathbb{C}$, the barycentric coordinates of the projection of s_n^o into t_k^i are obtained [Hei05a]. With them, the index j in $\vec{c}^i$ associated to the projection is computed [Yuk08a]. By obtaining the index h of s_n^o in $\vec{c}^o$, $\vec{c}_j^i$ can be copied into $\vec{c}_h^o$.

In summary, this approach finds for each sample in m^o the closest color in m^i. Assuming topological similarity between a mesh and its remeshed version; it is expected that the colors for the new mesh will be the

same as the closest points in the original mesh. This is reasonable, as Poisson remeshing tends to generate a smoother version of the original mesh, preserving the topology unless a very degenerated mesh is used.

The closest triangle is found using the distance to the center. Even though this can fail in some cases, it is generally a good approximation, allowing an efficient implementation with matrix operations. Additionally, it can be compensated by finding the N nearest triangles in m^i for each s_n^o, averaging the N projection colors. This recovers a less detailed version of the original colors. Nonetheless, the general colors are restored, and previous progress can be used.

Implementation details

Batch optimization is used with a batch size of four samples, and the initial mesh is estimated from a frontal and a side sample unless otherwise stated. Additionally, to compensate for the loss in color detail after remeshing, a long-tail training is used in which the last iterations focus on color only, fixing the shape. Remeshing is applied every N iterations and before the long tail when used, but not after.

Following [Col22a, Goe20a], we use a differentiable path tracing render as it reveals artifacts hidden by local lighting. We employ *pyredner* [Li18a], having added the mesh colors implementation of [Goe20a] to a recent version. The mesh colors resolution is set to three, and optimization renders use one bounce and four ray samples unless otherwise stated.

The estimated color vector is converted into a texture to export the result. This is done using optimization. From the reconstructed mesh, N sample images are generated from random views. Using them as references, a random texture mapped to the mesh is optimized using the color loss of Equation 2 and the texture smoothness regularization of [Mun21a]. Otherwise stated, ten reference views, 100 steps, and a texture resolution of 2048 by 2048 pixels are used. Finally, the optimizations are done using Adam, with a learning rate of 0.005 for the reconstruction and 0.05 for the texture generation.

4. EXPERIMENTAL RESULTS

Two synthetic examples were used for testing our system. Even though in Section 2 we stated why synthetic datasets do not fully represent our task, they are a baseline. Synthetic examples lack subjectivity, making their results the best case possible. Additionally, using synthetic samples allowed us to alter their quality, studying different input factors.

Through this section, we first present the data used and the baseline results. Next, we summarize how various quality factors of the sketches influence the results and show the effects of our sampling. Finally, we compare our results with those obtainable with Goel et al.'s [Goe20a] and Munkberg et al.'s [Mun21a] systems.

Figure 2. Models used and synthetic sketches in three styles.

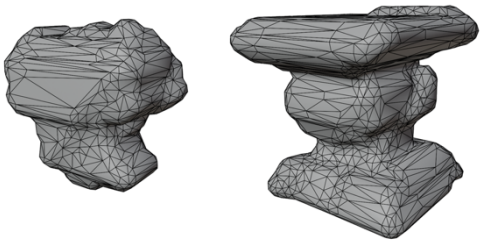


Figure 3. Initial mesh for Axolotl and Vasque.

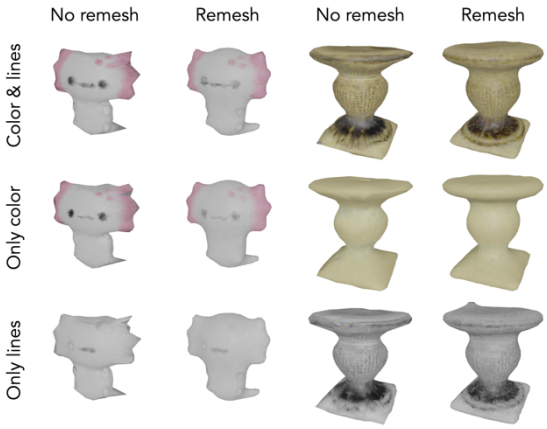


Figure 4. Baseline results for different styles.

Datasets

We gathered two 3D models to generate sketch-like references: Axolotl [fel20a] and Vasque [Fre07a], processing them to fit a cube of two units. The reasons behind this choice resided in their characteristics, representative of commonly drawn objects. While Axolotl has multiple colors, roundness, asymmetry, and a fictional appearance, Vasque represents a manufactured object with symmetry, sharp edges, and curves. Using Blender's Freestyle module as a rendering pipeline, we generated synthetic samples in three styles: lined sketches without color, flat-colored lined sketches, and flat-colored sketches without lines, shown in Figure 2. Each one of the sets contained 128 training and 128 validation random view samples of 512 by 512 pixels. The views were placed at a distance of five units around the target looking at it. From now

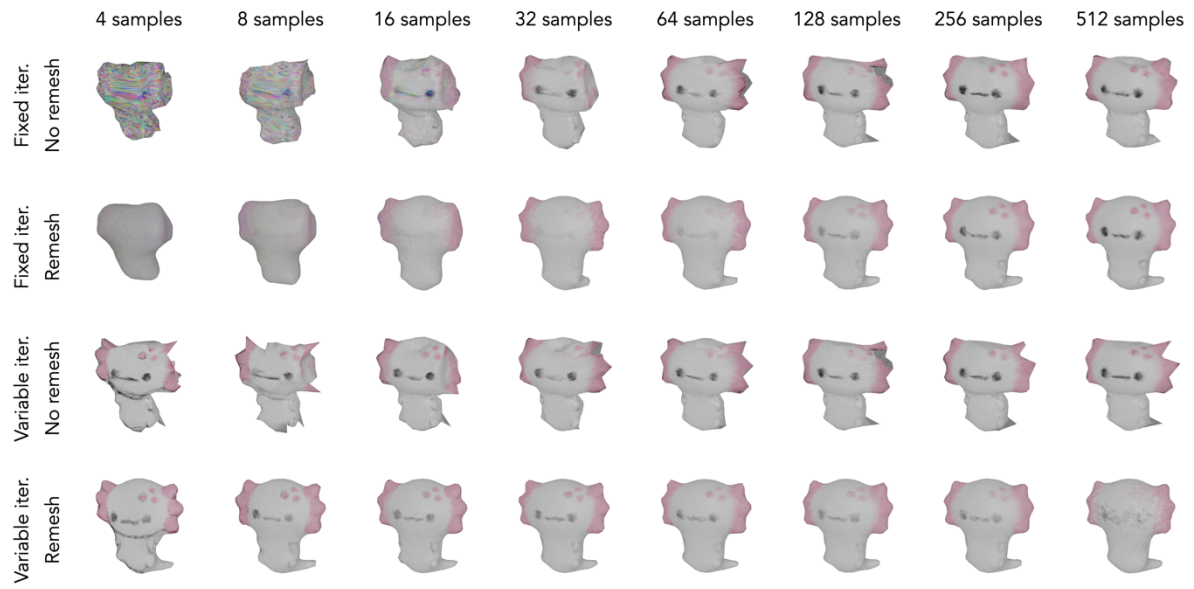


Figure 5. Reconstruction results over Axolotl with increasing numbers of samples.

on, we refer to the flat-colored lined sketches as the reference set.

Additionally, we generated modifications of the initial datasets to study the impact of quality factors. In each case, only the property under study was altered, keeping the rest of the parameters as the reference set.

Baseline results

We reconstructed each initial training set with remeshing and without it. Figure 3 shows the visual hull mesh initializations. In all cases, 30+3 iterations were used, meaning the last three only refined color. When using remeshing, it was applied every two iterations. We refer to this configuration as the reference configuration.

We evaluated the reconstructions using the validation sets and original models, measuring image metrics and average Chamfer distance. To compensate for scale mismatches, we measured both the pure distance and the distance after scaling the reconstruction to fit the largest dimension of the reference model.

Figure 4 and Table 1 show that the results present an acceptable quality. Better color estimation is observed from the style without lines, and remeshing is better for Axolotl than Vasque due to its rounded nature.

Quality factors study

Sketches contain noise and inconsistencies. Factors to consider are the number of available samples, their resolution, the precision of the masks, the consistency in the geometry between views, and the consistency between viewpoints and views. Here we summarize the results of their study while further details are provided in the complementary material.

The two most influential factors are the number of samples and the accuracy of the viewpoints provided

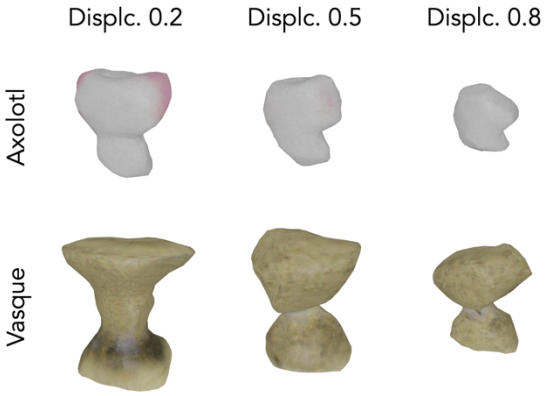


Figure 6. Results without remeshing of increasing levels of camera inaccuracy.

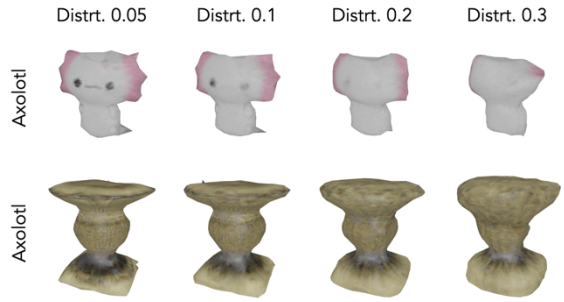


Figure 7. Results without remeshing of increasing levels of geometric inconsistency.

for the drawings. Training sizes from 4 to 512 samples were used for the former, as seen in Figure 5. For each size, we used both 30+3 iterations and an adapted number of iterations and remeshing steps to keep the number of updates constant. Results with and without remeshing were obtained. The reconstructions for Axolotl are shown in Figure 5, following a similar trend to the ones for Vasque. We observe how the

Model	Axolotl						Vasque					
Style	Color + lines		Only color		Only lines		Color + lines		Only color		Only lines	
Remesh.	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
MSE↓	0.007	0.007	0.005	**0.004**	0.007	0.006	0.026	0.027	**0.009**	0.010	0.033	0.031
PSNR↑	21.62	21.79	23.30	**23.99**	21.76	22.23	15.91	15.69	**20.57**	20.06	14.91	15.11
SSIM↑	0.923	0.926	0.934	**0.939**	0.925	0.929	0.792	0.789	**0.871**	0.870	0.790	0.789
LPIPS↓	0.109	**0.107**	0.140	0.139	0.147	0.149	**0.185**	0.197	0.191	0.190	0.246	0.247
Chamfer↓	0.057	0.083	**0.056**	0.082	0.057	0.081	0.079	0.113	0.076	0.108	**0.074**	0.119
Scaled Chamfer↓	0.014	**0.008**	0.013	**0.008**	0.013	**0.008**	0.025	0.034	0.023	0.030	**0.020**	0.037

Table 1. Results over 128 validation samples for reconstructions over 128 samples in different styles.

Model	Axolotl								
Set	B		NC		C6			CM	
System	Ours	Mun21a	Ours	Mun21a	Ours	Mun21a	Goe20a	Ours	Mun21a
MSE↓	0.006	**0.004**	**0.006**	0.007	0.008	**0.007**	0.033	0.020	**0.017**
PSNR↑	22.41	**23.70**	**22.61**	21.58	21.08	**21.67**	14.82	17.00	**17.83**
SSIM↑	0.924	**0.947**	0.925	**0.926**	0.917	**0.929**	0.907	0.912	**0.896**
LPIPS↓	0.104	**0.086**	0.142	**0.141**	0.107	**0.099**	0.128	0.133	**0.125**
Chamfer↓	0.068	**0.049**	0.065	**0.032**	0.058	**0.047**	0.048	0.033	**0.025**
Scaled Chamfer↓	0.005	**0.003**	**0.005**	0.016	**0.006**	0.010	0.064	**0.021**	0.022
Model	Vasque								
Set	B		NC		C6			CM	
System	Ours	Mun21a	Ours	Mun21a	Ours	Mun21a	Goe20a	Ours	Mun21a
MSE↓	**0.026**	0.033	0.030	**0.026**	**0.037**	0.043	0.095	0.067	**0.041**
PSNR↑	**15.96**	14.85	15.35	**15.88**	**14.41**	13.81	10.28	11.77	**13.87**
SSIM↑	0.791	**0.817**	0.793	**0.821**	0.779	**0.783**	0.746	0.757	**0.773**
LPIPS↓	**0.185**	0.195	0.239	**0.214**	**0.202**	0.214	0.251	0.260	**0.244**
Chamfer↓	0.073	**0.063**	0.077	**0.065**	0.056	0.086	**0.044**	0.136	**0.072**
Scaled Chamfer↓	0.024	**0.017**	0.024	**0.018**	**0.024**	0.031	0.086	0.110	**0.023**

Table 2. Validation results over 128 samples of the reconstructions with different systems (ours,

(NC), a set with six canonical views (C6), and the set with a camera displacement of 0.5 (CM).

quality increases with the number of samples and iterations. Good results are obtained with 16 and 32 samples and enough iterations, not improving significantly for more than 256 samples.

The camera inaccuracy was simulated by disturbing camera positions randomly so that cameras do not look at the mesh. Meanwhile, our system expects all views to look at it, effectively causing a discrepancy. We experimented with displacements of 0.2, 0.5, and 0.8. The results can be seen in Figure 6, showing how they quickly degrade. This is caused by the inconsistency of color and silhouette positions in the ground truth relative to the camera used for rendering. Consequently, results are averaged through the view space, reducing the mesh to what is consistently seen.

The rest of the factors were evaluated following similar procedures. A random scaling vector was applied to each mesh before rendering each sample, simulating geometric inconsistency. Magnitudes of scaling of 0.05, 0.1, 0.2, and 0.3 were evaluated. The

results, seen in Figure 7, show that the inconsistency reduces the quality, averaging all the seen shapes.

The resolution study showed that lower resolutions decrease the reconstruction quality, as seen in Figure 8. However, good results are obtained from 128 by 128 pixels onwards. High resolutions increase the quality mildly, mainly improving color.

Finally, the mask precision was studied by generating eroded and dilated masks. The results, seen in Figure 9, are only slightly worse, being the effects averaged while reducing their negative impact.

Sampling method

Figure 10 shows the results of different approaches for dealing with color shuffling after remeshing under the reference set and configuration. Comparing grey restart as in [Goe20a] with our proposal, the former washes out colors, being the grey tone still noticeable. Meanwhile, leaving the system to refine the colors automatically gives a close result to our approach. However, mismatched color patches and higher bleeding still appear due to the shuffling, while our system significantly reduces these effects.

Comparison with inverse-rendering-based reconstruction techniques

We compared our performance with Goel et al.'s system (SFT) [Goe20a] and NVDiffRec [Mun21a] to study how our proposal adapts to sketches compared with standard inverse rendering. Synthetic datasets were used to see how the systems react to adversities common in hand-drawn sketches. We evaluated the reconstructions for the already introduced reference set (B), lined sketches without color (NC), and set with a camera displacement of 0.5 (CM). Additionally, we used a set with the same properties as the reference set but with only six canonical views (C6).

The configurations were the following. Our system used 30+10 iterations, applying remeshing for Axolotl every two iterations and no remeshing for Vasque. The normal, shape, spring, and smooth regularizations were removed when using remeshing; further details are found in the ablation study in the complementary material. For NVDiffRec, we used 5000 iterations, a fixed white environment, a grid of 128, and the remaining default parameters. For SFT, we used our initial meshes, a limit of 2048 triangles, and 12 cycles. For the first 10, the iterations for material –diffuse color and roughness– and geometry were limited to 5 and 150, respectively. From that point, the limits were set to 75 and 300. Learning rates of 0.01 and 0.0005 were used for material and geometry.

The reconstructions were evaluated with the reference validation set, considering only diffuse colors. Table 2 displays the results. Due to time constraints, SFT was only evaluated for C6, as its execution time with the other sets was significantly higher. Nonetheless, the

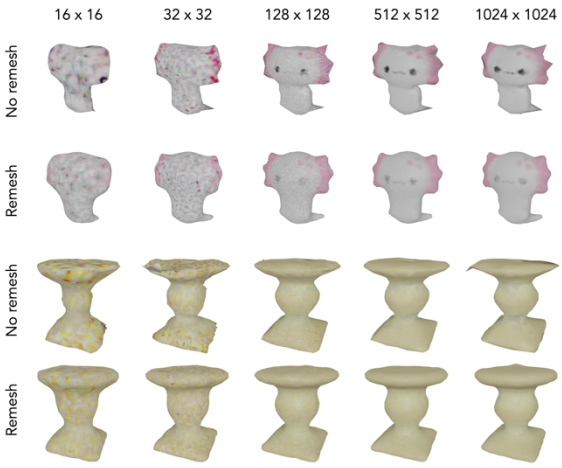


Figure 8. Results from different resolutions.

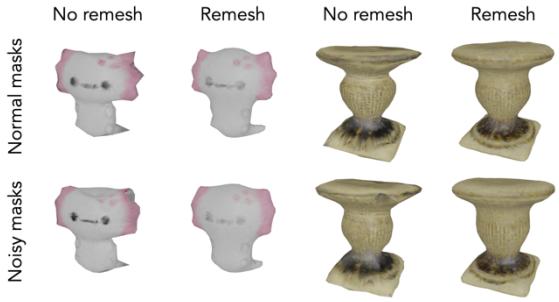


Figure 9. Results from altered masks.

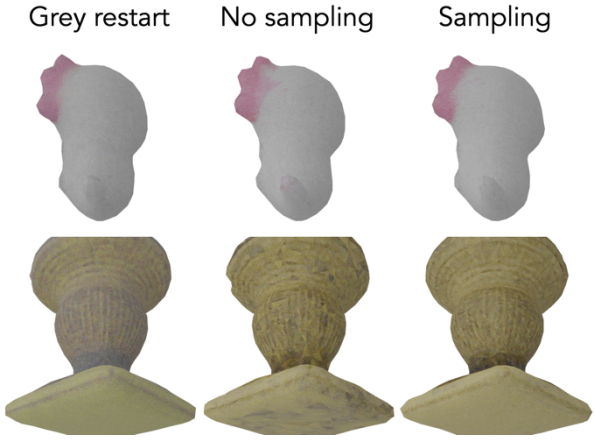


Figure 10. Comparison of the results obtained with different methods to fix color shuffling.

available results show how our system adapts better to sketches than SFT. This reflects that our split loss captures better the shape and color while our regularizations help guide the optimization. When comparing with NVDiffRec, we observe mixed results. It is important to note that we work with 807 triangles on average against 35037 for NVDiffRec. In the baseline case, we can see that NVDiffRec presents more accurate results thanks to a sharper shape and color estimation. However, in the case of Vasque, we also observe that image metrics are penalized. This is because NVDiffRec estimates specular properties,

which can interfere with the diffuse color. This was also reported in [Col22a] and further backs using only diffuse materials to avoid material ambiguity.

We observe better performance for our approach under a lack of color, as NVDiffRec generates distorted surfaces with holes like in [Col22a]. This is mainly seen in Axolotl due to the lower line density. In C6, NVDiffRec generates squared reconstructions while our approach preserves roundness and closer shape estimation. With CM, both fail, allowing our system a higher surface uniformity than NVDiffRec. Finally, thanks to the joint estimation, our proposal presented a temporal cost similar to NVDiffRec, taking two to three hours. Meanwhile, our experiments with SFT on the same hardware have shown times ranging from 10 hours to several days.

5. CONCLUSION

This paper has presented how inverse rendering techniques can be used for 3D reconstruction from multi-view sketches. Considering the inherent uncertainty of drawings, we proposed dedicated and tailored losses for shape and color, easing the use of joint optimization. Additionally, we presented a resampling method to partially restore colors after remeshing. Finally, we reported how the quality factors of sketches affect the system's performance.

The results have shown that our system obtains acceptable reconstructions with enough samples and consistency, being more robust and flexible than previous inverse rendering techniques to the lack of color and the use of canonical views. By designing the system to find approximations rather than replicas, we obtained a broader genericity at the cost of lower baseline performance.

We have seen how remeshing can solve degenerations but tends to round the shape, which is not always beneficial. Therefore, its use should be considered case by case, being most helpful with inconsistencies or few samples.

Nonetheless, there are still issues to cover. The large number of samples required to obtain good results is a limiting factor when considering real use cases. The same applies to the high dependency on the view poses and their accuracy. This dependency is inherent to a method based on inverse rendering, as cameras are a crucial component. Future work is required to increase the robustness and reduce the need for provided viewpoints and large amounts of samples.

In conclusion, we proposed the use of inverse rendering to recover shape and color from multi-view sketches, with the caveat of requiring viewpoint specification and enough informative samples for good results. Therefore, overcoming these caveats will be critical for practically applying the system in real applications and use cases with hand-drawn sketches.

6. REFERENCES

[Wan20a] J. Wang, J. Lin, Q. Yu, R. Liu, Y. Chen, and S. X. Yu. 3D Shape Reconstruction from Free-Hand Sketches. *CoRR*, vol. abs/2006.09694, 2020.

[Col22a] J. Colom, and H. Saito. 3D Shape Reconstruction from Non-realistic Multiple-view Depictions Using NVDiffRec. Asia-Pacific Workshop on Mixed and Augmented Reality 2022, Yokohama, Japan, 2022.

[Lun17a] Z. Lun, M. Gadelha, E. Kalogerakis, S. Maji, and R. Wang. 3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks. *CoRR*, vol. abs/1707.06375, 2017.

[Hei05a] W. Heidrich. Computing the Barycentric Coordinates of a Projected Point. *J. Graphics Tools*, vol. 10, pp. 9–12, Jan. 2005, doi: 10.1080/2151237X.2005.10129200.

[fel20a] felixyadomi. Cute Axolotl. 2020. URL: https://sketchfab.com/3d-models/cute-axolotl-e4625a288edf41afab1054a0fa529b3a

[Li18a] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 37, no. 6, p. 222:1-222:11, 2018.

[Xia22a] C. Xiao, W. Su, J. Liao, Z. Lian, Y.-Z. Song, and H. Fu. DifferSketching: How Differently Do People Sketch 3D Objects? arXiv, 2022, doi: 10.48550/ARXIV.2209.08791.

[Mun21a] J. Munkberg, J. Hasselgren, T. Shen, J. Gao, W. Chen, A. Evans, T. Müller, and S. Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. arXiv, 2021, doi: 10.48550/ARXIV.2111.12503.

[Nie04a] M. B. Nielsen, and A. Brodersen. Inverse rendering of polished materials under constant complex uncontrolled illumination. *WSCG*, vol. 12, no. 1-3, pp. 309-316, 2004.

[Gry20a] Y. Gryaditskaya, F. Hähnlein, C. Liu, A. Sheffer, and A. Bousseau. Lifting Freehand Concept Sketches into 3D. *ACM Trans. Graph.*, vol. 39, no. 6, Nov. 2020, doi: 10.1145/3414685.3417851.

[Yuk08a] C. Yuksel, J. Keyser, and D. H. House. Mesh Colors. Department of Computer Science, Texas A&M University, 2008.

[Hop93a] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh Optimization. *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1993, pp. 19–26, doi: 10.1145/166117.166119.

[Oht01a] Y. Ohtake, A. Belyaev, and I. Bogaevski. Mesh regularization and adaptive smoothing.

Computer-Aided Design, vol. 33, no. 11, pp. 789–800, 2001, doi: https://doi.org/10.1016/ S0010-4485(01)00095-1.

[Lai20a] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila. Modular Primitives for High-Performance Differentiable Rendering. arXiv, 2020, doi: 10.48550/ARXIV.2011.03277.

[Kim16a] K. Kim, A. Torii, and M. Okutomi. Multi-view Inverse Rendering Under Arbitrary Illumination and Albedo. in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 750–767.

[Bos20a] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. P. A. Lensch. NeRD: Neural Reflectance Decomposition from Image Collections. arXiv, 2020, doi: 10.48550/ ARXIV.2012.03918.

[Mil20a] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. arXiv, 2020, doi: 10.48550/ ARXIV.2003.08934.

[Zha21a] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron. NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination. *ACM Transactions on Graphics*, vol. 40, no. 6, pp. 1–18, Dec. 2021, doi: 10.1145/3478513.3480496.

[Kat17a] H. Kato, Y. Ushiku, and T. Harada. Neural 3D Mesh Renderer. arXiv, 2017, doi: 10.48550/ ARXIV.1711.07566.

[Wan21a] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. arXiv, 2021, doi: 10.48550/ARXIV.2106.10689.

[Gry19a] Y. Gryaditskaya, M. Sypesteyn, J. W. Hoftijzer, S. Pont, F. Durand, and A. Bousseau. OpenSketch: A Richly-Annotated Dataset of Product Design Sketches. *ACM Trans. Graph.*, vol. 38, no. 6, Nov. 2019, doi: 10.1145/3355089.3356533.

[Boj17a] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam. Optimizing the Latent Space of Generative Networks. arXiv, 2017, doi: 10.48550/ARXIV.1707.05776.

[Li22a] Z. Li, L. Wang, X. Huang, C. Pan, and J. Yang. PhyIR: Physics-based Inverse Rendering for Panoramic Indoor Images. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12703–12713, doi: 10.1109/CVPR52688.2022.01238.

[Zha21b] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. Snavely. PhySG: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting. arXiv, 2021, doi: 10.48550/ARXIV.2104.00674.

[Han20a] Z. Han, B. Ma, Y.-S. Liu, and M. Zwicker. Reconstructing 3D Shapes From Multiple Sketches Using Direct Shape Optimization. *IEEE Transactions on Image Processing*, vol. 29, pp. 8721–8734, 2020, doi: 10.1109/TIP.2020.3018865.

[Li18b] C. Li, H. Pan, Y. Liu, A. Sheffer, and W. Wang. Robust Flow-Guided Neural Prediction for Sketch-Based Freeform Surface Modeling. *ACM Trans. Graph. (SIGGRAPH ASIA)*, vol. 37, no. 6, p. 238:1-238:12, 2018, doi: 10.1145/3272127.3275051.

[Goe20a] P. Goel, L. Cohen, J. Guesman, V. Thamizharasan, J. Tompkin, and D. Ritchie. Shape From Tracing: Towards Reconstructing 3D Object Geometry and SVBRDF Material from Images via Differentiable Path Tracing. arXiv, 2020, doi: 10.48550/ARXIV.2012.03939.

[Li16a] C. Li, H. Lee, D. Zhang, and H. Jiang. Sketch-based 3D modeling by aligning outlines of an image. *Journal of Computational Design and Engineering*, vol. 3, no. 3, pp. 286–294, 2016, doi: 10.1016/j.jcde.2016.04.003.

[Zha21c] S.-H. Zhang, Y.-C. Guo, and Q.-W. Gu. Sketch2Model: View-Aware 3D Modeling from Single Free-Hand Sketches. arXiv, 2021, doi: 10.48550/ARXIV.2105.06663.

[Gao22a] C. Gao, Q. Yu, L. Sheng, Y.-Z. Song, and D. Xu. SketchSampler: Sketch-based 3D Reconstruction via View-dependent Depth Sampling. arXiv, 2022, doi: 10.48550/ ARXIV.2208.06880.

[Liu19a] S. Liu, T. Li, W. Chen, and H. Li. Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning. arXiv, 2019, doi: 10.48550/ARXIV.1904.01786.

[Sch16a] J. L. Schönberger and J.-M. Frahm-Structure-from-Motion Revisited. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113, doi: 10.1109/CVPR.2016.445.

[Wan18a] L. Wang, C. Qian, J. Wang, and Y. Fang. Unsupervised Learning of 3D Model Reconstruction from Hand-Drawn Sketches. *Proceedings of the 26th ACM International Conference on Multimedia*, New York, NY, USA, 2018, pp. 1820–1828, doi: 10.1145/3240508.3240699.

[Fre07a] Fredo6. Vasque. 2007. URL: https://3dwarehouse.sketchup.com/model/de673ddf9df03b8278cf1a714198918/Vasque-in-Sketchu

ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.31, No-1-2, 2023

Bias mitigation techniques in image classification: fair machine learning in human heritage collections

Dalia Ortiz Pablo
Centre for Digital Humanities Uppsala
Department of ALM
Uppsala University
Sweden 75126, Uppsala
dalia.ortiz_pablo@abm.uu.se

Sushruth Badri **Erik Norén** **Christoph Nötzli**
Department of Information Techology
Uppsala University
Sweden 75126, Uppsala
{sushruth.badri.6580 | erik.noren.3194 |
christoph.notzli.7006}@student.uu.se

Abstract

A major problem with using automated classification systems is that if they are not engineered correctly and with fairness considerations, they could be detrimental to certain populations. Furthermore, while engineers have developed cutting-edge technologies for image classification, there is still a gap in the application of these models in human heritage collections, where data sets usually consist of low-quality pictures of people with diverse ethnicity, gender, and age. In this work, we evaluate three bias mitigation techniques using two state-of-the-art neural networks, Xception and EfficientNet, for gender classification. Moreover, we explore the use of transfer learning using a fair data set to overcome the training data scarcity. We evaluated the effectiveness of the bias mitigation pipeline on a cultural heritage collection of photographs from the 19th and 20th centuries, and we used the FairFace data set for the transfer learning experiments. After the evaluation, we found that transfer learning is a good technique that allows better performance when working with a small data set. Moreover, the fairest classifier was found to be accomplished using transfer learning, threshold change, re-weighting and image augmentation as bias mitigation methods.

Keywords

Image classification, Fairness, Bias mitigation, Gender classification, Transfer learning, Human Heritage Collection.

1 INTRODUCTION

Artificial intelligence (AI) systems have become a key instrument in many human decision processes. Their presence ranges from basic day-to-day tasks such as listening to music at home using technologies the size of a donut [McL19] to transcribing hand-written characters into machine-actionable text data [Cor20]. Those systems present clear benefits, mostly thanks to computers being able to perform tasks at a velocity that humans cannot and without getting tired. However, not all the outcomes are positive with AI. One major issue that those algorithms have presented is bias and unfairness. For example, the recruiting algorithm developed by Amazon, where the system learnt key traits from successful applicants' resumes to rate and find the top new candidate' CVs, exhibited a preference for males in technical positions [Kod19]. This problem does not only occur in contemporaneous sets but also historical scenarios.

Galleries, archives and museums carry deep insights into human memory and expression. Therefore not only collecting the remains of the past is relevant, but also analysing the objects that have been obtained. Furthermore, it is possible to apply and evaluate the technologies of the present, such as image classification, on the remnants of the past to understand how we can have more accurate and meaningful descriptions and classifications of heritage collections. Moreover, it is crucial to examine the ways in which qualitative aspects of human nature, such as bias, and the subjective nature of AI, intersect. This is especially relevant in the case of museum artifacts collections, which tend to exhibit inherent biases due to factors such as the methods of acquisition and digitization [Kiz21].

Literature on bias in AI mentions that bias is often encoded in the training data set and that this type of bias affects especially the models' accuracy [Par18]. However, even with well-balanced data sets, bias can be introduced in other steps of the ML pipeline. The work of

Wang et al. [Wan19] in gender classification is a good illustration of this. The authors showed that even when their data sets were perfectly balanced, the trained models resulted in biased predictions. Thus, assuming that the input data set is the only source and actor of bias is erroneous. In other studies, researchers have identified, named and classified many other sources of bias, for example, algorithmic bias, which appears due to inappropriate algorithmic choices, and is not present in the input data, or evaluation bias, which happens when the evaluation data set is not representative of the problem or the evaluation metric is incorrect for the task [Fah21, vanG22, Meh21].

The objective of this study is to investigate the effectiveness of algorithmic bias mitigation techniques for gender classification on a museum data set, thus bridging the gap between AI and gender classification in cultural heritage collections. Specifically, we aim to evaluate the performance of modern classification approaches and bias mitigation techniques on a highly diverse and gender-biased data set consisting of low-quality images of ancient people with varying ethnicity and ages. To the best of our knowledge, no studies have yet implemented these techniques for this purpose.

2 RELATED WORK

2.1 Bias mitigation

Technical bias mitigation techniques can be divided into five stages: problem understanding, preprocessing, in-processing, post-processing and deployment [Hor22, Bel18]. Bias mitigation methods performed on the training data are considered preprocessing; techniques performed while training ML models are categorised as in-processing; and post-processing methods are applied to trained ML models [Hor22, Bel18].

One of the most common techniques for bias mitigation in the pre-processing step is data augmentation - a technique used to increase the amount of data in the training set by performing modifications (e.g. rotations, colour transformation, reflection) on the original set [Mij18]. In the work done by McLaughlin et al. [McL15], the authors evaluated the effectiveness of different data augmentation techniques in re-identification systems. They found that changing an image background increases the performance of a model only when a combination of other augmentation techniques, such as cropping and mirroring, are also used.

Other bias mitigation methods have also been investigated in the literature. The work of Wang et al. [Wan20] evaluates strategic re-sampling, adversarial training, domain discriminative training, and domain-independent training in a gender classification scenario using CNNs and a data set composed of pictures of celebrities. The authors found that oversampling outperformed the other techniques, and that was followed closely by domain-independent training. Similarly, Lee et al. [Lee22] revise bias mitigation methods for CNNs, e.g. ReBias and vanilla, to create a benchmark for the bias mitigation pipeline. The study showed that state-of-the-art approaches achieved different approaches depending on the training data set, which suggests that a bias mitigation process is task specific.

2.2 The FairFace data set

Neural network models have been shown to learn and amplify biases in training data [Hal22]. This is partly the motivation for the creation of the FairFace data set, presented in the work by Kärkkäinen and Joo [Kar19]. The FairFace data set contains 108 501 images, and it is balanced concerning gender, ethnicity, and age, and therefore does not suffer from the same bias that other big data sets do. In [Kar19], it is shown that models trained with the FairFace data set generalize better than other existing face data sets to unseen and ethnically diverse data. In the work of Kotti et al. [Kot22], the data set is used in their experiments for evaluating bias in Generative Adversarial Networks (GANs). Moreover, a model trained on the FairFace data set was used in [Dev22] in order to benchmark the performance of their new fair model. In our work, we conduct transfer learning using the FairFace data set for the models to learn first for a known fair data set.

3 BACKGROUND

3.1 Deep learning models

Our classifiers are based on existing networks provided in the tensorflow framework. We chose Xception [Chol17] and EfficientNet [Tan19] as base networks. Both were implemented using Keras' model API.

The Xception network was first introduced in 2017 in the paper "Xception: Deep Learning with Depthwise Separable Convolutions" by Chollet [Chol17]. The model is a fully convolutional network designed with the goal of being a more efficient variant of the Inception architecture from 2014 [Sze15]. The technique that separates the Xception architecture from the Inception architecture is the implementation of depthwise separable convolutions instead of regular convolutions. Regular convolutions work by performing convolutions on all channels at once, unlike depthwise separable convolutions which performs a single convolution operation on each input channel. The Xception architecture has 36 convolutional layers and has an input size of 299x299.

EfficientNet was introduced in the 2019 paper "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" [Tan19]. Rather than being a single network, EfficientNet is best described as a family of

networks architectures, created from scaling the baseline network EfficientNet-B0. The paper introduces a new approach to scaling models which included scaling the network's width, depth and image resolution together. The EfficientNet architecture that we implemented was EfficientNet-B3 which has an input size of 300x300 which is similar to Xception. In [Tan19], the EfficientNets' performances, compared to other convolutional neural networks, is the state-of-the-art of several data sets, while reducing the size of the models. The EfficientNet-B3 architecture is about half the size of Xception, with 12 million versus 23 million parameters.

Both the models have the same four top layers appended to produce the binary classification. These consists of a GlobalAveragePooling2D layer, a BatchNormalization layer, a Dropout layer with a dropout rate of 0.2, and lastly, a dense layer for binary classification with a sigmoid activation function [Keras].

3.2 Fairness metrics

Bias quantification metrics are closely linked to the concept of *fairness*, which has two main definitions. The first definition is individual fairness, which involves assessing similar individuals and expecting them to be treated similarly by the model [Cho17]. The second definition is group fairness, which refers to the absence of prejudice and favoritism towards a particular group [Meh21]. Further, there are two common categories of fairness metrics related to the previous definitions: *Definitions Based on Predicted Outcome* and *Definitions Based on Predicted and Actual Outcome*. Some examples of fairness metrics that belong to these categories are:

- Demographic (or statistical) Parity Difference (DPD), which focuses on ensuring that there are similar amounts of positive predictions across groups. In this context, a classifier is called fair if [Ver18]:
$$\mathrm{TP}+\mathrm{FP}=\mathrm{TN}+\mathrm{FN} \tag{1}$$
- Proportional Parity Difference (PPD) is a normalized version of DPD [Koz21]. In the binary case a classifier is called fair if:
$$\frac{\mathrm{TP}+\mathrm{FP}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}}=\frac{\mathrm{TN}+\mathrm{FN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}} \tag{2}$$
In a problem with more than two classes, the denominators in equation 2 would be different, resulting in different scaling factors for each class. However, in the binary case, the denominators are the same. In our work, we treated them as different because we did not explicitly account for the normalization constant in any of the experiments.
- Equality of Opportunity. In a binary classification task, it aims to ensure that both groups have equal rates of true positives. Definitions are based on Predicted and Actual Outcome, and in this case a classifier is fair if [Gar20]:
$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}=\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \tag{3}$$
- Predictive Rate Parity Difference. This measure, in a binary classification task, ensures that both groups have equal rates of predicted positives [Cho17]. The definition is based on Predicted and Actual Outcome [Ver18]. A classifier is fair if: In the binary case:
$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}=\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FN}} \tag{4}$$

This metrics are popular in the general area of fair machine learning and can be applied in image classification as well. Examples of this are Demographic Parity used in [Fra21] and Equality of Opportunity used in [Sto22, Zha18]. Therefore, in this project, we use demographic parity difference, equality of opportunity, proportional parity difference and predictive rate parity difference as fairness metrics.

3.3 Bias mitigation techniques

3.3.1 Reweighting

Reweighting is a pre-processing bias mitigation method. The idea of this method is to give classes that are more common in the training data set a lower weight i.e. the sample has less effect on the training of the model. Reweighing approach also maintains a high accuracy level [Kam12]. Pre-processing techniques try to transform the data so that the underlying discrimination is removed [Meh21]. The class weights are assigned and passed to the model while training through Keras implementation.

3.3.2 Image augmentation

Image augmentation is an in-processing method for bias mitigation. In this project, data augmentation is implemented by altering the training data in order to deal with classification bias in under-representation of certain groups. Data augmentation can reduce classification error for discriminated groups. Furthermore, even though different classifiers do not perform equally good, they exhibit positive results when data augmentation takes place [Ios18]. In our project the data augmentation has four pre-processing layers that are added at the beginning of the model. These layers perform random flip, random rotation, random translation (i.e. random movement), and random contrast. The pre-processing layers are implemented using the Keras API for image augmentation layers [Keras].

3.3.3 Threshold change

Changing the threshold of the model is a post-processing method [Hor22]. In the standard implementation of our model we use a threshold of 0.5 i.e. every predicted value below 0.5 is interpreted as Female. This threshold is not necessarily optimized for fairness. We applied the following threshold changes to the models:

Equal true In this case, we optimize the threshold to the minimal difference of predicted true values in each class:

$$\min\{|\mathrm{TP}_t - \mathrm{TN}_t|\} \quad \text{with t} = [0,1], \tag{5}$$

where t, here and in the following definitions, is the value of the threshold for the classifier.

Equal false In this case we optimize the threshold to the minimal difference of predicted false values in each class.

$$\min\{|\mathrm{FP}_t - \mathrm{FN}_t|\} \quad \text{with t} = [0,1] \tag{6}$$

Equal total In this case we optimize the threshold to predict the minimal difference of predicted values in each class.

$$\min\{|(\mathrm{TP}_t + \mathrm{FP}_t) - (\mathrm{TN}_t + \mathrm{FN}_t)|\} \quad \text{with t} = [0,1] \tag{7}$$

Equal opportunity In this case we optimize the threshold to predict the minimal difference of predicted values in each class.

$$\min\{|\frac{\mathrm{TP}_t}{\mathrm{TP}_t + \mathrm{FN}_t} - \frac{\mathrm{TN}_t}{\mathrm{TN}_t + \mathrm{FP}_t}|\} \quad \text{with t} = [0,1] \tag{8}$$

3.3.4 Transfer Learning

Transfer learning is the ML technique of taking the knowledge that is able to be learned by training a model on one task and then fine tuning it to a different but related task [Pan10]. In this project, transfer learning was implemented as a way overcome training data scarcity. For this part of the process, the FairFace data set is the source domain and the cultural heritage data set is the target domain. The tasks are similar in both domains, being binary image classification of gender in both cases.

Through our experiments, it will be investigated if it is possible to transfer a fair model trained on the FairFace data set. This will be done by comparing the results of the models implementing transfer learning against the ones trained on only target domain. The metrics for evaluating bias will be fairness metrics detailed in section 3.2 and section 3.3.

3.4 Cultural Heritage Data Set

The Cultural Heritage Data Set (CHDS) contains labelled images from the The National Museums of World Culture [VKM] in Sweden, and the data set was manually post-processed as part of an earlier part of the Quantifying Culture project. Within the data set there are 3128 images in total of people, where 1067 images are labeled male, and 2061 labeled female (a sample of the data set is shown in Figure 1). These images are photographs taken "in-the-wild" and do not guarantee that they contain only one person at a time, but ensure that all the individuals are of the same sex. The time period which the photographs stem from are either the nineteenth or twentieth century. After performing the face extraction described in section 4.1, 4897 faces were detected, 2331 labeled male, and 2566 labeled female.

Figure 1: Example of original image from the Cultural Heritage Data Set [VKM] containing two individuals labeled males.

4 IMPLEMENTATION DETAILS

4.1 Face extraction

Since the CHDS contains full-body images of people, and our model is designed to perform gender classification based on facial features, face extraction is performed on the CHDS. The face extraction module makes use of the Python API of the dlib ML toolkit [Kin09]. Within the module a pre-trained CNN model is used for face detection which then allows for extracting cropped images of the faces contained in the photographs of the CHDS.

4.2 Experimental setup

We use three experimental setups to test our networks. In the first experiment, we used only the CHDS to train and test our networks. In a second step, we used the FairFace data to pretrain validate and test them our models. The third experiments use the FairFace models from the second experiments for transfer learning. The transfer learned models are then validated, tested

and compared to the baseline models. For all experiments the fairness metrics discussed in section 3.2 are implemented for evaluation.

4.2.1 *Baseline CHDS experiments*

In the baseline model experiments we train and cross validate the models shown in section 3.1. For each of these two models, there are three variations tested, which depend on the bias mitigation method being evaluated. The first variation is without any bias mitigation method; the second is with class re-weighting applied; and the third is with data set augmentation applied. These bias mitigation methods are covered in section 3.3. The models were trained for a maximum of 20 epochs with the learning rate decreasing exponentially. We added an early stopping to the training, which means that the training will stop in case the training does not decrease the validation loss in three consecutive epochs.

4.2.2 *FairFace experiments*

The models described in section 3.1 are pretrained with the help of the FairFace data in three different ways. First we just train the models, in a second experiment we reweight the classes because there is a slight bias towards the Male class in the FairFace data. In the third experiment we add the augmentation layers described in section 3.3. These experiments are run to create the different variations of base models to be used for transfer learning, as well as a way to evaluate performance on the models on a data set that we know to be fair and balanced.

4.2.3 *Transfer learning experiments*

To build some the classifiers for this project, we train a base model, either EfficientNet-B3 or Xception, with the FairFace training data. During training, the weights of the models' layers are tuned and adjusted in order to increase accuracy. This trained model is then moved to the target domain and has a number of its layers and their parameters frozen in order to keep the knowledge learned in the source domain. For transfer learning with the EfficientNet-B3 model all layers except the the last forty layers are frozen. In the Xception case all layers except the last ten layers and the four top layers are frozen. Also, we compare four different version of the transfer learned models. We use the unweighted and unaugmented pretrained model to test transfer learning.

4.2.4 *Data set splits*

As mentioned in section 3.4 the cultural heritage data set does not contain a lot of data. Therefore we choose to use 80% of the data for training, 10% for validation and 10% for testing. We used the same splits for the training with the help of the FairFace data set. For cross validation we used a 5-Fold cross validation split, 80% of data for training and 20% for validation.

4.2.5 *Early stopping*

We use early stopping provided by Keras [Keras]. I.e. the training stops when the validation loss is not reduced in three consecutive training steps.

4.2.6 *Used infrastructure for training*

For the training of the models we used the Alvis cluster of Chalmers University [Alvis]. We used the A100 GPUs of the cluster. This allowed us to use larger batch sizes and smaller training time.

5 RESULTS

The results of our experiments can be found in this section. Within table 1 the results of the Baseline CHDS experiments and the Transfer learning experiments can be seen. Positive values in the fairness metrics Demographic Parity Difference, Proportional Parity Difference and Predictive Rate Parity Difference show a bias in favor of the Male class. In the Equality of Opportunity metric the bias is in favor of the Male class if the values are negative.

5.1 Baseline CHDS experiments

Baseline CHDS experiments were performed for both the EfficientNet-B3 and Xception models. Figure 2 shows the performance in accuracy for EfficientNet and Xception. It can be seen that all three variations of the EfficientNet model reaches higher accuracy than the Xception variations. Figure 3 shows the training and validation accuracies for the Xception variations. The EfficientNet variations also have greater performance in the fairness metrics as well as seen in table 1.

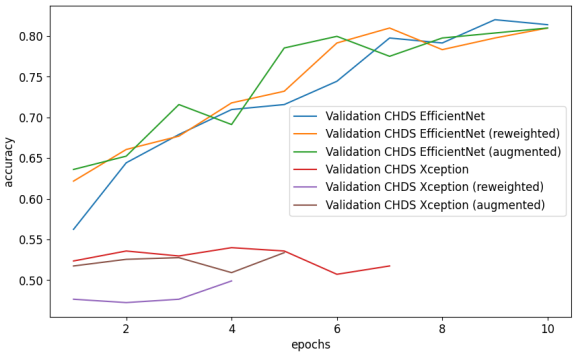


Figure 2: EfficientNet and Xception validation accuracy for the CHDS experiments.

5.2 FairFace experiments

The model that achieves the highest performance in accuracy on the FairFace data set is EfficientNet with a validation accuracy of ∼91% as seen in Figure 4. The best performing Xception model is the one implemented with augmentation which achieves a validation accuracy of ∼87%. Figure 5 shows the performance of the models with regards to Demographic Parity Difference where all variations except Xception with

Network name	Transfer learning	Threshold change	Reweighting	Image augmentation	Accuracy	Demographic Parity Difference	Proportional Parity Difference	Equality of Opportunity	Predictive Rate Parity Difference
EfficientNet	No	No	No	No	78.5 +/- 1.2%	-39 +/- 54.6	-0.04 +/- 0.06	0.01 +/- 0.06	-0.03 +/- 0.05
EfficientNet	No	Equal false	No	No	78.9%	4	0.008	0.036	-0.053
EfficientNet	No	No	Yes	No	80.4 +/- 1.6%	-64 +/- 33.8	-0.06 +/- 0.03	0.04 +/- 0.03	-0.01 +/- 0.03
EfficientNet	No	Equal total	Yes	No	79.9%	6	0.012	0.041	-0.056
EfficientNet	No	No	No	Yes	78 +/- 2.2%	-155 +/- 77.8	-0.16 +/- 0.08	0.13 +/- 0.09	0.04 +/- 0.06
EfficientNet	No	Equal opp.	No	Yes	80.1%	-8	-0.016	0.013	-0.039
EfficientNet	Yes	No	No	No	84.5 +/- 0.5%	-45 +/- 42.3	0.05 +/- 0.04	0.01 +/- 0.04	0.02 +/- 0.03
EfficientNet	Yes	Equal false	No	No	83.7%	-8	-0.016	0.017	-0.038
EfficientNet	Yes	No	No	Yes	83.6 +/- 0.3%	-115 +/- 26.2	-0.11 +/- 0.02	-0.08 +/- 0.04	0.03 +/- 0.04
EfficientNet	Yes	Equal false	No	Yes	82.1%	12	0.024	0.056	-0.064
EfficientNet	Yes	No	Yes	No	84.2 +/- 1.5%	-101 +/- 73.5	-0.1 +/- 0.07	0.07 +/- 0.08	0.02 +/- 0.06
EfficientNet	Yes	Equal false	Yes	No	80.7%	14	0.028	0.059	-0.066
EfficientNet	Yes	No	Yes	Yes	83.9 +/- 1.4%	-13.4 +/- 48.2	-0.01 +/- 0.05	0.02 +/- 0.04	-0.04 +/- 0.03
EfficientNet	Yes	Equal total	Yes	Yes	82.1%	0	0	0.031	-0.049
Xception	No	No	No	No	52.4%	-492	-1	-1	-0.524
Xception	No	No	Yes	No	47.6%	492	1	1	0.476
Xception	No	No	No	Yes	52.4%	-492	-1	-1	-0.524
Xception	Yes	No	No	No	65.6 +/- 3.7%	-7 +/- 435	-0.01 +/- 0.44	-0.01 +/- 0.45	-0.06 +/- 0.15
Xception	Yes	Equal total	No	No	78.3%	2	0.004	0.032	-0.051
Xception	Yes	No	No	Yes	72.4 +/- 0.1%	-338 +/- 401	-0.34 +/- 0.40	-0.33 +/- 0.41	0.1 +/- 0.16
Xception	Yes	Equal opp.	No	Yes	80.9%	-16	-0.033	-0.002	-0.029
Xception	Yes	No	Yes	No	66.4 +/- 6%	-341 +/- 398	-0.35 +/- 0.41	-0.34 +/- 0.41	0.09 +/- 0.19
Xception	Yes	Equal opp.	Yes	No	77.4%	-14	-0.028	-0.002	-0.033
Xception	Yes	No	Yes	Yes	73.1 +/- 7.8%	-389 +/- 169	-0.39 +/- 0.17	-0.37 +/- 0.19	0.15 +/- 0.08
Xception	Yes	Equal opp.	Yes	Yes	81.3%	4	0.008	0.039	-0.054

Table 1: Validation results of the experiments. The results of the CHDS experiments are the entires marked with "No" in the transfer learning column. Entries marked "Yes" in the Transfer learning column are implemented with transfer learning from the FairFace data set. In the table the performance of different combinations of the different bias mitigation methods can be seen.

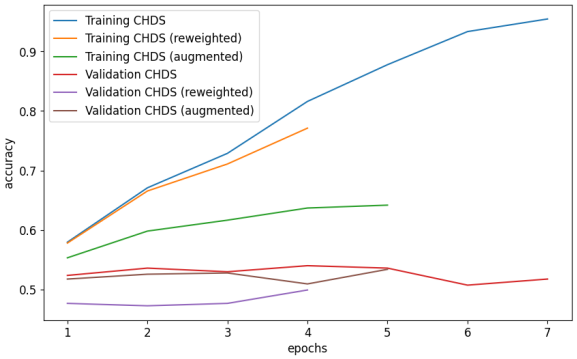


Figure 3: Xception training and validation accuracy for the CHDS experiments.

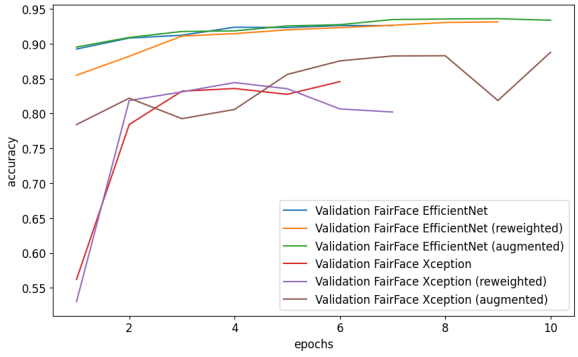


Figure 4: EfficientNet and Xception validation accuracy for the FairFace experiments.

reweighting have similar results. This is also the case when evaluating with regards to Equality of Opportunity Difference, as seen in Figure 6.

5.3 Transfer learning experiments

Using the models created in the FairFace experiments for transfer learning, the accuracy results shown in table 1 were achieved. EfficientNet achieves the highest accuracy of the transfer learned models with 83.7%.

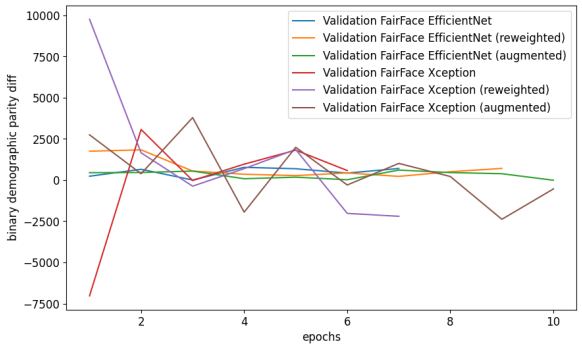


Figure 5: EfficientNet and Xception validation Demographic Parity Difference performance for the FairFace experiments.

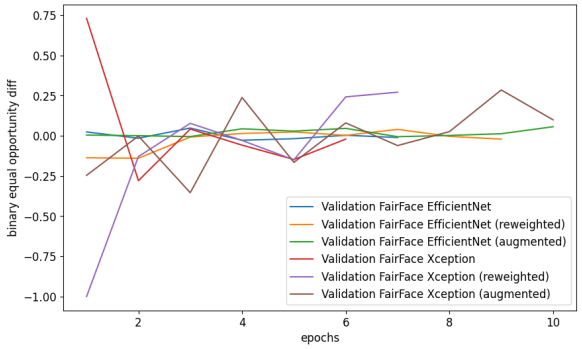


Figure 6: EfficientNet and Xception validation Equality of Opportunity performance for the FairFace experiments.

In the fairness metrics different transfer learning variation of both EfficientNet and Xception have the best performance. Measured in Demographic Parity Difference the EfficientNet model with equal total threshold change, re-weighting, and image augmentation performs best. If Equality of Opportunity is considered instead the re-weighted Xception model is best performer. When using augmentation and the EfficientNet network the bias metrics stay close to 0 during the training as seen in Figure 8 and Figure 7.

6 DISCUSSION

The Xception network is not ideal for a use case with a small data set, in this case the CHDS. In table 1 the results show that the Xception network is not performing well due to overfitting as seen in Figure 3. More regularization within the network and the final layers could be a possible solution to fix the overfitting in the base case of the Xception model. However, here the value of transfer learning is shown to be an effective way to improve a model which poor performance is partly due to lack of training data. Xception used with transfer learning allowed for a validation accuracy of 73.1%, which greatly outperforms the baseline CHDS trained models as seen in Table table 1.

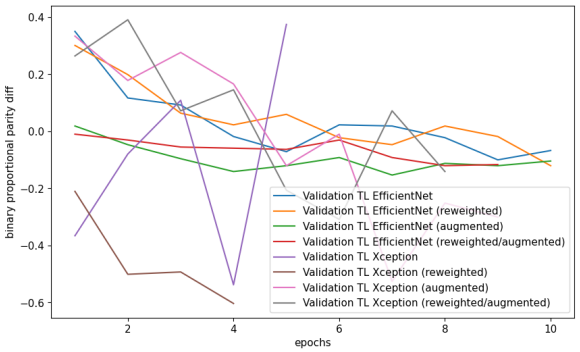


Figure 7: EfficientNet and Xception validation Proportional Parity Difference for the transfer learning experiments.

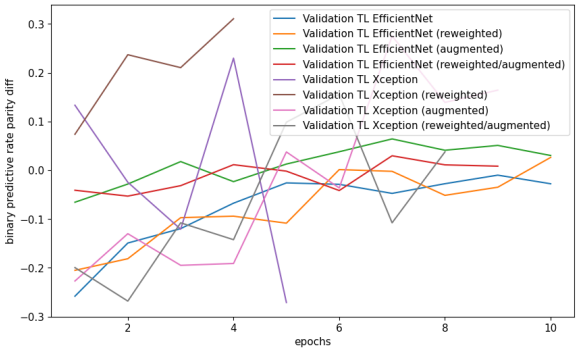


Figure 8: Xception training and validation Predictive Rate Parity Difference for the transfer learning experiments.

The results in section 5 show that EfficientNet has a higher accuracy and is better than Xception in regard to fairness in the transfer learning variations. These results are expected since EfficientNet is a newer network and also shows better performance in the benchmarks [Tan19].

Using transfer learning to train the models is a good choice in our use case. The biggest advantage is that we get a higher accuracy due to training the models with more data. Comparing the EfficientNet model, with or without transfer learning, it can be seen in table 1 that accuracy in increased from 78.5% to 84.5%. However, as a method for bias mitigation transfer learning does not make much of an impact. For the experiments on the EfficientNet variations the fairness metrics do not appear to be influenced by applying transfer learning or not. Due to the poor performance of the baseline variations of Xception, it is not possible to see if previous results translate to the Xception model.

When transfer learning and augmentation are used together with reweighting on EfficientNet, the fairness metrics are improved. This is also the case for accuracy as seen in table 1. Using reweighting alone does not show any significant improvement in the metrics for both EfficientNet and Xception. It could be due to the

fact that there is not a big difference in the amount of samples in each of the classes in CHDS.

Comparing the EfficientNet CHDS results in table 1, with and without image augmentation, it can be seen that accuracy is not changed much, the accuracy score being 78.5% +/- 1.2% without and 78% +/- 2.2% with. Moreover, when evaluating these same two variations by the fairness metrics no apparent improvement can be seen in any case. This pattern emerges in the transfer learning experiments as well for EfficientNet. Using image augmentation alone slightly worsens performance in terms of accuracy and improvement cannot be seen in the fairness metrics either. However, when comparing the transfer learning variations, with and without image augmentation, with reweighting implemented, the pattern does not repeat. In this case image augmentation improved performance in all accord with all fairness metrics while reaching a perfect score for Demographic- and Proportional Parity. For transfer learning experiments for Xception, image augmentation improves the fairness metrics when paired together with reweighting, similarly to EfficientNet. Image augmentation improves accuracy for the Xception variations. One important observation is that augmentation stabilizes the training (metrics always close to the ideal value) for EfficientNet as seen in Figure 8 and Figure 7. This is advantageous because we can stop the training at any point with similar bias metric values.

When applied, threshold change improves performance in the fairness metrics consistently. In all EfficentNet experiments the improvement in fairness came at the cost of accuracy, this is however not translated in the Xception experiments. It could be because of Xception network overfitting on the data and it favours a certain class during prediction. This can be seen from the metric values in 1. When using threshold change, some of the wrong predictions move over to the other side of the decision boundary and so accuracy improves.

Regarding the CHDS data set, it shows a slight bias favouring the Female class (52.4% of the images in the Female class). This is also the case for our overall best model (EfficientNet, with reweighting and image augmentation but no thresholding). For example, the value of bias according to the Proportional Parity Difference is 1 +/- 5% towards the Female class. Gender is something that should be considered concerning fairness; however, there is a cause to consider fairness in relation to other metrics as well. With the CHDS data set only having gender as its label, further investigation into other sources of bias is difficult to attain. The CHDS data set is diverse concerning age, ethnicity, culture, and the period the photo was taken. All of these aspects add the possibility for bias. Photographs portraying a given group of people of a certain ethnicity could have been taken in poorer conditions than photographs of another group due to the time period or other external conditions. As a result, this could lead to it being harder for the model to learn to classify the first group correctly, which might not show up in our results evaluated by gender.

Due to using a pre-trained model in the face extraction module for the cultural heritage data set (section 3.4), there is an additional potential source of bias. For example, it is possible that the pre-trained model was itself trained with biased data, thus leading to it potentially having a better ability to extract the faces of a certain group. Since we did not investigate the total amount of faces belonging to each class in the photographs in the CHDS, no evaluation of the potential bias of the face extraction module was performed.

7 CONCLUSION

In this paper, we evaluated three novel bias mitigation techniques for image classification in a cultural heritage data set. We found that individually implementing augmentation, class re-weighting, and threshold change does not lead to a fairer model compared to the baseline model. We have also evaluated the performance of the classifiers when implementing transfer learning. We have shown that, for this task, combining transfer learning with image augmentation, class re-weighting, and threshold change is the best way to reach a fair classifier.

8 FUTURE WORK

In our current implementation the process of fine tuning was not included in the transfer learning implementation. In fine tuning the final models are trained, with all layers unfrozen, with a very low learning rate for just a few epochs. This could potentially improve performance, mainly accuracy.

Further work could be done in tuning the hyperparameters of our models. Since the priority of this project was the evaluation of the bias mitigation methods, less focus was put on perfecting model performance. Therefore work can be done in investigating optimal learning rate, batch size, weight decay etcetera.

As discussed in section 6, improvement to our analysis could be done if we would have had access to multiple labels, e.g. age, ethnicity, etcetera. With more labels we could take a look a the bias from an intersectional stand point. With the help of intersectionality we could show further biases in the approach, and we could take action to reduce these biases.

Another interesting addition to the project would be to compare the results of the EfficentNet with an implementation of the EfficientNetV2. EfficientNetV2 includes data augmentation layers and we had promising preliminary results for the base model.

9 ACKNOWLEDGMENTS

This work has been partially supported by the WASP-HS (Autonomous Systems and Software Program-Humanities and Society) grant - an initiative of the Wallenberg Foundations; project title: "Quantifying Culture: AI and Heritage Collections"; project number: MAW 2020.0054. The computations/data handling were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at Chalmers Centre for Computational Science and Engineering (C3SE) partially funded by the Swedish Research Council through grant agreement no. 2018-05973, project Dnr: SNIC 2022/22-1091.

10 REFERENCES

[All22] Allawala, A., Ramteke, A., & Wadhwa, P. Performance Impact of Minority Class Reweighting on XGBoost-based Anomaly Detection. International Journal of Machine Learning and Computing, 12(4), 2022.

[Alvis] "Chalmers Centre for Computational Science and Engineering": https://www.c3se.chalmers.se/about/Alvis/.

[Bel18] Bellamy, R. K., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., ... & Zhang, Y. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. arXiv preprint arXiv:1810.01943, 2018.

[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.

[Cho17] Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. Big data, 5(2), 153-163, 2017.

[Cho20] Chouldechova, A., & Roth, A. A snapshot of the frontiers of fairness in machine learning. Communications of the ACM, 63(5), 82-89, 2020.

[Con00a] Conger., S., and Loch, K.D. (eds.). Ethics and computer use. Com.of ACM 38, No.12, 2000.

[Cor20] Cordell, R. Machine learning and libraries: a report on the state of the field. Library of Congress, 2020.

[Dev22] Deviyani, A. Assessing Dataset Bias in Computer Vision. arXiv preprint arXiv:2205.01811, 2022.

[Dro20] Drozdowski, P., Rathgeb, C., Dantcheva, A., Damer, N., & Busch, C. Demographic bias in biometrics: A survey on an emerging challenge. IEEE Transactions on Technology and Society, 1(2), 89-103, 2020.

[Fah21] Fahse, T., Huber, V., & van Giffen, B. Managing bias in machine learning projects. In Innovation Through Information Systems: Volume II: A Collection of Latest Research on Technology Issues (pp. 94-109). Springer International Publishing, 2021.

[Fra21] Franco, D., Oneto, L., Navarin, N., & Anguita, D. Learn and Visually Explain Deep Fair Models: an Application to Face Recognition. In 2021 International Joint Conference on Neural Networks (IJCNN) (pp. 1-10). IEEE, 2021.

[Gar20] Garg, P., Villasenor, J., & Foggo, V. Fairness metrics: A comparative analysis. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 3662-3666). IEEE, 2020.

[Hal22] Hall, M., van der Maaten, L., Gustafson, L., & Adcock, A. A systematic study of bias amplification. arXiv preprint arXiv:2201.11706, 2022.

[Hor22] Hort, M., Chen, Z., Zhang, J. M., Sarro, F., & Harman, M. Bias mitigation for machine learning classifiers: A comprehensive survey. arXiv preprint arXiv:2207.07068, 2022.

[Ios18] Iosifidis, V., & Ntoutsi, E. Dealing with bias via data augmentation in supervised learning scenarios. Jo Bates Paul D. Clough Robert Jäschke, 24, 11, 2018.

[Jin20] Jin, X., Barbieri, F., Davani, A. M., Kennedy, B., Neves, L., & Ren, X. Efficiently mitigating classification bias via transfer learning. arXiv preprint arXiv:2010.12864, 2020.

[Kam12] Kamiran, F., & Calders, T. Data preprocessing techniques for classification without discrimination. Knowledge and information systems, 33(1), 1-33, 2012.

[Kar19] Kärkkäinen, K., & Joo, J. Fairface: Face attribute dataset for balanced race, gender, and age. arXiv preprint arXiv:1908.04913, 2019.

[Keras] Keras, C. F. Theano-based deep learning libraryCode: https://github.com/fchollet. Documentation: http://keras.io, 2015.

[Kin09] King, D. E. Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research, 10, 1755-1758, 2009.

[Kiz21] Kizhner, I., Terras, M., Rumyantsev, M., Khokhlova, V., Demeshkova, E., Rudov, I., & Afanasieva, J. Digital cultural colonialism: measuring bias in aggregated digitized content held in Google Arts and Culture. Digital Scholarship in the Humanities, 36(3), 607-640, 2021.

[Kod19] Kodiyan, A. A. An overview of ethical issues in using AI systems in hiring with a case study of Amazon' s AI based hiring tool. Researchgate Preprint, 1-19, 2019.

[Kot22] Kotti, S., Vatsa, M., & Singh, R. On Biased Behavior of GANs for Face Verification. arXiv

preprint arXiv:2208.13061, 2022.

[Koz21] Kozodoi, N., & Varga, T. V. fairness: Algorithmic Fairness Metrics. URL https://CRAN. R-project. org/package= fairness. R package version, 1(1), 228, 2021.

[Lee22] Lee, J., Lee, J., Jung, S., & Choo, J. DebiasBench: Benchmark for Fair Comparison of Debiasing in Image Classification. arXiv preprint arXiv:2206.03680, 2022.

[McL15] McLaughlin, N., Del Rincon, J. M., & Miller, P. Data-augmentation for reducing dataset bias in person re-identification. In 2015 12th IEEE International conference on advanced video and signal based surveillance (AVSS) (pp. 1-6). IEEE, 2015.

[McL19] McLean, G., & Osei-Frimpong, K. Hey Alexa... examine the variables influencing the use of artificial intelligent in-home voice assistants. Computers in Human Behavior, 99, 28-37, 2019.

[Meh21] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. A survey on bias and fairness in machine learning. ACM Computing Surveys (CSUR), 54(6), 1-35, 2021.

[Mij18] Mikołajczyk, A., & Grochowski, M. Data augmentation for improving deep learning in image classification problem. In 2018 international interdisciplinary PhD workshop (IIPhDW) (pp. 117-122). IEEE, 2018.

[Pan10] Pan, S. J., & Yang, Q. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359, 2010.

[Par18] Park, J. H., Shin, J., & Fung, P. Reducing gender bias in abusive language detection. arXiv preprint arXiv:1808.07231, 2018.

[Sto22] Stone, R. S., Ravikumar, N., Bulpitt, A. J., & Hogg, D. C. Epistemic Uncertainty-Weighted Loss for Visual Bias Mitigation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2898-2905), 2022.

[Sze15] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9), 2015.

[Tan19] Tan, M., & Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR, 2019.

[vanG22] van Giffen, B., Herhausen, D., & Fahse, T. Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods. Journal of Business Research, 144, 93-106, 2022.

[Ver18] Verma, S., & Rubin, J. Fairness definitions explained. In Proceedings of the international workshop on software fairness (pp. 1-7), 2018.

[VKM] "The National Museums of World Culture": https://www.varldskulturmuseerna.se/en/.

[Wan19] Wang, T., Zhao, J., Yatskar, M., Chang, K. W., & Ordonez, V. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5310-5319), 2019.

[Wan20] Wang, Z., Qinami, K., Karakozis, I. C., Genova, K., Nair, P., Hata, K., & Russakovsky, O. Towards fairness in visual recognition: Effective strategies for bias mitigation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8919-8928), 2020.

[Zha18] Zhang, B. H., Lemoine, B., & Mitchell, M. Mitigating unwanted biases with adversarial learning. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society (pp. 335-340), 2018.

ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.31, No-1-2, 2023

A Resource Allocation Algorithm for a History-Aware Frame Graph

Roman Sandu
Phystech School of Applied Mathematics and Computer Science
Moscow Institute of Physics and Technology
Institutskiy Pereulok, 9 Dolgoprudny, Moscow Oblast, 141701, Russia
sandu.ra@phystech.edu

Alexandr Shcherbakov
Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Moscow, 119991, Russia
alex.shcherbakov@graphics.cs.msu.ru

ABSTRACT

We consider the problem of memory consumption by a real-time GPU-accelerated graphical application. A history of a resource is defined for a particular frame to be the final contents of such a resource at the end of the previous frame. When organizing a graphical application using a frame rendering graph approach, it makes sense to implement automatic serving of resource history read requests of nodes. In absence of history resource requests, allocating resources for a fixed frame graph is the classic problem of *dynamic storage allocation* (DSA). In this paper, we formulate a generalization of DSA that enables memory reuse for resources with history requests and provide a practical approximate algorithm for solving it.

Keywords

frame graph, dynamic storage allocation, resource aliasing, gpu memory reuse, dx12, vulkan

1 INTRODUCTION

Memory Reuse

A computation graph based approach, which has been presented in previous works [ODo17; Wih19], can be considered the current state-of-the-art in real-time render engine design. A typical implementation receives a user-defined directed acyclic graph $(\mathbf{V},\mathbf{E})$, a set of resources $\mathbf{R}$, and a resource usage function $\mathbf{U} : \mathbf{V} \to 2^{\mathbf{R}}$, such that $\mathbf{R} = \bigcup_{v\in\mathbf{V}} \mathbf{U}(v)$. We refer to the tuple $(\mathbf{V},\mathbf{E},\mathbf{R},\mathbf{U})$ as a *frame rendering graph*, or simply a *frame graph*. Vertices in $\mathbf{V}$ are referred to as nodes and represent tasks that dispatch GPU work. Elements of $\mathbf{R}$ represent *transient* resources, temporary per-frame data like the g-buffer, intermediate images during a blur, etc. A node $v \in \mathbf{V}$ may only use GPU resources from the set $\mathbf{U}(v)$ during the dispatched work. The condition that $\mathbf{R}$ is equal to the union across the image of $\mathbf{U}$ ensures that every resource is used at least once. The implementation then executes the nodes every application frame, respecting the dependency order $\mathbf{E}$, and allocates and providing the requested resources to nodes. One of the resources in $\mathbf{R}$ is considered to be the final frame image, which is presented on the screen after the frame graph finishes executing. In some cases, several resources are considered to be the final result of a frame, such as when CPU read-back is needed or when rendering in stereo for a VR application.

This approach has been shown to offer significant advantages. Leaving aside numerous architectural benefits, we focus on memory reuse. Prior to the availability of modern low-level graphics APIs such as Vulkan Direct3D 12, applications usually had to resort to resource pooling when memory consumption of transient resources became problematic. However, a pooling approach has limited memory reuse capabilities due to the abundance of incompatible resource types in graphics programming. In the simplest approach, even two textures with different resolutions cannot be substituted during execution, even though their lifetimes may be disjoint. While certain engineering tricks may be employed to enable greater resource reuse, they are often error-prone, difficult to implement and may incur a runtime performance penalty. Another memory reuse strategy is to create and destroying GPU resources on-

demand, although this approach has also proven to be inefficient due to various driver and allocator induced overheads. In presence of a modern graphics API and a frame graph, however, an application runtime is able to take a more nuanced approach to memory reuse.

Related Works

Let $\{v_i\}_{i=0}^{n} = \mathbf{V}$ be the node execution order chosen by the runtime and $\{\rho_i\} = \mathbf{R}$ an arbitrary indexing of resources. For a resource ρ_i, define $l_i = \min_{\rho_i \in \mathbf{U}(v_j)} j$, $r_i = 1 + \max_{\rho_i \in \mathbf{U}(v_j)} j$ the *lifetime* of this resource, and s_i, its size in bytes. With a modern graphics API, we can allocate memory heaps and place resources in them at certain offsets, which are to be chosen such that no two resources overlap in time and memory simultaneously. In other words, memory reuse within such a system reduces to the classic problem of *dynamic storage allocation* [GJ90, p. 226] (DSA), which in general is stated as follows. For an arbitrary set of allocations (l_i, r_i, s_i), find an allocation function $\alpha(i)$ that assigns an offset in memory to every resource, with the minimal value of $makespan = \max_i \alpha(i) + s_i$, subject to the following restriction: for any $i \neq j$, either $[l_i, r_i) \cap [l_j, r_j) = \emptyset$, or $[\alpha(i), \alpha(i) + s_i) \cap [\alpha(j), \alpha(j) + s_j) = \emptyset$. Geometrically speaking, given a set of axis-aligned rectangles on a plane, minimize the used vertical sapace by only moving the rectangles along the vertical axis, such that no two rectangles intersect. See figure 1 for an example of gathering lifetime information and building an allocation schedule for a frame graph.

The DSA problem has been relatively well studied over the years [Wil+95; Kie88; Kie91; Ger99; Buc+03]. In the special case of all resources having unit size, the problem trivially reduces to interval graph coloring and is solved in polynomial time by a greedy online algorithm. However, even the case of 2 different sizes is NP-hard[1], so approximate algorithms must be used. Most research focuses on a special case of this problem, called online DSA, where an algorithm must make a decision on $\alpha(i)$ only based on resources $1..i$. This special case formalizes the well-known notion of an allocator inside a language runtime, but it has been proven that there is a lower bound on the efficiency of such an algorithm. Define the load at time t for an instance of DSA to be $L(t) = \sum_{t \in [l_i, r_i)} s_i$, the total load as $LOAD = \max_t L(t)$, OPT to be the optimal *makespan* for that instance, and s_{max}, s_{min} to be the largest and smallest resource sizes respectively. A well-known result [Rob71] of Robson shows that $OPT/LOAD \geqslant 4/13 \cdot (2 + \log_2 s_{max}/s_{min})$. In the context of computer graphics, the s_{max}/s_{min} ratio has been observed by the authors to routinely reach values above 32, which makes the bound be $OPT/LOAD \geqslant 2$. Furthermore, even in the unit-sized allocations case, $OPT/LOAD$ is bounded below by 3 [CŚ88], suggesting that the general case lower bound of Robson may be improved. These results are also known as the fragmentation problem. However, when considering a render graph based system, we are not, in fact, limited to online algorithms. The study of offline DSA started off with reductions to interval graph coloring [CŚ88], the simplest of which has been proven to have a *performance ratio*, the upper bound on $makespan/LOAD$, of 80 [Kie88]. After a more advanced reduction scheme [Kie91] was used to achieve a ratio of 6, two consequent results by Gergov decreased the best known ratio to 5 [Ger96] and then 3 [Ger99]. A 2003 paper [Buc+03] then presents an algorithm with a performance ratio of $2 + \varepsilon$, which is the best currently known result. Moreover, that paper presents a polynomial time approximation scheme (an algorithm with a ratio of $1 + \varepsilon$) for the special case of s_{max} being bounded. This result is of particular interest to render graph systems, as resource sizes usually do not exceed a bound induced by the user's display resolution.

Our Contribution

Although for a simple frame rendering graph runtime, the memory reuse problem reduces to DSA, while implementing such a system for Gaijin's Dagor Engine, we have identified a need for a generalization. Many modern computer graphics algorithms use the notion of a *resource history*, the data of a particular resource as it was at the end of a previous frame. Such algorithms, among others, include various screen-space effects like ambient occlusion [Jim+16] and reflections [Sta15], temporal anti-aliasing [YLS20], and occlusion culling techniques [Jim+16]. In fact, among the resources currently tracked through the frame graph in Dagor Engine, almost half require the history to be read at some point while computing a frame. It is clear to see that the previously described mathematical model does not permit tracking such resources inside a frame graph and reusing their memory while the resource is not needed by any node.

Our contribution is as follows. First, we generalize the DSA problem to cyclic time, repeating the same resource usage schedule across two frames. Second, we present a best-fit greedy algorithm that solves this generalization in $O(n \log n)$ time with good practical performance ratio. Finally, we show that no algorithm of a certain natural class, similar to the classic interval graph coloring algorithm, can guarantee an optimal solution for the case of unit-sized resources.

2 RESOURCE HISTORY REQUESTS

Problem statement

We extend the $(\mathbf{V}, \mathbf{E}, \mathbf{R}, \mathbf{U})$ frame graph formalization introduced earlier with the addition of resource history

[1] Although unpublished, this result is due to Stockmeyer, 1976, according to [Buc+03].

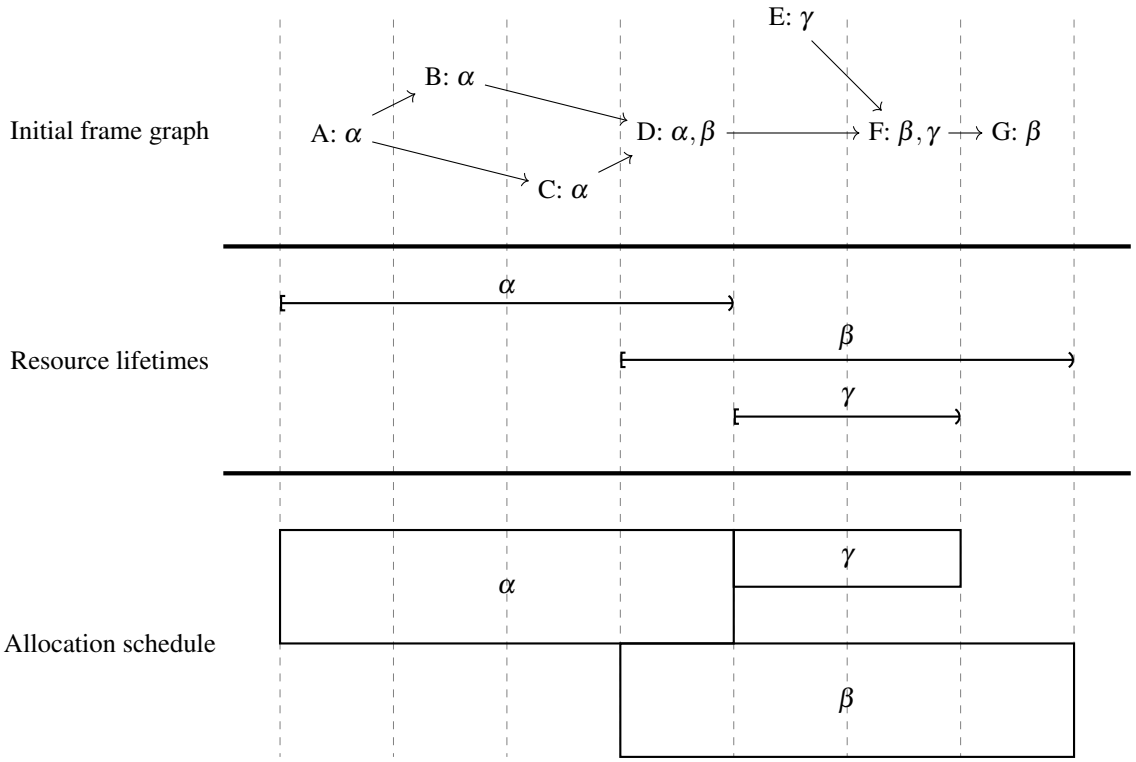


Figure 1: A visualization of a sample frame graph compilation process. Nodes from A to G list the used resources after a colon. Nodes are executed from left to right, and the horizontal axis represents time common to all 3 subfigures. Vertical guidelines represent moments between node executions, when resources start or end their lifetimes. The middle subfigure shows segments that denote lifetimes of resources α, β and γ. Finally, on the bottom, an example allocation schedule for these resources is shown, where the vertical axis represents memory locations. This example can have the following interpretation. α is the g-buffer of an application, β is the final picture, while γ is a low resolution temporary image for particles. Nodes A through D clear, draw things to and resolve the g-buffer into γ respectively, node E renders particles into γ, node F blends γ into the final picture, and G applies tone-mapping to β.

usage function $\mathbf{H} : \mathbf{V} \to 2^{\mathbf{R}}$. With this addition, we will need to start differentiating between the *logical* resources in $\mathbf{R}$ and underlying physical GPU resources. Nodes that read resource history usually use it to produce the same logical resource for the current frame. As such, we create two physical resources for every logical resource and alternate between them on even and odd frames. Note that the sizes of the two physical resources are the same and are determined by the logical resource. We represent a lifetime of a physical resource as an arbitrary pair of elements of $\mathbb{Z}_{2n}$, the cyclic group of order $2n$, and build a resource (de)allocation schedule over two consecutive frames (recall that n is the number of nodes in the graph). Somewhat abusing the notation, we denote such pseudo-intervals in $\mathbb{Z}_{2n}$ as $[l,r)$, and for each frame graph resource $\rho_i \in \mathbf{R}$ the two corresponding physical resources are $[l_i^e, r_i^e)$ and $[l_i^o, r_i^o)$. Geometrically, we now need to place the $2|\mathbf{R}|$ axis-aligned squares on an infinite cylinder $\mathbb{Z}_{2n} \times \mathbb{Z}_{\geqslant 0}$ with no intersections, such that the taken vertical (along the infinite axis) space is minimal. Although the previously defined values l_i and r_i are not applicable to our generalization, for brevity, we define the even and odd frame lifetimes in terms of them and a new value $r'_i = 1 + \max_{\rho_i \in \mathbf{H}(v_j)} j$:

$$
\begin{aligned}
l_i^e &= l_i, \\
l_i^o &= n + l_i, \\
r_i^e &= \begin{cases} n + r'_i, & r'_i \text{ well-defined} \\ r_i, & \text{otherwise} \end{cases} \\
r_i^o &= \begin{cases} r'_i, & r'_i \text{ well-defined} \\ n + r_i, & \text{otherwise} \end{cases}
\end{aligned}
$$

Note that r'_i is well-defined iff there is at least one history request for i in $\mathbf{H}$. See figure 2 for a visualization of these lifetimes. Finally, we equate any such lifetime pseudo-interval with the set of elements of $\mathbb{Z}_{2n}$ they contain: for $[l,r)$, if $l < r$, the set is $\{l, ..., r-1\}$, and $\{0, ..., r-1\} \cup \{l, ..., 2n-1\}$ otherwise. Note that for a pair $[x,x)$, the associated set is the entirety of $\mathbb{Z}_{2n}$. All regular set operations apply.

We now are ready to state our generalization of DSA, *cyclic dynamic storage allocation* (CDSA). Given a

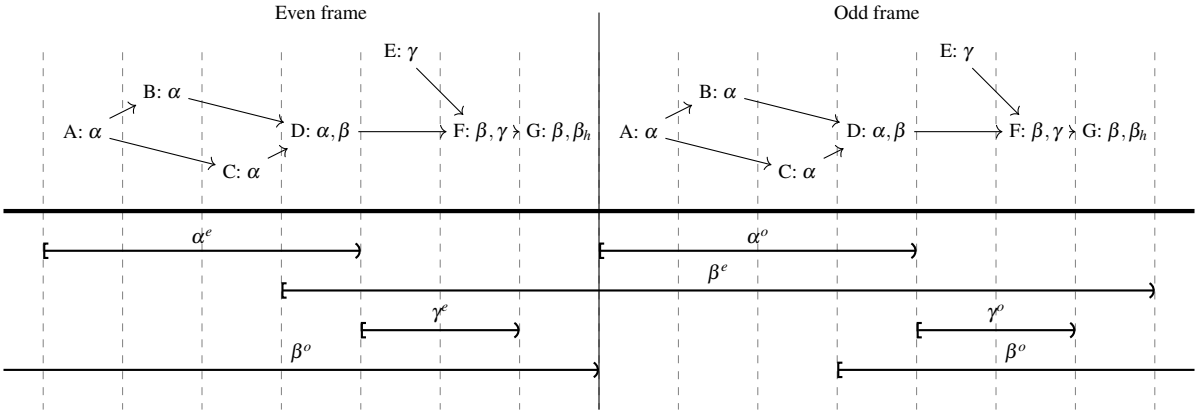


Figure 2: Visualization of physical resource lifetimes for two consequent frames, notation analogous to figure 1. Here, β_h represents a logical resource history read request of a node, while greek letters superscripted by e and o represent physical resources produced from corresponding logical resources on even and odd frames, respectively.

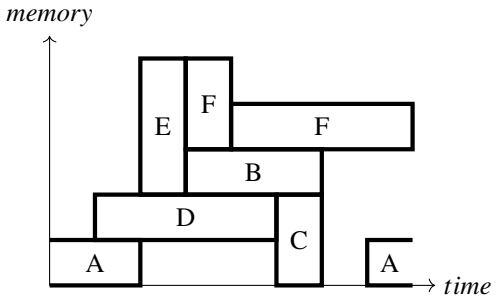


Figure 3: An optimal packing for an instance of CDSA. Here, $OPT = 5$, while $LOAD = 4$. Note that resource A has length 3 and wraps around to zero at the end of the timeline.

timeline size $T \in \mathbb{N}$, a set of arbitrary pairs of elements of $\mathbb{Z}_T$, called allocations, denoted as $\{[l_i, r_i)\}_{i=1}^{m}$ and each equipped with an integral size $s_i > 0$, find an allocation function $\alpha : \{1, ..., m\} \to \mathbb{N}_{\geqslant 0}$ that minimizes the value $makespan = \max_{1 \leqslant i \leqslant m} \alpha(i) + s_i$, such that for every pair of allocations $i \neq j$ either $[l_i, r_i) \cap [l_j, r_j) = \emptyset$ (where the intersection is interpreted as explained above), or $[\alpha(i), \alpha(i) + s_i) \cap [\alpha(j), \alpha(j) + s_j) = \emptyset$. See figure 3 for an example of a solved CDSA instance.

Before proceeding to our analysis of this problem, let us recall that for DSA, the load at time point t is defined as $L(t) = \sum_{t \in [l_i, r_i)} s_i$, and the total load as $LOAD = \max_{t \in \mathbb{Z}_T} L(t)$. These notions trivially extend to CDSA. Note that for an instance of DSA with unit allocation sizes, henceforth called UDSA for brevity (UCDSA for unit CDSA respectively), the first-fit greedy algorithm proves that $OPT = LOAD$.

Counterexamples

After stating CDSA, a question that arises naturally is whether $OPT = LOAD$ also holds for unit CDSA, and whether first-fit can be extended to find this optimal solution. To try and in some sense answer the second question, consider an arbitrary greedy algorithm of the following form (see algorithm 1).

Algorithm 1 General form of a greedy scanline algorithm for solving UCDSA. Here, "arbitrary" means determined by a concrete algorithm.

```
X ← the set of allocations
t ← 0
Choose α for all allocations alive at 0 sequentially
      and remove them from X
repeat
    i ← element of X with l_i − t  mod T smallest
    Choose α(i) arbitrarily such that no intersections
          with previous allocations occur
    Remove i from X
    t ← l_i
until no elements remain in X
```

The algorithm starts its scanline at time point 0. We argue that this is a reasonable assumption, as any other strategy of picking the starting point can be defeated by adding more resources to a counter-example. All resources alive at the starting time point are allocated sequentially, giving them the smallest α value that does not cause intersections. The algorithm then proceeds to scan the allocations in order of their lifetime start points with respect to the overall starting position and choose α for them using some arbitrary strategy, until no elements remain. Furthermore, we require that this strategy depends only on the lifetime of the current element, as well as the *back profile* and current *front profile*, defined as follows. Consider the start of the algorithm, where it choses the allocation function for all resources alive at 0. For every such resource with $\alpha(i) = s$, the back profile is defined as $p_b(s) = r_i$. For all other values of s, the back profile is defined to be 0. The initial front profile is similarly defined to be $p_f^0(s) = l_i$. After each iteration of the algorithm, a new front profile is defined in terms of the previous one as follows. If the allocation

decision made on iteration j is $\alpha(i) = s$, then $p_f^j(s) = r_i$. For all other values of s, leave $p_f^j(s) = p_f^{j-1}(s)$.

Note that this generalized algorithm is applicable to UDSA. In fact, if we restrict it to only UDSA, first-fit becomes an instance of such an algorithm and gives the optimal answer. Now consider two instances of UCDSA shown in figure 4. It is clear that both the

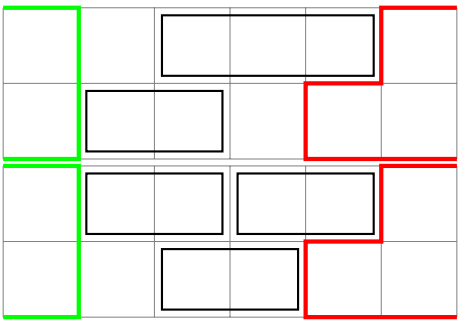


Figure 4: Counter-example to optimality of the generalized greedy UCDSA algorithm. Front profile shown in green, back profile in red, pending allocations in black.

front and back profiles, as well as the chosen resource are equal on the first iteration of the algorithm for the two instances shown. Therefore, the algorithm must choose the same offset for both instances. Obviously, picking the offset not depicted in the figure for either of the instances cannot yield an optimal solution at the end of the algorithm. Moreover, an analogous situation can easily occur in less contrived instances of UCDSA. Therefore, this demonstrates that the naive greedy approach does not in fact solve even UCDSA. Consequently, none of the existing works that reduce DSA to UDSA can be generalized to CDSA without loss of effectiveness.

We now go on to show that in fact even UCDSA is a harder problem than UDSA by proving that $OPT \geqslant 3/2 \cdot LOAD$ for UCDSA. In fact, a stronger assertion can be made: there are infinitely many instances of UCDSA for which the inequality holds. This lower bound obviously holds for general CDSA as well. The proof consists of a single picture, see figure 5. By stack-

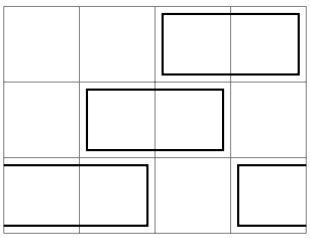


Figure 5: Instance of UCDSA where $OPT = 3$ but $LOAD = 2$

ing the instance in the figure vertically n times, we get an instance of size $3n$ where $LOAD = 2n$ but $OPT = 3n$. The result is evident.

Practical algorithm

We next present a greedy algorithm for CDSA that although has unbounded error in the general case, shows more than acceptable performance in practice. This is because CDSA instances produced by a frame graph often have a very particular structure: most resources are transient textures with sizes that are integral fractions of the user's monitor resolution. The algorithm (2) follows the general greedy scanline form from the counterexamples section, but supports non-unit weights by tracking free allocated blocks. The block to use for an incoming allocation is chosen according to the best-fit strategy. If no such block exists, a new one is allocated at the current highest used offset. When a resource's

Algorithm 2 Proposed best-fit greedy scanline algorithm for solving CDSA

$X \leftarrow$ the input set of allocations
$Y \leftarrow \emptyset$ – set of alive allocations
$A \leftarrow \emptyset$ – set of free
    blocks $(offset, size, until)$
$H \leftarrow 0$ – current heap size
$t_0 \leftarrow$ time point with smallest $L(t)$
$t \leftarrow t_0$
for allocation i alive at t **do**
    $\alpha(i) \leftarrow H$
    $H \leftarrow H + s_i$
    Move i from X into Y
end for
repeat
    $i \leftarrow$ element of X with $l_i - t \mod T$ smallest
    Move i from X into Y
    for $j \in Y$ **do**
        if j is no longer alive **then**
            Remove j from Y
            $until \leftarrow r_j$ if j is alive at t_0, ∞ otherwise
            Add $(\alpha(j), s_j, until)$ to A
            Defragment A
        end if
    end for
    if picking from A will fail on next step **then**
        Add a block (H, s_i, ∞) to A
        Defragment A
        $H \leftarrow H + s_i$
    end if
    $a \leftarrow$ block with smallest $size \geqslant s_i$ in A such
        that $[l_i, r_i) \cap [until, t_0) = \emptyset$ or $until = \infty$
    Remove a from A
    $\alpha(i) \leftarrow a.offset$
    if $a.size > s_i$ **then**
        Add $(a.offset + s_i, a.size - s_i, a.until)$ to A
        Defragment A
    end if
    $t \leftarrow l_i$
until no elements remain in X

lifetime ends, we return the block used for it to the free list A. Every time a block gets added to A, we defragment the list by merging blocks adjacent in memory into a single block. The non-trivial idea here is to additionally store an "available until" marker on each free block, that represents the back profile. When picking a free block from the list for an incoming allocation, we ensure that the allocation will not intersect with any of the allocations that were alive at t_0 by rejecting blocks that become unavailable during the current resource's lifetime. It is also important that when defragmenting the free list A, we never merge blocks with unequal $until$ markers. For this to not inhibit all defragmentation, only resources that were alive at t_0 actually store their r_i in $until$. In all other cases the sentinel value ∞ is used, that tells the algorithm that there is no "available until" limit for a block. Note that the value t_0 cannot be used as the sentinel, as it conflicts with the case of a $[t_0, t_0)$ always-alive resource that is created at time point t_0.

Note that this algorithm obviously does not have a bounded performance ratio even for DSA, as a simple sequence of $[i, i+2)$ allocations of size 2^i, $i = 0..n$, will lead to extreme fragmentation and *makespan* for the result will be $2^{n+1} - 1$, while load is $3 \cdot 2^{n-1}$. However, despite its theoretical limitations, we have observed that the algorithm provides solutions of more than acceptable quality in practice.

3 EXPERIMENTAL RESULTS

We implemented our algorithm in Gaijin's Dagor Engine and ported a significant part of Enlisted's rendering code into a frame graph system. As of May 2023, the frame graph for Enlisted on ultra presets consists of 81 nodes and tracks 27 resources, 14 of which have their history read by at least one node. As one can see from figure 6, a lot of resources alias with resources that cross the frame boundary. This aliasing saves about 10% of memory, or 6 MB, when compared with allocating frame boundary crossing resources separately and never reusing their memory. It must however be noted that a lot of GPU resources are not managed by the render graph system in Enlisted yet, so we expect to see an improvement in these numbers, as suggested by our synthetic tests that follow. The implementation runs in $O(n \log n)$ time, or about 20 µs on Enlisted's frame graph, which potentially enables mid-game changes to the structure of the frame graph.

As our data set for measuring characteristics of the proposed algorithm is extremely limited, consisting of several quality presets in a couple of games, we borrow the bootstrapping technique from statistics. Varying the game and quality presets, we gathered discrete distributions for the following data: resource lifetime length $|[l_i, r_i)|$, resource size s_i, and timeline length T. We assume that lifetime length and resource size are independent random variables, and that a resource is equally likely to begin its lifetime at any time moment. Synthetic tests for N resources were generated by resampling these discrete distribution and choosing l_i uniformly. We then ran the algorithm on these synthetic tests in two configurations, 2000 times each: with and without prohibiting aliasing for resources alive at time point t_0. This simulates a naive treatment of history requests, i.e. allocating resources with such requests separately and never reusing their memory. With this naive treatment, our algorithm becomes a variation of an online greedy allocator, commonly employed by render graph implementations. Our results are presented in figure 7. As can be seen from the plots, the algorithm shows a competitive performance ratio of around 1.1 on average, which is significantly better than what handcrafted counter-examples might suggest. A clear trend can be seen in the plots with history reuse, see figure 8. For inputs distributed similarly to real data produced by a graphics application, the algorithm shows a consistent improvement in memory reuse with increasing resource count, both on average and in the 90th percentile. The same does not hold true for tests with naive treatment of history-requested resources: the ratio instead increases with resource count.

4 CONCLUSION

We've presented a novel memory saving strategy for real-time graphics applications based on a render graph architecture. By treating resources that must outlive a frame boundary due to history read requests uniformly with all other frame graph resources and generalizing the classic DSA problem, we are able to decrease the competitive performance ratio by a significant margin and save about 10% of memory on average. Even though generally speaking CDSA is a harder problem than DSA, as shown by our counter-examples, a greedy approach can still yield good results in practice. We suspect that the presented algorithm has bounded error when restricting it to inputs with bounded allocation size, but are yet to prove this. It must however be noted that in no way can the presented algorithm be considered optimal, even for the use-case of computer graphics. Even small reductions in VRAM usage can have a significant impact on the performance of large-scale high-performance applications, or applications running on memory-constrained devices, such as smartphones, and the greedy nature of this algorithm suggests that further research should be able to find better algorithms to solve CDSA. Of especial interest is generalizing the DSA algorithm from [Buc+03], as it seems to have good performance characteristics on data sets with bounded resource size and highly clustered distribution of resource sizes.

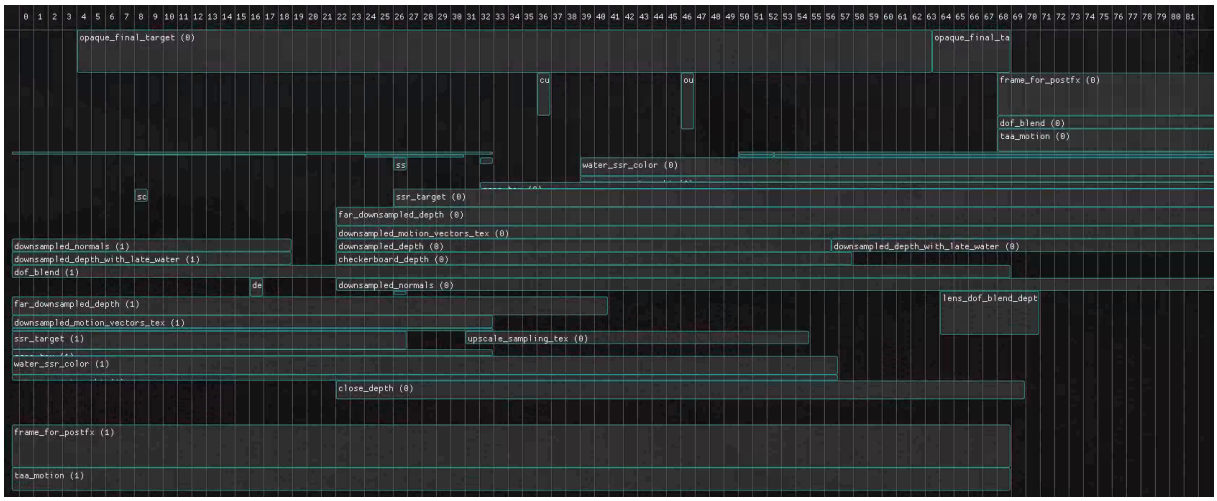


Figure 6: Resource allocation schedule in Enlisted for an even frame on ultra graphics. Horizontal axis is time, vertical axis is memory. Resources with history reads span outside of the 0-81 node range.

Further questions of interest include theoretical properties of the CDSA problem. General algorithms with bounded performance ratio and algorithms for special cases, especially polynomial approximation schemes, are yet to be found. Furthermore, even for the uniform CDSA case, it is not clear whether our lower bound of $OPT/LOAD \geqslant 3/2$ is optimal, and whether an efficient optimal algorithm exists.

5 REFERENCES

[Buc+03] Adam L. Buchsbaum et al. “OPT versus LOAD in dynamic storage allocation”. In: *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. 2003, pp. 556–564.

[CŚ88] Marek Chrobak and Maciej Ślusarek. “On some packing problem related to dynamic storage allocation”. In: *RAIRO - Theoretical Informatics and Applications* 22.4 (1988), pp. 487–499.

[Ger96] Jordan Gergov. “Approximation algorithms for dynamic storage allocation”. In: *European Symposium on Algorithms*. Springer, 1996, pp. 52–61.

[Ger99] Jordan Gergov. “Algorithms for compile-time memory optimization”. In: *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*. 1999, pp. 907–908.

[GJ90] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. USA: W. H. Freeman & Co., 1990.

[Jim+16] Jorge Jiménez et al. “Practical real-time strategies for accurate indirect occlusion”. In: *SIGGRAPH 2016 Courses: Physically Based Shading in Theory and Practice* (2016).

[Kie88] H. A. Kierstead. “The Linearity of First-Fit Coloring of Interval Graphs”. In: *SIAM Journal on Discrete Mathematics* 1.4 (Nov. 1988), pp. 526–530.

[Kie91] Hal A. Kierstead. “A polynomial time approximation algorithm for dynamic storage allocation”. In: *Discrete Mathematics* 88.2 (1991). Publisher: Elsevier, pp. 231–237.

[ODo17] Yuriy O’Donnell. “FrameGraph: Extensible Rendering Architecture in Frostbite”. Game Developers Conference. 2017.

[Rob71] John Michael Robson. “An estimate of the store size necessary for dynamic storage allocation”. In: *Journal of the ACM (JACM)* 18.3 (1971), pp. 416–423.

[Sta15] Tomasz Stachowiak. “Stochastic Screen-Space Reflections”. SIGGRAPH. 2015.

[Wih19] Graham Wihlidal. “Halcyon: Rapid innovation using modern graphics”. Reboot Develop. 2019.

[Wil+95] Paul R Wilson et al. “Dynamic storage allocation: A survey and critical review”. In: *Memory Management: International Workshop IWMM 95 Kinross, UK, September 27–29, 1995 Proceedings*. Springer. 1995, pp. 1–116.

[YLS20] Lei Yang, Shiqiu Liu, and Marco Salvi. “A survey of temporal antialiasing techniques”. In: *Computer graphics forum*. Vol. 39. 2. Wiley Online Library. 2020, pp. 607–621.

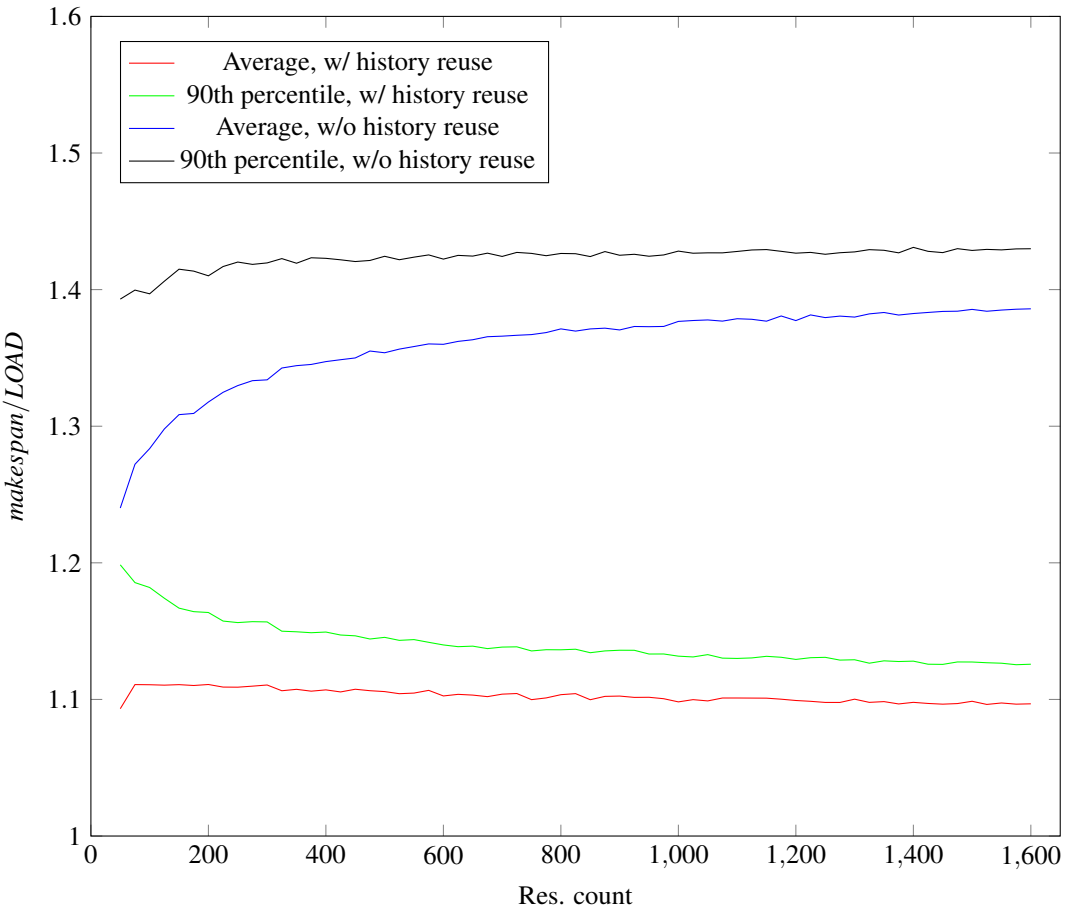


Figure 7: Performance measurements for our best-fit greedy scanline CDSA algorithm. Cumulative over 2000 runs on synthetic tests, bootstrapped from production data.

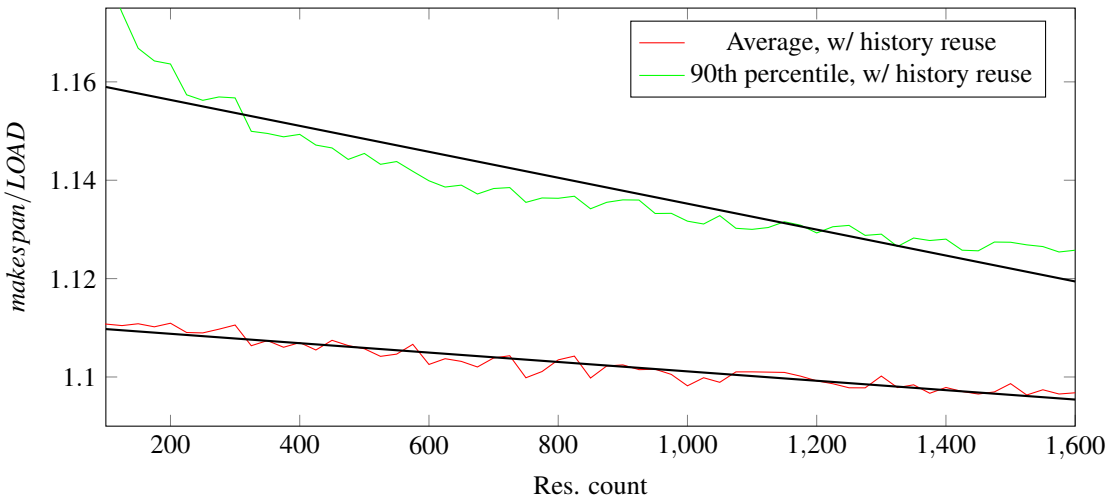


Figure 8: Enlarged plots with history reuse from figure 7. Clear downward trend can be observed.

ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.31, No-1-2, 2023

Real-time Light Estimation and Neural Soft Shadows for AR Indoor Scenarios

Alexander Sommer[1]
alexander.sommer@hs-rm.de

Ulrich Schwanecke[1]
ulrich.schwanecke@hs-rm.de

Elmar Schoemer[2]
schoemer@uni-mainz.de

[1] Computer Vision and Mixed Reality Group, RheinMain University of Applied Sciences Wiesbaden Rüsselsheim, Germany
[2]Institute of Computer Science, Johannes Gutenberg University Mainz, Germany

ABSTRACT

We present a pipeline for realistic embedding of virtual objects into footage of indoor scenes with focus on real-time AR applications. Our pipeline consists of two main components: A light estimator and a neural soft shadow texture generator. Our light estimation is based on deep neural nets and determines the main light direction, light color, ambient color and an opacity parameter for the shadow texture. Our *neural soft shadow* method encodes object-based realistic soft shadows as light direction dependent textures in a small MLP. We show that our pipeline can be used to integrate objects into AR scenes in a new level of realism in real-time. Our models are small enough to run on current mobile devices. We achieve runtimes of 9ms for light estimation and 5ms for neural shadows on an iPhone 11 Pro.

Keywords: augmented reality, light estimation, shadow rendering, neural soft shadows

1 INTRODUCTION

We propose a method for realistically inserting virtual objects into indoor scenes in the context of augmented reality applications. Thereby we first estimate the current lighting situation in the scene from a single RGB image captured by the camera of, for example, a mobile device. Then we use this information to insert the virtual object into the existing scene as plausibly and realistically as possible.

The light situation in an existing scene can be captured by placing a light probe at the position of the image. This can create a 360° high-dynamic range (HDR) panorama, also called environment map, of the scene. Such an HDR image contains a large amount of information about bright and dark areas that would be clipped as black or white in an ordinary 8Bit low-dynamic range (LDR) image. Since the map contains information about the illumination of each direction of the scene at a given point, this environment map can be utilized to illuminate an object as if it were in the scene using methods from image-based lighting (IBL) [8]. Some techniques have been developed to estimate this environment map from a single limited field-of-view LDR image without additional 360° cameras using neural nets and deep learning [12, 28, 27]. However, such an environment map is only valid for a single specific point in the scene. Moreover IBL techniques can be used to realistically illuminate objects with spatial varying light. Shadows in IBL are created by tracing the path of light and its interaction with other objects in the scene. While this produces a very realistic shadow cast, a large number of path traces is required. This is computationally intensive and therefore not suitable for real-time applications on mobile devices.

Figure 1: Example application of our pipeline: The light estimation determines the light direction and ambient color for rendering the inserted object. Based on the determined light direction, additional neural soft shadows are generated to create a realistic shadow cast as texture.

Alternatively, parametric models exist that describe the light sources as physical objects in a 3D scene. In contrast to an environment map from IBL, these models are valid for the entire scene. In the simplest case, such a parametric model can be for example a directional light with a fixed direction. Parametric light sources have a long history in computer graphics and there are several methods to efficiently calculate the shadow cast by objects. However, they often have to be modeled manually by a 3D artist for an existing scene. Recently, methods came up to estimate parametric lights directly from an input image using neural networks [5, 13, 11]. Our work takes up on these methods. We restrict ourselves to reliably predict the main light direction at a given point from a limited field-of-view LDR indoor RGB camera image and additionally determine the light color as well as the ambient color.

Especially in indoor scenes, the lighting situation is very complex. For the realistic overlay of virtual objects in the context of AR [26], it makes a big difference whether a realistic or visual convincing shadow cast is present. Many other light estimation works map the existing lighting situation, but are only able to realistically insert virtual objects through offline rendering, e.g. ray tracing. Most shadows in indoor scenes are soft shadows as they are caused by light objects in a relatively short distance with a certain surface area. They are much more complex to compute than hard shadows caused by a quasi infinitely distant light source like the sun in outdoor scenes. We present a method to use the estimated light direction from the previous part to generate realistic indoor soft shadows in real-time. For this purpose we present a novel approach to encode precomputed ray-traced soft shadows using a neural network. This small network can be queried in real-time to generate a shadow texture depending on the light direction (see Fig. 1).
Our *main contributions* are as follows:

1. An improved deep neural network for parametric light direction estimation in indoor scenes.

2. A new method for encoding shadow textures in an MLP that is memory friendly and fast to query.

3. A complete pipeline for light estimation and shadow creation for real-time AR applications on mobile devices.

2 RELATED WORK

Existing work related to ours can be roughly divided into two categories. On the one hand, research in the area of estimating the existing light situation in the real scenery and, on the other hand, research on how to use this information for the realistic insertion of virtual objects into the augmented reality.

Light estimation is a classical problem in the field of computer vision or computer graphics as a subarea of 3D scene reconstruction. An accurate determination of the existing lighting conditions is crucial for a convincing insertion of virtual objects into the real environment.

Classic approaches usually require multiple images and/or more detailed knowledge about the underlying scene geometry. For example, Debevec and Malik showed how the omnidirectional HDR radiance map can be reconstructed using multiple shots of a reflecting sphere with different exposure settings [9] and how to render synthetic objects into real scenes [8]. Lombardi and Nishino [19] showed how illumination can be reconstructed from a single image of an object with known geometry. Balcı and Güdükbay [2] reconstructed illumation based on the shadows in scenes that were mainly illuminated by the sun. Baron and Malik [3] reconstructed not only the illumination but also the geometry and reflectivity of an unknown object from an image using shape priors. Lopez-Moreno et al. [20] presented an approach based on heuristics that does not require geometric knowledge.

With the rise of machine learning based approaches the need for information about the scenery could be further reduced. There exist quite some work that estimate lighting information and environment maps. For example, Hold-Geoffroy et al. [14] used a deep neural net to predict the illumination in outdoor scenes from a single image by relying on a physically-based sky model. Gardner et al. [12] estimated an HDR illumination map for indoor scenes also from a single image by splitting the process into light position estimation and HDR intensity estimation. Song and Funkhouser [28] used a multi-stage approach to predict a 360° LDR map from a single image and completed geometry and intensity on HDR scale. Somanath and Kurz [27] predicted a true HDR map from a single camera image in a single stage approach tailored to mobile augmented reality (AR) real-time applications. Other approaches focus more on estimating light in form of low dimensional parameters. Garon et al. [13] used spherical harmonic coefficients as light model. Cheng et al. [5] also used spherical harmonics for their light model, but used the images from the front and rear camera for the estimation.

Gardner et al. [11] described a deep neural net that estimates light parameters for individual light objects. This method is the closest to our work. They used the Laval Indoor HDR dataset [12] which contains about 2100 HDR maps to train the network. These parameters for the training data were determined by fitting ellipses on the HDR intensity maps. The brightest area in the map was detected and the ellipse then was fitted by region growing. This area was masked and the process was repeated to determine a number of light sources.

The parameters of the light source were defined by the size of the ellipse, average HDR intensity in the ellipse area and average HDR color value. Furthermore, a predicted depth map was used to determine the distance to the light source. We also use the Laval Indoor HDR dataset and with a DenseNet pretrained on ImageNet a similar network architecture. However, unlike Gardner et al. we estimate a light direction and therefore do not need to rely on predicted depth maps for the dataset.

Shadow calculation is a very broad and relatively old field of research in computer graphics. It ranges from simple methods for computing hard shadows, such as projection shadows [4], shadow mapping [31] and shadow volumes [7] to more advanced methods for computing soft shadows like image-based soft shadows [1], geometry-based soft shadows [22] and volumetric shadows [18].

In contrast to previous work, we present a new approach in which we encode pre-computed shadow textures for an object in the weights of a neural network. This has the advantage that realistic soft shadows can be displayed in real-time on mobile devices, since the network can be queried very quickly. The idea of encoding images or textures in neural networks is not new. Stanley [29] encoded image information in Compositional Pattern Producing Network (CPPN) inspired by encoding in natural DNA. Rainer et al. [25, 24] used neural networks to compress the bidirectional texture function (BTF). Mildenhall et al. [21] trained an multilayer perceptron (MLP) to generate novel views from unknown perspectives of complex scenes. They used a mapping for the input coordinates to create a higher dimensional input space that allowed more high frequency variations in their output. This strategy was inspired by the positional encoding in the Transformer architecture [30] and is also used by our method.

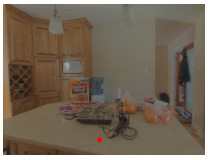
3 LIGHT ESTIMATION

To estimate the existing light situation in a scene using a single RGB image, we characterize the light situation by a set of parameters

$$(\boldsymbol{d}, \boldsymbol{c}, \boldsymbol{a}, o)\,. \tag{1}$$

Here $\boldsymbol{d} \in \mathbb{R}^3$ is a unit vector which determines the light direction, $\boldsymbol{c} \in \mathbb{R}^3$ is the light color defined by RGB values with normalized components in $[0,1]$ and $\boldsymbol{a}$ is an RGB vector corresponding to the ambient lighting of the scene. The parameter o is a scalar value and measure for the opacity of the shadow texture described in Section 4. A value $o = 1$ corresponds to an alpha value of 100% and a value of $o = 0$ corresponds to an alpha value of 0%, i.e. invisible shadows. We train a convolutional neural network (CNN) to predict these parameters from a single RGB image with a resolution of 256x192 pixels.

(a) Original panorama



(b) Cropped image

(c) Warped panorama

Figure 2: For a given panorama (a), the image information from inside the red frame is used to create the rectified cropped image (b). A warped panorama (c) is projected around the insertion point (red point in (b)).

3.1 Input Data

For training the network, a large number of images is needed for which the exact light situation of the whole scene is known. 360° HDR panoramas are particularly suitable for this, since one can crop a limited field-of-view image from them to obtain input images (see Fig. 2a & Fig. 2b), while still being able to recover the entire lighting situation of the scene. We use the Laval Indoor HDR dataset [12] which contains about 2100 HDR panoramas, taken at different indoor scenes.

Like Gardner et. al [12], we extract 8 different limited field-of-view crops per panorama at random polar angles θ between 60° and 120° and azimuth angles ϕ between 0° and 360°. We use a field-of-view (FOV) of 85° to approximate the viewing angle of non-wide-angle cameras in modern smartphones. We perform a rectilinear projection (see red frame in Fig. 2a)) to back-project the distortion in the panoramas. The 360° HDR panorama describes the light situation of the whole scene at the point where the camera was placed for the panorama. However, this does not correspond to the exact lighting situation around the cropped image. For finding out the exact light situation at that point, one would have to shoot a new 360° HDR panorama at the virtual camera location of the cropped image. To estimate the light situation at this location we rotate the original panorama so that the cropped area is exactly in the center and then apply the same warping operator as described in [12]. The resulting new panorama (see Fig. 2c) is an approximation of the panorama around the virtual camera location of the cropped area.

We use each of the warped panoramas to extract the light parameters for the corresponding cropped input

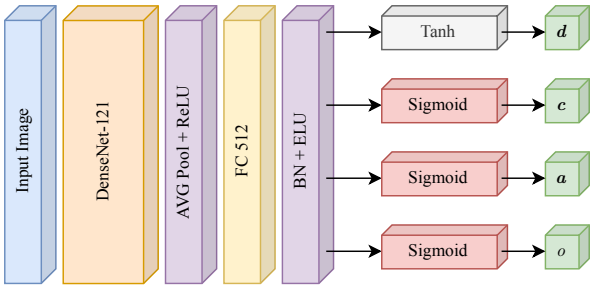


Figure 3: Proposed light estimation network architecture.

image. We first determine the pixel intensity I_{ij} by adding the individual RGB channels with weights that correspond to the natural perception of the individual colors, i.e.

$$I_{ij} = 0.0722 \cdot R_{ij} + 0.7152 \cdot G_{ij} + 0.2126 \cdot B_{ij}, \quad (2)$$

where i is the pixel's column and j its row.

Then we mask the areas where the intensity is greater than 5% of the maximum intensity $I_{\max}$ as highlights. It should be noted that this is only applicable when working with HDR data. To determine the average light direction from the highlight area, we introduce two weights. First, the light direction of each pixel is weighted by its intensity. Second, the light direction of each pixel is weighted by the area that this pixel occupies on the unit sphere:

$$\omega_{ij} = \frac{2\pi^2}{w \cdot h} \sin\left(\frac{j+0.5}{h}\pi\right) \quad (3)$$

where j is the pixel's row and w, h are the width and height of the panorama. This is necessary because, for example, an area near the poles occupies significantly more pixels on the panorama than an area with the same size at the equator. The resulting average light direction is the parameter $\boldsymbol{d}$. To determine the light color $\boldsymbol{c}$, the same weights are applied to the individual RGB values of the highlight area in a tone-mapped version of the panorama to obtain a mean highlight color. The ambient color $\boldsymbol{a}$ can be determined from the remaining pixel values of the tone-mapped panorama by using the same procedure. We determine the value for the opacity parameter o from the quotient of the summed weighted intensities for the highlight areas I_l^{tot} and analog for the remaining areas I_a^{tot}:

$$o = 1 - \tanh\left(\frac{I_a^{\text{tot}}}{0.05 \cdot I_l^{\text{tot}}}\right). \quad (4)$$

The less the intensities from the highlight areas differ from those of the ambient area, the lower the opacity of the shadow textures.

3.2 Network Architecture

As mentioned before we use a CNN to estimate the parameters from the input RGB image. Since the dataset is too small to train a network from scratch, we use a DenseNet-121 [17], pretrained on ImageNet [10] as an encoder. The block configuration is (6, 12, 24, 16) with a growth rate of 32, a compression of 0.5 and a batch norm size of 4. Furthermore, 64 initial features, ReLU activations and 2D average pooling with a pool size of 4 are used. The classifier of the DenseNet is removed, so the network produces a latent vector with size 512. This is forwarded to a fully connected (FC) 512 layer with batch norm and ELU activation. For each of the four parameters there is a separate FC layer as network head. The heads for the parameters $\boldsymbol{c}$, $\boldsymbol{a}$, o are each normalized using a sigmoid function so that they lie between $(0,1)$. For the parameter $\boldsymbol{d}$ we use a tanh activation function and normalize the entire vector to unit length. The complete architecture is visualized in Figure 3.

Metric	Gardner19(1)	Ours
RMSE	0.1114	**0.1101**
si-RMSE	0.1518	**0.1501**
RMLE	0.07007	**0.06928**
Angular Error	3.556°	**3.542°**

Table 1: Comparision by different widely used metrics of our method with the state of the art parametric indoor light estimation by Gardner et al. [11] with one light source. Best results in bold.

3.3 Training & Implementation

During training, we directly compare the estimated parameters with the ground truth parameters. Thereby individual losses for each head are calculated as mean squared error. The total loss function is the weighted sum of the individual losses, i.e.

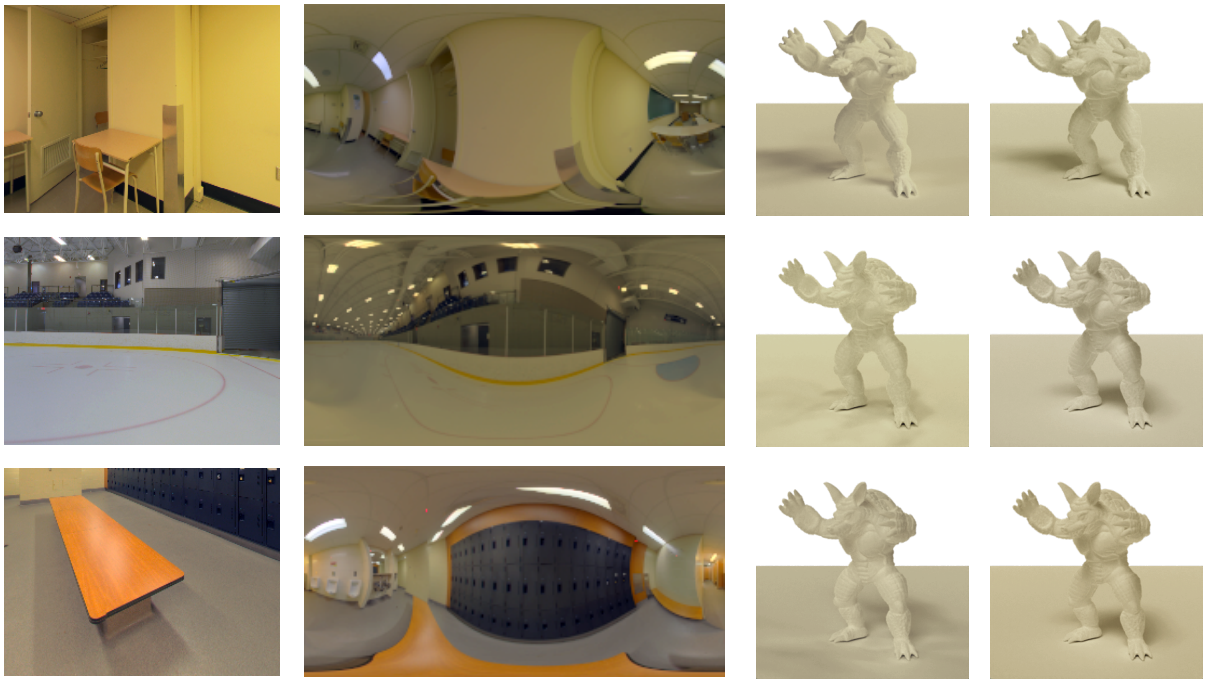
$$\begin{aligned}\mathscr{L} &= \omega_d l_2(\boldsymbol{d}^{\text{est}}, \boldsymbol{d}^{\text{gt}}) + \omega_c l_2(\boldsymbol{c}^{\text{est}}, \boldsymbol{c}^{\text{gt}}) \\ &\quad + \omega_a l_2(\boldsymbol{a}^{\text{est}}, \boldsymbol{a}^{\text{gt}}) + \omega_o l_2(o^{\text{est}}, o^{\text{gt}}).\end{aligned} \quad (5)$$

We weight the individual losses differently with the weights $\omega_d = 5$, $\omega_c = 2$, $\omega_a = 2$ and $\omega_o = 1$. Since a correct estimation of the direction is of utmost importance for us, ω_d gets the highest value.

We train the network for a total of 60 epochs using an Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate $l_r = 0.001$ is halved every 15 epochs. We use a batch size of 128 samples and a random 85/15 split of the dataset for training and validation. Scenes unknown to the network were used for testing. Typically, training takes about 2 hours on two Nvidia RTX A6000 GPUs. In total, our network consists of 7.7M parameters. The interference time on the iPhone 11 Pro GPU is 62ms and on the Apple Neural Engine (ANE) 9ms.

3.4 Evaluation

Comparing the results of two light estimation approaches is challenging. Since qualitative evaluation



(a) Input image (b) GT HDR panorama (c) GT result (d) Our result

Figure 4: Exemplified representation of our evaluation. (a) shows the input image, (b) the corresponding GT HDR panorama, (c) the GT image of the Armadillo rendered with IBL techniques, and (d) the image of the same Armadillo rendered using the light parameters from our light estimation.

always contains personal bias, we rely on a purely quantitative measure for this evaluation. We don't compare our method with approaches, that do not estimate a parametric light direction but spatially varying light coefficients like spherical harmonics [5, 13] or complete environment maps [12, 28, 27], because we especially need the light direction for the shadow calculation (Sec. 4). We therefore compare our light estimation approach only with the work of Gardner et al. [11] when using one main light source. We neglect our opacity parameter o at this point, since its use is mainly for the shadow textures presented in Section 4 and will be evaluated in the overall pipeline evaluation in Section 5.

We use a simple scene with an armadillo and a plane as a shadow catcher (see Fig. 4c). For a given input image (see Fig. 4a), we render a GT image (see Fig. 4c) with the corresponding warped GT environment map (Fig. 4b), as described in Section 3.1, with IBL techniques. We then estimate light parameters with the respective light estimation. The same scene is rendered again with a parametric light source and ambient color (see Fig. 4d).

To compare renderings of the two predictions with the GT image we use 4 different metrics. On the one hand RMSE as well as the scale-invariant si-RMSE and RMLE and on the other hand a per pixel RGB angular error [15]. The standard RMSE is a good measure for the error in the relation between ambient and light intensity. The two scale-invariant measures filter out differences in the scales of the two images and are therefore good measures for errors in light position due to difference in shadows. The RGB angular error, on the other hand, comes from whitebalance research and is a good measure to evaluate the color predicition of the light source and the ambient color.

In total, we evaluated 977 images from a test set unknown to the network. We used Blender [6] for all renderings. Table 1 shows the results of our evaluation. It can be seen that our method performs 1-1.2% better than the previous state-of-the-art method in all metrics when considering only a single light source.

4 SHADOWS

We aim to generate a planar shadow texture (see Fig. 5b), i.e. a 2D grayscale image, depending on the light direction defined by a unit vector $\boldsymbol{d} \in \mathbb{R}^3$ for a specific object (see Fig. 5a). Our experiments showed that the use of cartesian coordinates leads to a more stable training than spherical coordinates since the network seems to have problems with the discontinuity between $\phi = 2\pi$ and $\phi = 0$. This results in a *shadow function*

$$f : \mathbb{R}^5 \longrightarrow \mathbb{R}, \quad f(i, j, \boldsymbol{d}) \longrightarrow v, \tag{6}$$

(a) Object (b) Shadow texture

Figure 5: A chair lit by a front light (a) with the corresponding shadow texture (b).

Figure 6: Shadow textures of the Armadillo (see Fig. 4c) from different light directions $\boldsymbol{d}$.

that maps pixel position (i, j) together with a light direction $\mathbf{d}$ to a grayscale value v. We use a MLPs are a universal function approximator [16], to represent the desired shadow function.

4.1 Input Data

We train one specific network for each individual model. As training data, we use shadow textures for a variety of different light directions. These textures are created with a simple scene setup and the Cycles render engine in Blender [6]. The scene consists of a quadratic plane that acts as a shadow catcher. The plane is dimensioned so that its side length is three times as long as the largest side of the bounding box that contains the object to be trained. The object is centered on the plane and is assigned a material that is invisible to the render engine but allows shadow casting. An orthographic camera from the top view captures the textures. A directional light (sun in blender) with an opening angle of 20° is used as the light source. This type of light is defined by one direction and still produces soft shadows. It's therefore well suited as an approximation for indoor shadows. This light source is set to different light directions for the individual training samples. We use uniformly distributed spherical angles θ, ϕ. Where θ takes values from 0° to 45° with an increment of 4.5° and ϕ takes values from 0° to 360° with an increment of 12°. This results in a total of 301 texture samples. For each sample, we use a resolution of 256x256 pixels. Figure 6 shows an example of shadow textures for different light directions for the Armadillo (see Fig. 4c).

4.2 Network Architecture

As mentioned before (see Eq. (6)), all information about the shadows is mapped by pixel-wise functions from 5D space to 1D grayscale information. Since neural networks tend to learn a low-frequency bias, we assist them in learning high-frequency details by mapping the 5D input to a higher dimensional space, as shown by Rahaman et al. [23]. This technique is also used very successfully with NeRFs [21]. Similar to Vaswani et al. [30] with Transformers, we use an encoder function Φ to map each of the five input dimensions $x \in \mathbb{R}$ to a higher dimensional sequence of alternating sine and cosine functions:

$$\begin{aligned}\Phi(x) = \big(&\sin\big(2^0\pi x\big), \cos\big(2^0\pi x\big), \ldots, \\ &\ldots, \sin\big(2^{L-1}\pi x\big), \cos\big(2^{L-1}\pi x\big)\big)\end{aligned} \tag{7}$$

where L is a dimensionality parameter. The image space (i, j) is normalized to values in [0,1]. For the image space encoding is done with a dimensionality parameter $L = 10$. The elements of the light direction vector $\boldsymbol{d}$ by definition take only values in [-1,1] and for their encoding we choose an $L = 4$ analogous to the viewing direction vector in [21]. In total we map the $\mathbb{R}^5$ input space to higher dimensional space of $\mathbb{R}^{64}$. The input passes through h hidden layers, each with a filter size s, and is activated with ReLUs after each hidden layer (see Fig. 8). In our experiments we use a filter size s of 128 to 256 and a number of hidden layers h from 1 to 4. The output value v of the network is normalized with a sigmoid function between 0 and 1.

4.3 Training

During training, for each shadow texture sample k with fixed light direction $\boldsymbol{d}$, we take a number of N random continuous pixel locations $(i, j) \in [0, 1]$. Here, the ground truth grayscale value $v^{\text{gt}}_{i,j}$ at the continuous location (i, j) is obtained by bilinear interpolating from the known values at the discrete surrounding known pixel values. It should be mentioned that it is also possible to train the network without interpolation only on random known discrete pixel values. This speeds up the training by a factor of 5 since filtering is a bottleneck. On the other hand, it reduces the quality of the network, and the ability to predict different resolutions with the

(a) Ground truth

(b) Ours

Figure 7: An example of our qualitative evaluation. Left: Coffee table rendered with the ground truth HDR panorama around the insertion point. Right: Coffee table rendered with a directional light and ambient color from our light estimation and shadow texture from our neural soft shadows.

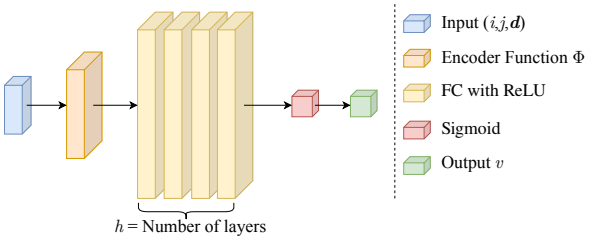


Figure 8: Proposed shadow network architecture.

network is lost. As loss function $\mathscr{L}$ we take the mean squared error loss l_2 between estimated pixel value v^{est} and interpolated pixel value v^{gt}:

$$\mathscr{L} = l_2(v^{\text{est}}, v^{\text{gt}}). \tag{8}$$

4.4 Implementation

One advantage of our method is that the resulting network is very small and thus not only requires little memory, but a forward pass also has a low interference time. The forward passes for all 65536 pixels of a 256x256 texture need in total about 33ms on the GPU of the iPhone 11 Pro and 5ms on the ANE. Assuming a filter size $s = 128$ and a number of hidden layers $h = 3$ the network has just 58k parameters. The data set with its 301 grayscale images with a resolution of 256x256 is small enough to be loaded completely into the memory even with simple consumer GPUs. We train our network for a total of 10000 epochs and need about 5 minutes (or just under a minute without bilinear filtering) on an Nvidia RTX A6000. As in Section 3.3, we again use an Adam optimizer with standard values of $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We apply an exponential learning rate decay ($\gamma = 0.99977$) to the initial learning rate $l_r = 0.001$ so that it is reduced to one-tenth of the original value after 10000 epochs. Per texture sample we use $N = 256$ pixel locations, which results in 77k network passes per epoch.

	GT	Ours
Rating	3.49 ± 0.38	3.26 ± 0.46
Votes	0.544%	0.456%

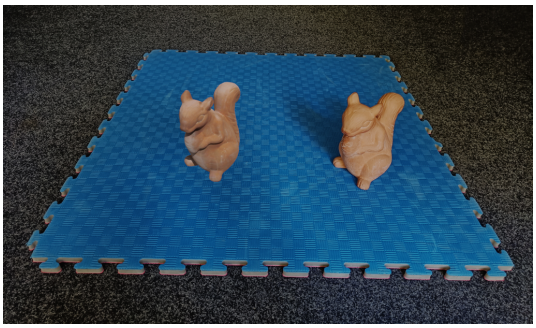
Table 2: Results of the qualitative evaluation (20 images, 50 participants). Rating describes how realistically an objects fits into the scene considering only lighting and shadows on a scale from 1 (very unrealistic) to 5 (very realistic). Votes denotes the percentage of which image was prefered in terms of realistic look (50% = perfect confusion).

4.5 Limitations

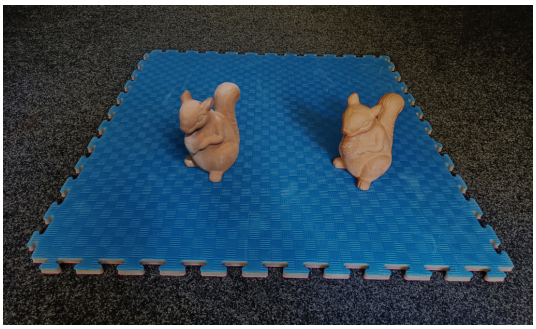
Currently, our method is only suitable for creating a planar shadow texture for the plane it sits on. This is sufficient for of AR applications, where an object is placed in the middle of an empty room and is far enough away from walls to cast a shadow on them. Problems arise when a virtual object should cast shadows on another virtual object or on non-virtual objects in the scene.

5 OVERALL PIPELINE EVALUATION

We determine the overall quality of our entire pipeline with a qualitative evaluation. For this we use new HDR panoramas that are not from the Laval Indoor HDR dataset and have not yet been seen by the network. For each panorama we choose a cropped rectified image where a virtual object should be inserted. We use the light estimation from Section 3 to determine the light direction, light color, ambient color and the opacity value for the shadows. We then use the light direction to determine the shadow texture using our method from Section 4. We insert the object into the image



(a) Without shadow cast



(b) With neural shadow texture

Figure 9: Comparison between a real clay squirrel (right) and the virtual object (left) rendered with the light parameter of the light estimation from Section 3. (a) shows the object without shadow cast and (b) with the neural shadow texture from our method in Section 4.

and render it using only a directional light and ambient lighting. We also add the neural shadow texture with the estimated opacity (see Fig. 7a). In comparison, we determine the warped panorama (see Sec. 3.1) at the insertion point and render the same object with ray traced IBL and a plane as shadow catcher (see Fig. 7b).

A total of 20 images (see supplementary material) were created for qualitave evaluation. We showed these images to 50 participants. On the one hand, the participants were asked to assess how realistically an object fits into the existing scene in terms of its lighting and shadows. For the rating, we use the Likert scale with values from 1 (very unrealistic) to 5 (very realistic). Explicitly the participants were told not to consider syle, proportions, object selection and context. On the other hand, the participants were shown both pictures (see Fig. 7) next to each other and they were asked to decide which of the two pictures they thought was more realistic looking in terms of lighting and shadows. Table 2 shows the results of our survey. It turns out that the participants as a whole give the ground truth visualizations only a slightly higher quality rating than our visualizations. This is also confirmed by the fact that quite a few participants prefer our visualization to the ground truth in a direct comparison.

Furthermore, in Figure 9 we compare a real object with a rendered virtual version. For this we place a real clay squirrel in the room and leave space for the virtual version. The photo was taken with an ordinary smartphone and the light estimation from Section 3 was used to determine the light direction, light color, ambient color and the opacity value of the shadows. The virtual squirrel was inserted on the left and rendered with the light parameters. Figure 9a shows the virtual squirrel without shadow cast. Figure 9b shows the squirrel with the neural soft shadow texture generated with our method from Section 4. It is easy to see that without shadows the object looks out of place in the scene. The subtle soft shadow of our method, on the other hand, conveys immersion.

6 CONCLUSION

We presented a complete pipeline for realistic embedding of virtual objects into indoor scenes. Our light estimation determines a parametric description of the light situation from an RGB image as input. Our neural soft shadow method generates realistic soft shadows as textures that allow to embed virtual objects in previously unknown levels of realismn in real-time into AR scenes. Of course, our method is not suitable for reproducing complex lighting situations exactly, but it is suitable for giving the viewer a convincing sense of immersion. This is supported by our user test where approximately the same number of subjects preferred our method over ground truth visualization. Our entire pipeline is real-time capable on current mobile devices.

In particular, our fundamental work in the area of neural soft shadows opens up a wide range of possibilities for future research. At the moment we are working on how to effectively transfer our method to the shadow cast on walls. In this case, the distance to the wall adds another degree of freedom to the problem. It would be interesting to incorporate more complex light sources, such as area lights, with further parameters like light size in neural shadows. It is also exciting to see if multiple light sources can be represented as neural soft shadows. Furthermore, we could imagine that complex shadows of semi-transparent objects could be another future application of our method.

ACKNOWLEDGMENTS

This project (HA project no. 1102/21-104) is financed with funds of LOEWE - Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence).

REFERENCES

[1] Maneesh Agrawala, Ravi Ramamoorthi, Alan Heirich, and Laurent Moll. Efficient image-based methods for rendering soft shadows. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 375–384, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[2] Hasan Balcı and Uğur Güdükbay. Sun position estimation and tracking for virtual object placement in time-lapse videos. *Signal, Image and Video Processing*, 11, 07 2017.

[3] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:1670–1687, 2013.

[4] James F. Blinn. Me and my (fake) shadow. *IEEE Computer Graphics and Applications*, 8:82–86, 1988.

[5] Dachuan Cheng, Jian Shi, Yanyun Chen, Xiaoming Deng, and Xiaopeng Zhang. Learning scene illumination by pairwise photos from rear and front mobile cameras. *Computer Graphics Forum*, 37:213–221, 10 2018.

[6] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2022.

[7] Franklin C. Crow. Shadow algorithms for computer graphics. In *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '77, pages 242–248, New York, NY, USA, 1977. Association for Computing Machinery.

[8] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 189–198, New York, NY, USA, 1998. Association for Computing Machinery.

[9] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 369–378, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[11] Marc-André Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagné, and Jean-François Lalonde. Deep parametric indoor lighting estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7174–7182, 10 2019.

[12] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *ACM Trans. Graph.*, 36(6), nov 2017.

[13] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-Francois Lalonde. Fast spatially-varying indoor lighting estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6901–6910, 06 2019.

[14] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2373–2382, 07 2017.

[15] S.D. Hordley and G.D. Finlayson. Re-evaluating colour constancy algorithms. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 1, pages 76–79 Vol.1, 2004.

[16] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366, jul 1989.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[18] James T. Kajiya and Brian P Von Herzen. Ray tracing volume densities. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, pages 165–174, New York, NY, USA, 1984. Association for Computing Machinery.

[19] Stephen Lombardi and Ko Nishino. Reflectance and illumination recovery in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(1):129–141, jan 2016.

[20] Jorge Lopez-Moreno, Elena Garces, Sunil Hadap, Erik Reinhard, and Diego Gutiérrez. Multiple light source estimation in a single image. *Computer Graphics Forum*, 32, 12 2013.

[21] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, dec 2021.

[22] Steven Parker, Peter Shirley, and Brian Smits. Single sample soft shadow. *Tech. Rep. UUCS-98-019*, 10 1998.

[23] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Dräxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron C. Courville. On the spectral bias of neural networks. In *ICML*, 2018.

[24] Gilles Rainer, Abhijeet Ghosh, Wenzel Jakob, and Tim Weyrich. Unified neural encoding of btfs. *Computer Graphics Forum*, 39:167–178, 05 2020.

[25] Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. Neural btf compression and interpolation. *Computer Graphics Forum*, 38:235–244, 05 2019.

[26] Kai Rohmer, Wolfgang BÃ¼schel, Raimund Dachselt, and Thorsten Grosch. Interactive near-field illumination for photorealistic augmented reality on mobile devices. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 29–38, 2014.

[27] Gowri Somanath and Daniel Kurz. Hdr environment map estimation for real-time augmented reality. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11293–11301, 06 2021.

[28] Shuran Song and Thomas Funkhouser. Neural illumination: Lighting prediction for indoor environments. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6911–6919, 06 2019.

[29] Kenneth O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131162, jun 2007.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[31] Lance Williams. Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph.*, 12(3):270–274, aug 1978.

ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.31, No-1-2, 2023

A Framework for Art-directed Augmentation of Human Motion in Videos on Mobile Devices

R. Debski, O. Schmitt, P. Trenz, M. Reimann, J. Döllner, M. Trapp
Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam, Germany
A. Semmo, S. Pasewaldt
Digital Masterpieces, Potsdam, Germany

Figure 1: Stills from videos using different motion effects applied to user-generated content in Lumo (from left to right): Chalky Silhouette, Dotted Silhouette, Light Trails, Dr. Strange, Square, Marionette, Glow Sticks, Glow Sticks with black background

ABSTRACT

This paper presents a framework and mobile video editing app for interactive artistic augmentation of human motion in videos. While creating motion effects with industry-standard software is time-intensive and requires expertise, and popular video effect apps have limited customization options, our approach enables a multitude of art-directable, highly customizable motion effects. We propose a graph-based video processing framework that uses mobile-optimized machine learning models for human segmentation and pose estimation to augment RGB video data, enabling the rendering and animation of content-adaptive graphical elements that highlight and emphasize motion. Our modular framework architecture enables effect designers to create diverse motion effects that include body pose-based effects such as glow stick or light trail effects, silhouette-based effects such as halos and outlines, and layer-based effects that provide depth perception and enable interaction with virtual objects.

Keywords: Video Stylization, Mobile Processing, Human Motion, Depicting Dynamics, Video Effects

1 INTRODUCTION

User-generated short-form videos are one of the most influential formats on social media. While platforms such as TikTok and YouTube (Shorts) offer a variety of filters and visual effects, users still like using their imagination to create their own. Typically, industry-standard software such as Adobe After Effects or Resolve Fusion is used to create video effects and stylization, but their effective usage often requires in-depth knowledge and time. To produce especially motion-based video effects without such software, users require a significant amount of time and effort, such as in the #Glowstickdance trend [Cha20], in which participants adhered glow lights to their bodies to dance in the dark.

The objective of our work is to analyze, design, and implement a mobile application framework that facilitates the development and synthesis of motion visualizations suitable for user-generated content, such as those shown in Fig. 1. Our primary focus is on enabling users to film, edit, and share content instantly using mobile devices. To allow for rapid editing, we aim to automate the effect generation process, while at the same time retaining creative control over significant portions of the effect design. Such creative control should manifest in parameterized control over effect variables such as adjustments of color, range of the glow, or background modifications. Our ultimate goal is to open up the world of effect design to new audiences. Additionally, we aim to make the implementation and variation of effects easier for developers by creating a simple and flexible framework.

Problem Statement. To achieve our objective, we implement a framework that processes multidimensional video data to produce art-directed motion visualiza-

tions. In this endeavor, we identified the following challenges that our framework should address:

Interactivity / Usability: Most professional video editing programs, such as Adobe After Effects, are rich in functionality but also have a steep learning curve. Creating motion-based effects is often accomplished through time-consuming frame-by-frame editing. To ensure our app's suitability for user-generated content, we seek to create an application enabling the creation of complex effects with excellent visual quality, even by nonprofessional users. Nonprofessional video editing of user-generated videos is already possible with popular mobile apps such as TikTok or Instagram that offer customization options that enable users to adjust the size and order of effects. However, these options are often hidden behind several menus with different interfaces, leading to a complicated workflow. To address this issue, we aim to provide a user-friendly interface that allows users to apply, customize, and delete effects quickly and easily. To achieve this, we integrate all application options into a single menu that displays all options at once thereby avoiding complicated workflows while retaining a high level of creative control. Users can apply effects in sequence, reorder them, and adjust their timing. They can preview the effects in real-time and make adjustments to effect settings while observing the results. We achieve real-time performance and rendering to ensure that the preview is accurate and responsive. As our effect control options are specifically designed for motion effects, our interface can restrict the number of available options compared to industry-standard software.

Exchangeability/Adaptability: Motion effects in existing mobile apps such as TikTok offer limited customization options for the effect designer. To address this limitation, we introduce a highly adaptable framework that provides users with a wide range of customization options. Users have control over numerous parameters, such as color scheme, background color, and stylization. This adaptability is also reflected in the framework's design and enables the developer to create new and exciting effects or combine existing ones for the user to experiment with. The framework's different components, or processing nodes, can be easily interchanged and combined to achieve various visual outputs. If a modification is necessary, the code is simple to modify for new data and usage scenarios.

Approach & Contributions. Our technical approach consists of a framework for building video processing pipelines that allows for the real-time synthesis of content-aware motion visualizations. To this end, the Red-Green-Blue (RGB) video data is extended by extracted information such as depth, segmentation, or human body poses. To achieve real-time performance, our framework enables concurrent processing and presentation of such multidimensional data. To summarize, this paper presents

1. a framework for automatically creating content-aware video effects to visualize motion. Our approach uses semantic data such as human body pose and segmentation information to create content-aware effects that focus on a subject's movement.
2. a software system that processes multi-dimensional data simultaneously on a frame-by-frame basis. The framework hides complexity by encapsulating processing functionality in easily extendable processing nodes, organized in a graph-based structure to form processing pipelines.
3. an intuitive timeline-based user interface for art directing, applying, customizing, and removing content-aware effects on multi-dimensional videos without requiring domain knowledge. Our supplemental video[1] demonstrates the interface and interaction with the app.

The remainder of this work is structured as follows. Sec. 2 reviews related work with respect to motion visualization in general and motion-based video effects in particular. Sec. 3 analyzes popular real-world motion effects and derives semiotic aspects and designs for our video effects. Sec. 4 presents a conceptual overview of the proposed approach. Sec. 6 details implementation aspects of the framework. Sec. 7 evaluates the system's performance, presents a post-deployment user study on the intuitiveness and usability of the application, and discusses results and limitations. Finally, Sec. 8 concludes this work and presents ideas for future research.

2 BACKGROUND & RELATED WORK

Motion Visualization. The visualization of motion is a challenge to artists, scientists, and image creators alike, as Cutting *et al*. [Cut02] state in their analysis of contemporary artistic motion visualization. They suggest several effective ways to visualize motion, which include dynamic balance, multiple stroboscopic images, affine shear, photographic blur, and action lines. These techniques are based on principles of physics, perception, and visual design, and have been widely used in various applications. Nienhaus *et al*. [Nie05] propose an automated depiction system for visualizing dynamics in 3D scenes. They follow design principles found in visual art, such as those used in comic books, and

[1] https://drive.google.com/file/d/1ERgXif9aQDC_Ug_fVfxAxnUokNfKp89w

introduce visual metaphors and symbolization, such as object bending or squashing to depict motion in scenes. Bouver-Zappa *et al*. [Bou07] use 3D skeletal motion capture data to generate motion cues in a still image of the captured character using arrows, noise waves, and stroboscopic motion. Kwon *et al*. [Kwo12] further utilize the skeletal structure of the motion capture for improved motion cues. Schmid *et al*. [Sch10] introduce a method for visualizing motion by extending surface shaders to programmable motion effects with knowledge of the global spatio-temporal trajectory. Their approach enables the creation of complex motion effects, such as speed lines and blurring, that can be customized to different styles.

Motion-based Video Effects. In addition to their use in 3D scenes, motion visualization techniques, such as speed and focus lines, motion blur, ghosts, motion lines, and halos, are also utilized as photo and video effects. Collomosse *et al*. [Col03; Col05] present a system to render such motion cues using a variety of traditional feature-tracking-based computer vision techniques, whereas Nienhaus *et al*. [Nie08] use background subtraction and focus on the forward lean affine shear as a motion cue. Umeda *et al*. [Ume12] employ such techniques for real-time manga-like depiction in videos and the Vivid [Sem19] mobile app applies similar visual metaphors for depicting dynamics in live-photos. Lu *et al*. [Lu13] convert video sequences into movement depictions of annotated body parts using arrows and motion particles. However, these approaches are either one-shot non-interactive methods and thus not suitable to perform art-direction [Ume12; Col05; Lu13] or are only suitable for creating stylized images [Col05; Nie08; Lu13; Sem19]. Mayer *et al*. [May21] propose a mobile application for artistic silhouette visualization in videos that allows for interactive effect editing in a timeline-based graphical interface. We adopt a similar interface design for our Lumo mobile app. However, we enable a much wider range of content-adaptive effects and customization options, that allows for the interactive visualization of scenes, silhouettes, and human body poses. Our proposed processing pipeline design furthermore allows for easy extension and customization of new effects.

3 EFFECT ANALYSIS AND DESIGN

Analysis of Real-world Examples. To characterize motion-based video effects, we first examined actual instances of artwork that use additional motion graphics features in their artwork such as those shown in Fig. 2. We collected a large sample of videos and photos as reference, identified common semiotic aspects, and analyzed their composition into the video effects. In the analyzed video effects, common themes include playing with light, video annotations, and addition of artificial elements into the scene. All of these add new or

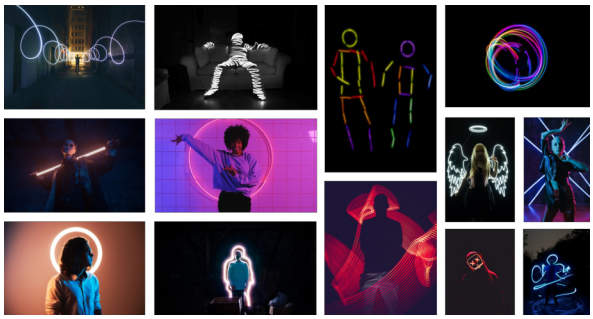


Figure 2: Overview of selected artworks that use motion-based video effects.

enhance already existing strong visual focus points to the image or video and often make it visually more appealing and striking.

Semiotic Aspects. Based on the analysis results, 11 semiotic aspects distinguishing motion-based elementary primitives and mechanisms were identified. They are visually summarized in Fig. 3 and described in the following.

Motion Effect-Type: The effect may be of a silhouette, skeleton or action line/light-trail, or generic annotation type.

Graphical Elements: Outlined shapes (e.g., triangles, squares, and circles) are formed by (poly) lines and curves, which can be represented by Bézier curves. Lines can also be dashed or dotted. Lines and shapes can appear once or in bigger numbers.

Stroke Weight: Lines may have different widths but are mostly of medium weight.

Base Color: Lines have saturated colors and hues mainly in reds, turquoise, blue, purple, greens, and yellows, which dominate the color palettes. Lines may exhibit varying degrees of transparency.

Texture: Lines are solid or can have patterns, such as a sketchy line.

Neon Glow: Lines may have the appearance of neon lights. This look is achieved by a brighter core in the line's center and a transparency gradient fall-off.

Background Brightness: The depicted scene may appear fully illuminated, dusky, or deeply dark.

Position: Annotations and lines can be offset from a position such as a point on or outline of a person.

Animation: Effect elements and their attributes can be animated meaning they change position, shape, color, etc. over time in a controlled manner.

Attachment: Effect elements such as shapes or lines appear to be affixed to anchor points on individuals in the scene, with their position dynamically animated to track the motion of the subjects. As a result, the elements remain temporally coherent,

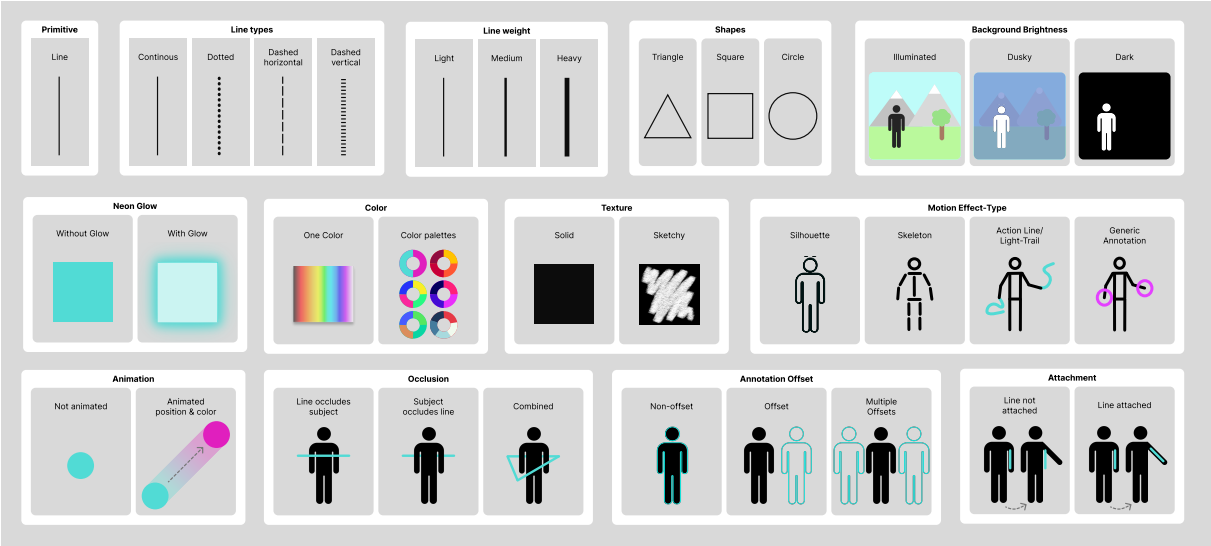


Figure 3: Summarizing presentation of semiotic aspects.

maintaining consistent attributes such as position, shape, and color, and do not change abruptly or unpredictably over time.

Occlusion: Effect elements can be occluded by subjects and objects in the depicted scene. The element thus appears to be behind the subjects or objects resulting in the illusion of depth.

Exemplary Effect Design. Combining the different semiotic aspects an exemplary effect design can be achieved as follows. Neon-colored lines are positioned on the limbs of a person in the video, anchored to the respective joints. Each line features a neon glow and medium stroke weight with a solid texture and opaque opacity. The lines are not occluded by the person, move coherently over time, and are not animated. The scene appears to be set at night. This results in an effect denoted as "Glowstick Effect", shown in Fig. 4, that is similar to glow sticks attached to the limbs of a person (e.g., [Cha20]).

Figure 4: "Glowstick Effect" Design.

4 FRAMEWORK

Based on the previous motion-based effect analysis and design, this section describes preliminaries and requirements (Sec. 4.1), the fundamental graph-based processing framework (Sec. 4.2), and the conceptual modeling of motion-based effects (Sec. 4.3).

4.1 Preliminaries & Assumptions

This section summarizes the functional and non-functional requirements for the framework's implementation.

Functional Requirements. We identified the following functional requirements for our framework: The application should be able to capture, process, and export multi-dimensional videos, i.e., RGB video streams with additional information channels. Those multi-dimensional videos should be loaded and stored persistently. Furthermore, work snapshots should be savable during effect creation and should be subsequently restorable and editable. The framework should enable the creation of silhouette and human pose-based video visualizations. To provide a user-friendly interface to the user, the application should present a Graphical User Interface (GUI) that allows to create, edit, and delete effects. Furthermore, the application should provide a real-time preview while editing.

Non-functional Requirements. We also identified the following non-functional requirements: The framework should be reliable in creating, modifying, deleting, and consistently reproducing effects. The framework must prioritize high usability, with an intuitive interface that requires no professional knowledge of video editing. From the developer's perspective, the framework must be maintainable, easily adaptable to different data and use cases, and efficiently extendable with new visualizations and additional parameters. Finally, the effects generated by the framework should be visually appealing, with a fluid and artistic appearance that enhances the overall visual characteristics of the video.

4.2 Graph-based Video Processing

To efficiently process our captured video frames we use a pipeline-based approach, i.e., a directed graph of pro-

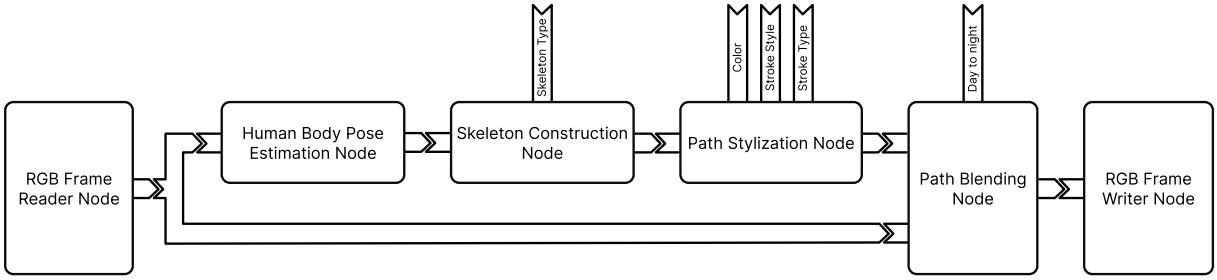


Figure 5: An exemplary pipeline that constructs the "Glowstick Effect".

cessing components similar to Lugaresi *et al*. [Lug19]. The processing graph allows a high degree of flexibility and easy implementation of different effects.

Node Port Reactivity. To allow interactivity, the inputs and outputs of the nodes are reactive, i.e., when a value at the input changes, the node processes it and passes the result forward via the output port. This results in a cascading change in the pipeline, leading to sequential or concurrent processing of nodes in the graph. To minimize processing requirements, a node only processes when an input value changes. A node can have more than one input and output port to increase flexibility. The processing of such multi-input nodes can be configured by the effect developer using a triggering rule. Common rules include the triggering based on either the change in any input port, a specific input port, a group of input ports, or all input ports since the last time the node underwent processing.

Graph Nodes for Specialized Processing Tasks. To illustrate the flexibility of the graph-based processing concept, we provide an exemplary processing pipeline that creates a stylized Bézier path from an RGB frame. To this end, we connect the processing nodes as shown in Fig. 5, and briefly describe them in the following:

Human Body Pose Estimation Node: It receives an RGB frame as input and transforms it into estimated human joint point positions in the frame, which are returned via the output port.

Skeleton Construction Node: It receives estimated joint point positions at its input port, filters out the relevant points, and subsequently creates a Bézier path, which is passed on via the output port.

Path Stylization Node: It receives a Bézier path at the input port, renders it, and stylizes it using parameters such as color, stroke type, or stroke style, which are provided at the other input ports, and returns the rendered and styled path. As the nodes are reactive, changing the color input of the line styling node will immediately change the line color in the resulting image after a cascade of processing.

Blending node: It receives two RGB frames as input, and returns the blended result. One of the inputs can be the output of the path stylization node and the other input can be the initial RGB frame.

To fetch input data and write output results of the processing graph to video file, read and write nodes are employed. The reader node reads multi-dimensional video from the disk frame by frame and provides the individual data as typed frame data at its output port. As nodes are reactive, the read results in a processing pass. Thus, the reader node can determine the frequency of processing passes and thus affects the video frame rate. To achieve the desired video frame rate, a scheduler is added to the read node. The write node ends a pipeline and writes the processed image to disk or allows displaying it in real-time.

Asynchronous processing. By offloading image data processing from the Central Processing Unit (CPU) to specialized hardware, such as Graphics Processing Units (GPUs) or Neural Processing Units (NPUs), and using the asynchronous programming paradigm, CPU resources can be freed up. While waiting for the GPU or NPU to finish processing and return results, the CPU can continue processing. The reactive node ports ensure a sequential flow of data, as calculation results are passed to subsequent nodes as soon as data is available. This asynchronous processing approach allows for flexibility in effect development and performance optimization. We could further improve performance by implementing a caching mechanism for output ports that allows results to be retrieved without re-processing the node if there have been no changes to its input port.

Precomputation Task. Processing-intensive steps, such as segmentation mask estimation in a video frame, can result in non-interactive processing performance if a user's hardware resources are insufficient. To solve this problem, precomputed human segmentations can be stored in the multi-dimensional video file before effect processing. The multi-dimensional video frame reader node can then read these frames along with the RGB frame and a segmentation estimation node is no longer required. We note that modern Apple mobile devices already run at interactive frame rates without such precomputations.

4.3 Modeling Motion-based Video Effects

A video effect is temporally divided into *Intro*, *Main*, and *Outro* stages (Fig. 6). Keyframes K_0 to K_3 are used

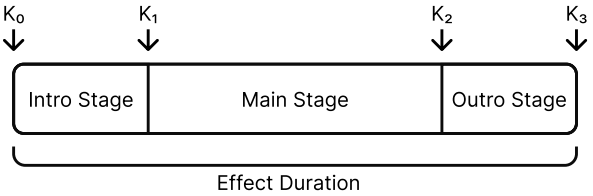


Figure 6: Keyframe-based parameterization of a motion-based video effect.

to define the start and end of an effect and the transitions between effect stages. The total duration of an effect is the time difference between K_0 and K_3. Animations of video effects can be created by interpolating visual variables between keyframes. The interpolation functions can be chosen from standard presets or defined using parameterized curves.

The additional video data generated from the Computer Vision (CV) framework, e.g., joint points, and segmentation mask can be used in several different ways to create a motion-based video effect, of which four will be explained in the following.

Connecting: We utilize joint point data and connect the points using Bézier curves, resulting in a skeleton-like effect. By adding an offset between the lines and post-processing the Bézier curves with a glow effect, an effect similar to attaching glow-sticks to the persons can be achieved.

Buffering: To create time-dependent effects, we buffer data over multiple frames, such as the data from the left and right wrists. We then draw a line through all of the joint points detected from previous frames. This results in a visual effect that resembles a light painting, with the line capturing the motion of the hand over time.

Constructing: We utilize detected human body poses as anchor points for further effect placement. For example, we use the detected root and neck points of a person to estimate the head placement, creating a convincing stick figure effect without the need for knowledge about the actual head rotation and placement. This is necessary because head rotation and placement might be partially obscured, making it difficult to place and size an ellipse around the head accurately.

Animating: We calculate metrics such as the distance between two points in the view space using joint points. The distance can then be used as an offset to change the position of Bézier curves already existing for the effect in the processed image. This results in an animation effect driven by the movement of the person in the video and thus motion visualization.

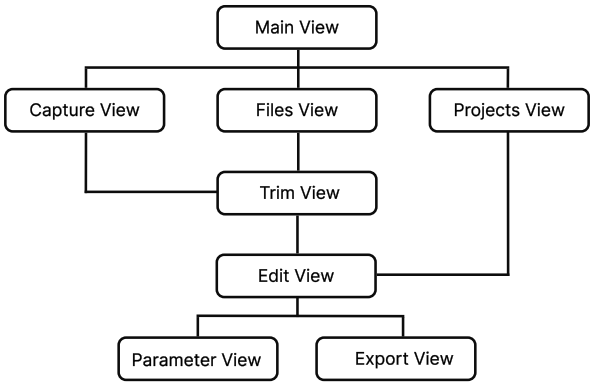


Figure 7: Diagram of the view hierarchy for individual views of our mobile application.

5 GRAPHICAL USER INTERFACE

To provide access to different functionality, the mobile app is structured into several views. Fig. 7 shows the view hierarchy of the mobile application, which we detail in the following.

Entry Point: The Main View, shown in Fig. 8, represents the entry point of the app, which provides buttons to navigate to the subordinate views. The Files View allows the user to load an existing video from the camera roll for editing. Saved projects can be opened again in the Projects View, accessed by the load projects button. Furthermore, the user can record a new video using the device's camera in the Capture View.

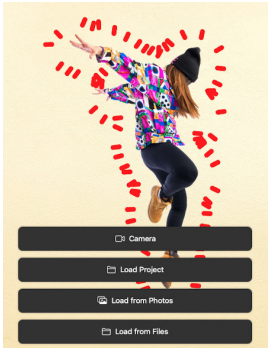
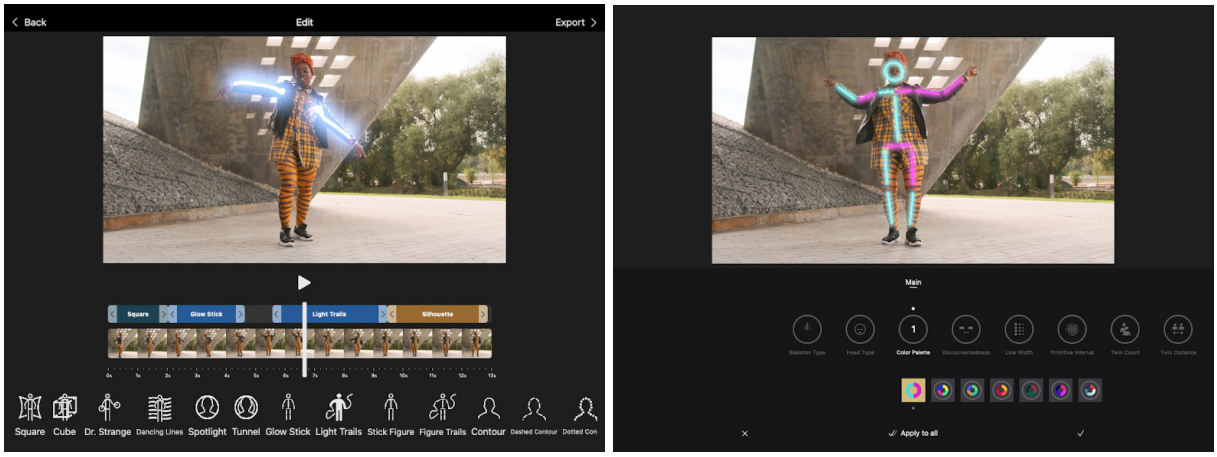


Figure 8: Main View

Trim View: Provides a video preview interface and allows to shorten a captured or loaded video.

Edit View: Provides a timeline-based video preview interface and allows to place video effects on the video (Fig. 9(a)). The top half of the screen displays the video preview and applied effects in real time allowing for interactive tweaking of effects. On the bottom of the screen the user can find various buttons to apply effects, represented by an icon and effect name. The user can find effects easily by horizontal scrolling of the effect button bar. Pressing and holding an effect button applies the effect to the video starting at the current video position for the duration of the button press while live previewing the video effect. The applied effect appears as a colored bar in the video effect timeline, found above the effect button row. To aid mobile usability there is a minimum effect length, that ensures effect bars are still easily visible and clickable in the timeline. Since only one effect can be active at a given time, an effect can only be applied if there is enough



(a) Edit View (b) Parameter View

Figure 9: Screenshots of the user interface. The Edit View allows the user to apply, shorten, expand or delete effects. The Parameter View enables changing the effect parameters.

space in the video timeline. Applied effects can be prolonged or shortened by adjusting the end handles of the effect bars in the timeline. Clicking or tapping on an effect bar opens a modal allowing the user to delete the effect from the timeline or to switch to the parameter view. Pressing the export button in the top right screen results in a high quality rendering of the video with the applied effects for exporting and saving to the device.

Parameter View: Allows the user to customize effect appearance by adjusting effect parameters (Fig. 9(b)) and preview the results in real time. Every parameter is symbolized by an icon and a label with the parameter's name, where adjusted parameters are highlighted in a different color while the default setting is marked with a point below the value. Depending on the parameter type, different user interface elements for parameter adjustments, such as integer, float and color sliders, switches, icon buttons or effect icon buttons, are used for interaction. The parameter changes can be discarded, applied to the current effect, or applied to all effects of the same type using the three buttons on the bottom of the screen.

Export View: Allows the user to export the resulting video and save or share it with other users.

6 IMPLEMENTATION ASPECTS

Our mobile processing framework requires sensor and processing hardware to capture and process multidimensional videos, such as provided by the iPad Pro 3rd generation, which we used as our main validation platform. The mobile app is developed in the Swift programming language (version 5) and makes use of several Apple software libraries and frameworks. Our system is encapsulated in several components, see the supplementary material for a component diagram, they are briefly described as follows:

User Interface component. The GUI component implements the graphical representation and is implemented using the SwiftUI framework and a reactive programming paradigm in the form of the Apple Combine Framework. The AVKit library is utilized to enable and control media playback.

Pipeline component. The pipeline implements the multidimensional video processing architecture. It represents the core component of the framework, enabling the video frames to step through all executing steps and use their multidimensional video data to create different effects. The pipeline consists of multiple components to implement nodes and effects. The Apple Vision, CoreImage, and AVFoundation libraries are used in the processing nodes to implement computer vision techniques, Input/Output (IO)-related operations, and image manipulation. Furthermore, the Combine framework is employed to implement the data flow between processing nodes, which empowers the reactivity of interfaces between nodes and the implementation of asynchronous processing routines.

IO & Processing Components. The IO component offers data access to the device's media collection, local memory, and sensors, enables the recording and storing of multidimensional data and enables the user to permanently save processed content as a compressed project. The processing component includes custom-developed processing routines to create varied motion-based effects and uses Apple's MetalKit library for GPU-accelerated processing.

7 RESULTS & DISCUSSION

This section qualitatively evaluates our approach by means of different application examples (Sec. 7.1) and a performed User Study (Sec. 7.2) as well as quantitatively regarding runtime performance (Sec. 7.3). Furthermore, it discusses limitations (Sec. 7.4)

(a) Silhouette-based Effects (b) Body Pose-based Effects (c) Combined Effects

Figure 10: Stills from exemplary videos generated using the presented mobile application.

7.1 Application Examples

Fig. 10 shows stills from selected application examples generated using Lumo.

Silhouette-based Effects. These effects emphasize the contour of an individual's silhouette (Fig. 10(a)). They are produced by identifying the silhouette of an individual in the frame as a binary bit-mask. The effects may be portrayed as a continuous, dotted, dashed, or animated line. Additionally, attributes such as the color or glow of the line can be adjusted.

Body Pose-based Effects. To create body pose-based effects, we detect the joint positions of an individual as screen coordinates and connect them in a specific sequence with lines (Fig. 10(b)). Different sequences can produce various interpretations, such as a "Glow-stick effect", where the person's body joints are disconnected, or a "stick-figure effect", where all body lines are connected. Furthermore, altering properties such as line color and background brightness can modify the appearance of the effects. Additionally, different sequences for connection enable us to create differing body types. We are also able to create effects by buffering joint information over several frames to create effects like the light trails effect.

Combined Effects. We also created combined effects, that simulate a person being placed in a 3D scene, by fusing silhouette and joint point detection. First, a person's silhouette is detected as a bit-mask and used to crop the individual from the scene. Layering methods are then applied to create the illusion that the subject is situated within a three-dimensional setting ((Fig. 10(c), right). Adding the joint detection allows us to animate the scene by positioning scene objects depending on body coordinates and movements. This allows for the creation of various effects, such as "laser eyes" or "Dr. Strange"-like effects (Fig. 10(c), left).

7.2 Usability Evaluation

One of our non-functional requirements for Lumo is usability. We aim to ensure that the software is both error-free and user-friendly, even for non-professional users. To achieve this, we conducted a user study focused on determining the following: (1) whether the users found the GUI intuitive, (2) whether the effect appliance was perceived as fast and easy and (3) whether users were satisfied with the level of creative control they had over the effect design.

Participants & Apparatus. For our study, we recruited 18 volunteers (9 female and 9 male) between the ages of 17 and 55 years to test our Lumo application for the first time. The participants had no or little prior experience with video editing and stylization, and only four of them reported editing videos professionally or as a leisure activity. All participants were conceptually familiar with video filters and effects. We conducted a supervised in-person study where every participant used an iPad Pro (11", 3rd generation with M1 GPU and 8 GB Random Access Memory (RAM), running iPad OS 15.6.1) and was given a document providing a list of tasks. Each study session followed the same structure: (*i*) a quick overview of the study's methodology was provided; then, (*ii*) the participant received a document containing a brief app description, as it would be typically seen in an App/Play Store; following that, (*iii*) the participant received three tasks to complete sequentially. The study sessions lasted approx. 20 min.

User Tasks. We designed three tasks to cover the main functionalities of Lumo, arranged in increasing levels of difficulty.

Task 1 (T1): The user was given a 7 s video, which he was asked to load and edit. The given task involved applying a specific effect ("Glow Stick Effect") for a certain amount of time and extending or shortening the application time afterwards. The task concluded with exporting the video. The objective of this task was to help participants understand the concept of the effects in Lumo and the app workflow. On average, the participants took 143 s (excluding export time) for this task.

Task 2 (T2): The user was given a video with one applied effect and was tasked to change its parame-

ter values. We provided two GUI elements to modify these parameters: a slider and a switch, both of which users were asked to utilize to customize the given effect. The objective of this task was to help participants understand different kinds of effect customization. On average, participants completed this task in 112 s.

Task 3 (T3): For the last challenge we asked the participants to delete the applied effect and apply at least two different new effects to the video. Participants were also required to switch between the two effects when the dancing person did an eye-catching move and adjust the parameters of both effects according to their judgment. The objective of this task was to test whether participants could reuse their learnings from the previous tasks and understand the variability of effects. On average, participants completed this task in 151 s.

Data Collection and Analysis. After the study each user was asked to fill out a questionnaire, based on the Questionnaire for User Interface Satisfaction (QUIS) [Chi88] and the Computer System Usability Questionnaire (CSUQ) [Lew95] without time constraints.

Task 1 was successfully completed by all participants. They were satisfied with the visual results and expressed interest in exploring more of the app's functionalities. However, it was noticeable that nearly all of the users tried to apply the effects using a drag-and-drop metaphor instead of tap-and-hold, even after being shown a help message. In the qualitative part of the questionnaire, participants expressed a desire for a drag-and-drop functionality to be implemented.

Task 2 was completed the fastest and the option to adjust effects was found easily by the participants. They were fascinated by the control they had over the appearance of the effect. Nevertheless, some participants remarked that the terminology of the effect parameters is too technical for the non-professional target user group. Further, 33.3 % of the participants rated the application's flexibility with 5/5 points on the Likert scale, 27.8 % with 4 points, and the remaining gave 2 or 3 points. This might indicate that users wish for more options to customize effects, while also desiring greater clarity in the existing options.

Task 3 was the most enjoyable for participants, who relished the opportunity to try out different effects and parameters. This challenge took the longest on average, not because of its complexity, but due to the users' tendency to explore all the options for customizing the effects. Furthermore, participants expressed a desire for a feature that enables layering of effects, which is currently possible in the software with the use of blending but not available in the GUI.

The primary goal of the user study was to evaluate the usability of Lumo. In terms of usability, (*i*) all of the users were able to find the app's functionalities relatively fast, 72.2 % of the participants agreed that performing tasks in Lumo is straightforward, and 66.7 % rated the interface as pleasant. However, most of the users were dissatisfied with the number of help and error messages. Further, the size of the text and icons was too small for the majority of the users. Regarding learnability, (*ii*) 83.3 % of the users found it easy to learn how to use the application and 83.3 % were satisfied with the reliability of the app. Finally, (*iii*), 50 % rated the application's experience as stimulating. Most participants were able to find all the functions in the app that they expected to find (66.7 %). The overall satisfaction was rated 55.6 % with 5/5 points on the Likert scale and 44.4 % with 4/5 points.

7.3 Run-time Performance Evaluation

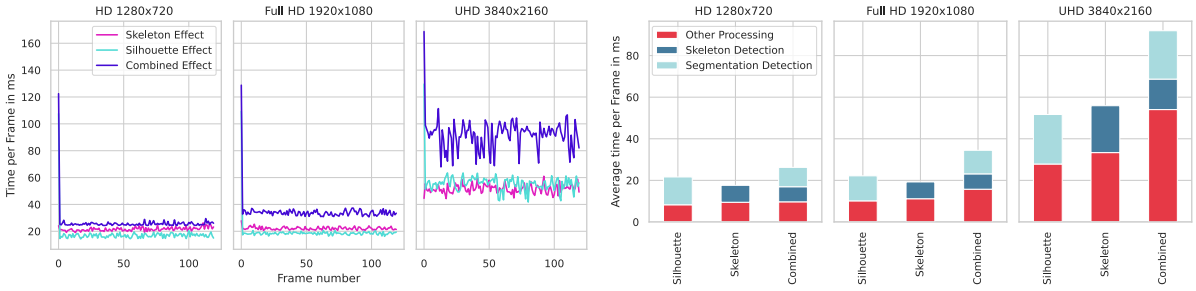
We tested the performance of Lumo with the following setup. Tests on mobile were performed in live preview mode, tested on a 3rd generation Apple iPad Pro 11-inch, equipped with an Apple M1 Chip and 8 GB RAM.

We conducted a run-time analysis using a test dataset comprising three 7-second-long, H.264/MPEG-4 AVC encoded RGB-videos with a frame rate of 24.86 Frames-per-Second (FPS) in three different resolutions: High Definition (HD) at 1280×720 pixels, Full High Definition (FHD) at 1920×1080 pixels and Ultra High Definition (UHD) at 3840×2160 pixels.

Tests were performed by applying a Skeleton effect, a Silhouette effect, and a Combined effect, which uses both body pose and segmentation estimation in its pipeline. For the final evaluation, we sampled 120 frames uniformly from the measurements. This step was necessary because the app's preview system skips frames to ensure interactivity when processing times exceed a certain threshold. Please view the supplementary material for details on the measurement setup and a per-node breakdown of timings.

The performance measurement in Fig. 11 demonstrates the real-time processing capabilities of the prototype app in HD and FHD, while maintaining interactivity at UHD resolution by dropping frames. We observe an initial spike in processing time in Fig. 11(a) attributed to loading the neural networks for body pose and segmentation detection. Processing time per frame is not strictly linear with increased input resolution, and could be influenced by proprietary Apple libraries used for up- or downsampling, tile-based rendering, and caching mechanisms.

In Fig. 11(b) we observe that processing time per frame increases with the effect complexity and detection of features required. Our approach is implemented in a pipeline-parallel manner. Thus, during the processing and rendering of one frame the segmentation and pose detection can already be performed for the next frame, which results in a noticeably more fluid experi-



(a) Processing time per frame

(b) Average processing time

Figure 11: Processing performance graphs for the applied effects by resolution. In (b), the processing time per frame is broken down by processing step. The processing nodes included in the "Other processing" category depend on the effect, and include nodes responsible for neon glow, day-to-night, line rendering, skeleton construction, silhouette detection, and blending and compositing.

Table 1: Peak RAM consumption of the app for different video resolutions (in pixels) and applied effects.

Video Resolution	Silhouette Effect	Skeleton Effect	Combined Effect
1280×720	113.96 MB	150.30 MB	133.23 MB
1920×1080	175.50 MB	125.80 MB	278.93 MB
3840×2160	444.96 MB	460.77 MB	2.43 GB

ence for the user, particularly when applying complex effects. We observe a shorter average segmentation detection time in the Combined Effect compared to the Silhouette effect, possibly due to internal resource optimizations in the Apple Vision Framework.

In Tab. 1 we observe, that the peak RAM consumption of the app increases with the input video resolution and effect complexity overall, but may display unexpected variations due to device-internal optimizations. Processing easily fits into the test system's RAM even at UHD resolutions.

7.4 Limitations

The presented approach has conceptual and technical limitations that can be addressed in future work.

The quality of the achieved effect is dependent on the detection accuracy of the computer vision models used for body pose estimation and segmentation provided by the Apple Vision framework. However, the detection performance for human body pose may be reduced under certain conditions. For instance, if some limbs are not visible in the frame, if the scene is was created from an unusual camera perspective (e.g. top-down), if subjects wear flowy garments (e.g. a wedding dress) or if the input video footage is very dark, body pose estimation accuracy is often degraded as Apple's developer documentation [Doc23] states. In some cases, shadows are mistakenly detected as humans resulting in undesirable artifacts when the effect is applied. In the case of body pose-based effects, the order of bone rendering is not depth-aware, which can result in skeleton joints or light trails being rendered in front of the body, even if they are occluded by other body parts.

8 CONCLUSIONS & FUTURE WORK

We presented a mobile framework for capturing and processing multi-dimensional videos to synthesize motion-based video effects. Our approach provides an exceptional level of artistic control and flexibility in creating motion-based video effects. We believe that our framework and mobile app will prove to be a valuable tool for professionals and amateurs alike, opening up new possibilities for artistic expression and, in particular, enabling non-professional users to design and share their own content-aware video effects.

While we currently extract segmentation and pose information from videos, the advanced sensor and processing capabilities of modern mobile devices offer exciting avenues for future work. Incorporating 3D data and (estimated) depth information is one such area that we plan to explore. By leveraging depth data obtained from sensors or depth estimation techniques, we can expand our range of visual effects by using the depth information to blend layers and simulate interactions with 3D elements. This addition has the potential to provide new opportunities for artistic expression and further variation in the possible visual effects.

REFERENCES

[Bou07] Simon Bouvier-Zappa, Victor Ostromoukhov, and Pierre Poulin. "Motion cues for illustration of skeletal motion capture data". In: *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*. 2007, pp. 133–140.

[Cha20] Sammi Chan. "Latest trend in China: Glow Stick Dance Challenge". In: *South China Morning Post* (May 2020).

[Chi88] John P. Chin, Virginia A. Diehl, and Kent L. Norman. "Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '88. Washington, D.C., USA: Association for Computing Machinery, 1988, pp. 213–218. ISBN: 0201142376. DOI: 10.1145/57167.57203.

[Col03] J.P. Collomosse and P.M. Hall. "Cartoon-Style Rendering of Motion from Video". In: *Vision, Video, and Graphics (VVG) 2003*. Ed. by Peter Hall and Philip Willis. The Eurographics Association, 2003. ISBN: 3-905673-54-1. DOI: 10.2312/vvg.20031016.

[Col05] John P Collomosse, David Rowntree, and Peter M Hall. "Rendering cartoon-style motion cues in post-production video". In: *Graphical Models* 67.6 (2005), pp. 549–564.

[Cut02] James E Cutting. "Representing motion in a static image: constraints and parallels in art, science, and popular culture". In: *Perception* 31.10 (2002), pp. 1165–1193.

[Doc23] Apple Developer Documentation. *Detecting Human Body Poses in Images*. Feb. 2023.

[Kwo12] Yunmi Kwon and Kyungha Min. "Motion Effects for Dynamic Rendering of Characters". In: *Lecture Notes in Electrical Engineering* 181 (2012), pp. 331–338.

[Lew95] James Lewis and James R. "IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use". In: *International Journal of Human-Computer Interaction* 7 (Feb. 1995), pp. 57–. DOI: 10.1080/10447319509526110.

[Lu13] Yijuan Lu and Hao Jiang. "Human movement summarization and depiction from videos". In: *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2013, pp. 1–6.

[Lug19] Camillo Lugaresi et al. "Mediapipe: A framework for building perception pipelines". In: *arXiv preprint arXiv:1906.08172* (2019).

[May21] Maximilian Mayer et al. "MotionViz: Artistic Visualization of Human Motion on Mobile Devices". In: *ACM SIGGRAPH 2021 Appy Hour*. SIGGRAPH '21. Virtual Event, USA: Association for Computing Machinery, 2021. ISBN: 9781450383585. DOI: 10.1145/3450415.3464398.

[Nie05] M. Nienhaus and J. Döllner. "Depicting dynamics using principles of visual art and narrations". In: *IEEE Computer Graphics and Applications* 25.3 (2005), pp. 40–51. DOI: 10.1109/MCG.2005.53.

[Nie08] Marc Nienhaus, Holger Winnemöller, and Bruce Gooch. "Forward Lean–Deriving Motion Illustrations from Video". In: *Proc. SIGGRAPH Asia Sketches* (2008).

[Sch10] Johannes Schmid et al. "Programmable Motion Effects". In: *ACM Trans. Graph.* 29.4 (July 2010). ISSN: 0730-0301. DOI: 10.1145/1778765.1778794.

[Sem19] Amir Semmo et al. "ViVid: Depicting Dynamics in Stylized Live Photos". In: *ACM SIGGRAPH 2019 Appy Hour*. SIGGRAPH '19. Los Angeles, California: Association for Computing Machinery, 2019. ISBN: 9781450363068. DOI: 10.1145/3305365.3329726.

[Ume12] Daiki Umeda, Tomoaki Moriya, and Tokiichiro Takahashi. "Real-Time Manga-like Depiction Based on Interpretation of Bodily Movements by Using Kinect". In: *SIGGRAPH Asia 2012 Technical Briefs*. SA '12. Singapore, Singapore: Association for Computing Machinery, 2012. ISBN: 9781450319157. DOI: 10.1145/2407746.2407774.

ISSN 1213-6972 (print)
ISSN 1213-6964 (online)
Journal of WSCG
http://www.wscg.eu
Vol.31, No-1-2, 2023

Anomaly Detection with Transformers in Face Anti-spoofing

Latifah Abduh
Department of Computer Science
Durham University
Durham, DH1 3LE, UK
0000-0002-6359-311X
latifah.a.abduh@dur.ac.uk

Luma Omar
Department of Computer Science
Durham University
Durham, DH1 3LE, UK
0000-0002-7215-9112
lama.omar@yahoo.com

Ioannis Ivrissimtzis
Department of Computer Science
Durham University
Durham, DH1 3LE, UK
0000-0002-3380-1889
ioannis.ivrissimtzis@dur.ac.uk

ABSTRACT

Transformers are emerging as the new gold standard in various computer vision applications, and have already been used in face anti-spoofing demonstrating competitive performance. In this paper, we propose a network with the ViT transformer and ResNet as the backbone for anomaly detection in face anti-spoofing, and compare the performance of various one-class classifiers at the end of the pipeline, such as one-class SVM, Isolation Forest, and decoders. Test results on the RA and SiW databases show the proposed approach to be competitive as an anomaly detection method for face anti-spoofing.

Keywords

Face presentation attack, Vision Transformer, ResNet, anomaly detection, one-class classification.

1 INTRODUCTION

While face recognition is the biometric authentication method of choice in many application domains, it is still considered extremely vulnerable to presentation attacks. In such attacks, an imposter is trying to gain unlawful access by presenting in front of the system's camera a printed photo, or an electronic screen playing a video of a rightfully registered person. The vulnerability of face recognition systems to such spoofing attacks means that they cannot be safely deployed in security-sensitive applications in uncontrolled environments, as for example ATM machines in the high street. Presentation attack detection (PAD) addresses this problem by developing binary classification algorithms aiming at distinguishing between the genuine, bona fide samples presented to the system's camera, and the imposter ones.

The most common approach to PAD is to train a binary classifier on both the bona fide and the imposter classes. In this case, training and testing are performed within specialised face anti-spoofing databases, which due to the high cost of producing imposter samples have limited variability, raising questions on the generalisation power of the classifier, especially on unseen attacks in scenarios that have not been covered by the testing databases. In particular, while the current state-of-the-art algorithms can show good results on unseen attacks within the same database, and some generalisation power between specific databases, a thorough cross-database validation is expected to show that they do not always generalise well. For example, in [1] all the eleven methods under comparison show HTERs between 24% and 60.6% in cross-database generalisation task from the Replay Attack database to the CASIA-MFSD.

An alternative approach aiming at addressing the generalisation problem is anomaly detection based on one-class training. We note that in the limited testing environments provided by the existing databases, anomaly detection approaches underperform two-class training under most testing protocols. However, they have the conceptually appealing property that they neither attempt to learn specific presentation attacks nor, most importantly, specific environments where such attacks where modelled during the creation of the database. Thus, anomaly detection for face anti-spoofing is still a very active research area [2, 3].

In this paper, we use the Vision Transformer (ViT) [4] and the ResNet [5] as backbones for anomaly detection for face anti-spoofing. Our motivation for using ViT was the observation that while in several computer vision tasks Transformers are replacing Convolutional Neural Networks (CNNs) as the new gold standard, and they have already been proposed for the PAD problem under a two-class training setting [6], they have not been used yet for PAD in the anomaly detection setting.

Regarding the use of ResNet, we note that the size of the receptive field is one of the primary distinctions between a CNN-based model and a transformer-based model. Whereas due to the self-attention mechanism, the transformer is superior in its ability to capture a pixel relation over a long distance [7], nonetheless, it

lacks a reliable way of capturing spatial information within each patch, so it may overlook a crucial spatial local patterns, such as textures. However, CNNs are different in this regard, focusing on textures rather than shapes to identify objects in images [8]. ResNet in particular is a highly efficient neural network architecture and its residual learning methodology addresses the degradation issue which exists in many other CNN models. Thus, overall, we leverage the strengths of two state-of-the-art architectures, a transformer and a CNN to extract reliable features. Our ablation study shows that the combined ViT ResNet backbone gives significant improvement over a single network backbone.

Our main contributions are summarised as follows:

- A novel Anomaly detection Vision Transformer (AnoFormer), with ViT and ResNet in the backbone, for presentation attack detection.
- A comparison of various one-class classification models, showing that a decoder with MSE as a loss function outperforms the other configurations.
- An ablation study showing that the use of a combination of ViT and ResNet in the backbone outperforms the use of single networks.

The rest of the paper is organised as follows. In Section 2 we review the related work. In Section 3 we present the proposed AnoFormer and its implementation details. In Section 4 we present the results, and we briefly conclude in Section 5.

2 RELATED WORK

2.1 Face Anti-spoofing

The earlier machine learning approaches to PAD were based on the extraction of handcrafted features such as Histograms of Oriented Gradient (HOG) [9], Differences of Gaussians (DoG) [10, 11], and Local Binary Patterns (LBP) [12, 13, 14]. Recently, deep learning has replaced traditional feature extraction, and the research focus has shifted towards the design of the most suitable neural network architectures. Yang *et al.* [15] were the first to use CNNs in face anti-spoofing, while, [16, 17], followed by [18, 19], proposed approaches competitive to the then state-of-the-art.

Better approaches were found to enhance results such as including Central Difference Convolutional Networks (CDCN), and transformers [1, 20, 21], and the use of a combination of more than one deep network type as in [22]. A newer approach is to rely on the use of independently trained neural networks to infer depth information [23, 24, 25], or Near Infrared (NIR) information [26]. Most recently, in this direction of work, [27] proposed the use of a dual-stream CNN framework. One stream uses learnable frequency filters to extract features in the frequency domain that are less influenced by variations in sensors and lighting, while the other stream uses standard RGB images to supplement these features. A hierarchical attention module is used to combine the information from these two streams at different stages of the CNN.

2.2 Anomaly detection in face anti-spoofing

In principle, applying anomaly detection to face anti-spoofing problems should lead to improved generalization capabilities, since it makes no assumptions about the type of attack or the environment in which it took place. Arashloo *et al.* [2] was the first to use anomaly detection for face anti-spoofing, using One-Class SRCs and One-Class SVMs as generative and non-generative classifiers respectively. One class GMMs were used in [28] and [29], while combinations of CNNs with one class classifiers were proposed in [30, 31, 32, 33]. Baweja *et al.* [34] introduced in the training a normally distributed pseudo-negative class and a pairwise confusion loss. Feng *et al.* [3] proposed a residual learning framework with a spoof cue generator and an auxiliary classifier. Abduh and Ivrissimtzis [35] used a convolutional autoencoder and augmented the training set with images from the in-the-wild.

2.3 Transformers

Transformers are for some years now the de facto standard in natural language processing (NLP) applications and recently have been established as a state-of-the-art computational technique in many computer vision problems too. The potential of the transformers in computer vision tasks was demonstrated by the groundbreaking Vision Transformer (ViT) [4], which is still in wide use, and it is the network that we use here. Liu *et al.* [36], introduced the Hierarchical Vision Transformer Using Shifted Windows, showing that it works very well as a general-purpose backbone for computer vision problems.

Regarding the use of transformers for anomaly detection in computer vision tasks, Mishra *et al.* [37] proposed a transformer based network for detecting and locating anomalous regions in images. By incorporating transformers, their method is sensitive to the spatial details of the patches, which are analyzed by a Gaussian mixture density network to identify anomalous regions. Lee *et al.* [38] proposed AnoVit, a ViT-based encoder-decoder for anomaly detection and localization, while Mukherjee *et al.* [39] proposed OCFormer, a one-class transformer for image classification.

Regarding the use of transformers in face anti-spoofing, George and Marcel [6] proposed a ViT-based model for the zero-shot PAD. Wang *et al.* [20] cross-layer

relation-aware attentions (CRA) and hierarchical feature fusion (HFF). Liu and Liang [40] proposed the Modality-Agnostic ViT (MA-ViT), using early fusion to aggregate data from all training modalities, improving the model's performance on arbitrary modal attacks. Finally, Huang *et al.* [41] use an adaptive transformer model for few-shot PAD across various databases. To the best of our knowledge, there is no study of transformer-based anomaly detection techniques for PAD.

3 THE ANOFORMER

The proposed method uses feature vectors provided by the pre-trained ViT and ResNet[5], which are then processed by a one-class classification technique. In our experimental study in Section 4 we show results obtained by the use of isolation forests and one-class SVMs. However, our focus is on training a decoder of the feature vectors and then comparing the reconstruction error against a threshold to take the classification decision.

3.1 Architecture

The architecture of the Anoformer is illustrated in Fig. 1. The backbone networks are ViT, which, has already demonstrated its potential as an embedding extractor for the face PAD problem in [6] where a two-class training of the ViT feature vectors gave results competitive to the state-of-the-art, and ResNet-18, both pre-trained on ImageNet [42].

Regarding the choice of specific ViT architecture, we first note that our proposed model can work with any version of ViT, providing compatibility with future improvements to ViT. Here, we employed the Data-Efficient Image Transformer [43] (DeiT-Base), which is an improved version of ViT of lightweight design. To learn diverse features, the training dataset's bona fide images are fed into the ViT and ResNet networks. The ViT divides the input image into patches and uses the extracted features as the sequence input for the transformer, followed by transformer layers. The transformer encoder layer is composed of multiple encoder blocks, each with multi-head self-attention (MSA) and multi-layer perceptron (MLP), as in [4].

The image patches undergo linear transformation to produce the queries (Q), keys (K), and values (V) of the self-attention mechanism, with position encoding (PE) added to keep track of each input token's position. The MLP contains two linear layers with a GELU activation function. Finally, the encoded patches are reshaped and projected into a reconstruction vector via a learned projection matrix.

$$\text{Attention}(Q,K,V) = \text{Softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) V \quad (1)$$

The ViT leverages the attention mechanism which gathers information from the entire input sequence. Self-attention layers scan through a sequence of elements and update them based on the information obtained from the entire sequence. In essence, they simulate explicitly every pair-wise interaction that occurs between the components of the input sequence. Thus, the self-attention maps, which are learned separately for each layer, are necessary for a transformer model to encode the dependencies between input tokens.

Fig. 2 shows attention maps from different ViT layers for a bona fide and an imposter input. We note the difference in ViT's behaviour between the two classes, in that certain prominent facial features such as eyes, nose, and mouth are more prominent in the imposter attention maps.

The decoder decodes the 768×1 reconstruction vector back to the original image shape. We used 4 transposed convolutional layers, with ReLU in between, except for the last layer, which is followed by a sigmoid as the final activation function. The decoder part of the Anoformer was trained with the features of the bona fide images extracted from ViT and ResNet. The decoder is trained with the objective of minimizing the error between the input and the output of the network, aiming at reconstructing bona fide images with high fidelity.

3.2 Implementation

To avoid contributions from the input image's background, we use the MTCNN algorithm for face detection, and cropped the images, retaining the face regions only. Then the images are rotated to have the eye centres horizontally aligned and finally resized to 224×224, which is the native resolution of the ViT transformer pre-trained on ImageNet.

In the final binary classification section, we used an Adam optimizer with initial learning rate of 1e-4 and batch size 16. A label smoothing cross entropy loss function was used to train the classifier. The MLP head is the binary classifier which contains two fully-connected layers of dimensions 512 and 2. The development environment was the PyTorch running on a PC with an Intel CPU, 64 GB of RAM, and Google Colab GPU.

Our backbone consists of two parts, the first part is the Deit-Base [43] which spatial position embeddings, to improve image processing capabilities which has an output embedding dimension of 768. The second component is a ResNet-18 network with 18 layers that have once more been trained on ImageNet. The size of the output from the ResNet is 2048. Therefore, in order to bring the size of the outputs (features) produced by ResNet down to the same level as those produced by transformer 768, we add one dense layer at the very end

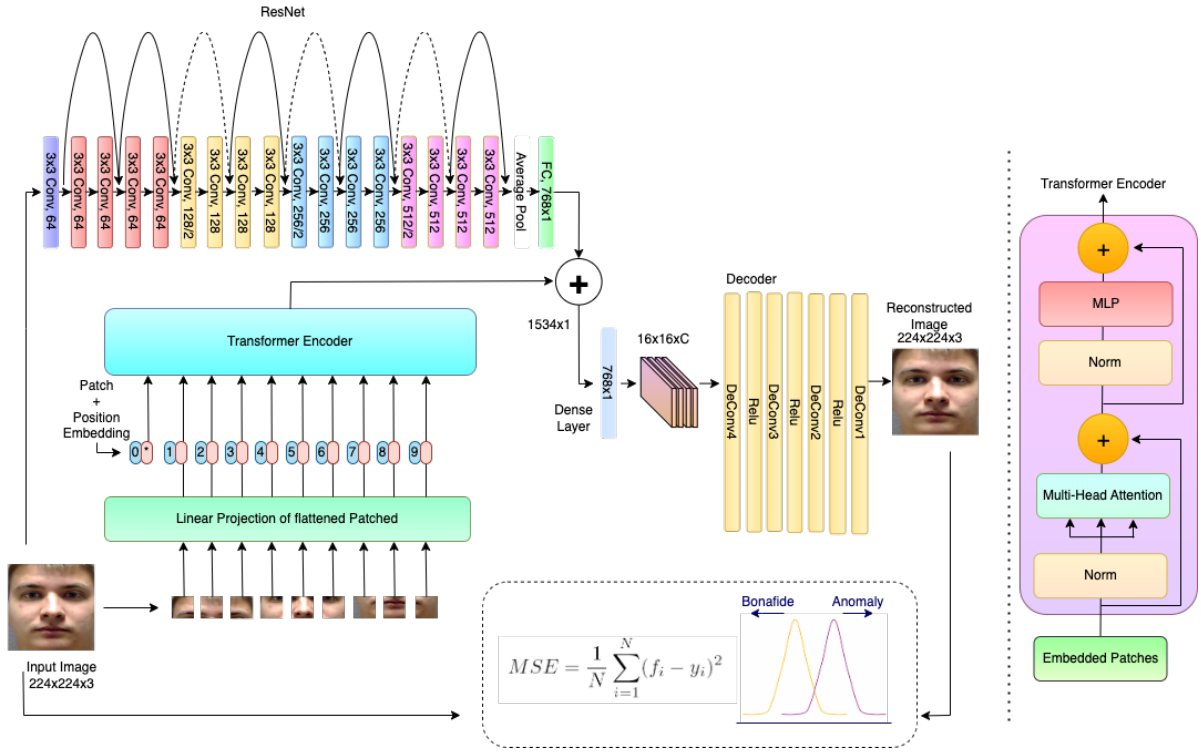


Figure 1: Architecture of the Anoformer.

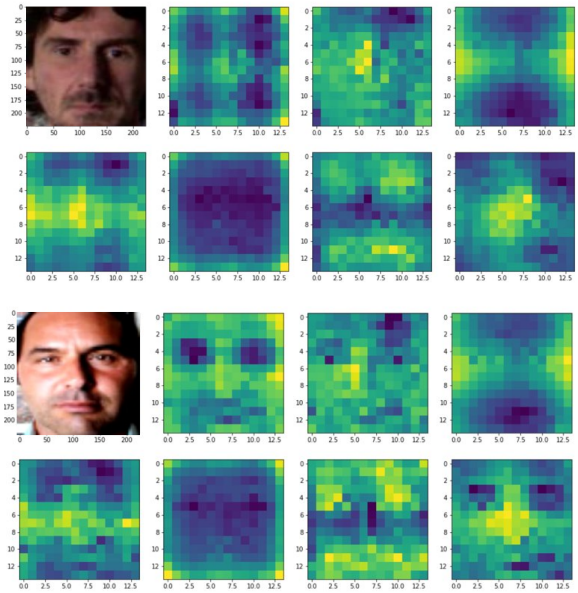


Figure 2: Visualisation of input faces from the Replay Attack database, and the attention maps of several ViT layers. **Top two rows:** a bona fide face. **Bottom two rows:** an imposter face.

of the network. Finally, then concatenate the two feature sets, the one from Deit-Base and that from Resnet-18, to create a vector of 1536 features. This data is then compressed to a size of 768 by adding one dense layer before being sent to the decoder.

The decoder was trained under the Mean Squared Error (MSE) loss function. While it is a pixel-level loss, assuming independence between pixels, it has been repeatedly shown that it works very well in practice, and its simplicity and the fact that it was supported by our development environment led to fast training times.

Regarding the computation of the anomaly score, that is, the error between the input and the reconstructed image, the natural choice is to use the loss function itself, and thus, we use MSE as our default. We experimented with other error metrics, such as Cosine Similarity, the Structural Similarity Index (SSIM), and the Frechet Inception Distance (FID) score [44]. We found that FID performed comparably to MSE, even though the decoder was trained with MSE, and thus, we include some relevant results in Section 4.

4 RESULTS

4.1 Databases

Our experiments were performed on two commonly used face anti-spoofing databases, the *Replay-Attack* (RA) [13], and the *Spoof in the Wild* (SiW) [17]. RA is a low-resolution dataset, containing live and spoof videos from 50 subjects, comprising three different presentation attack species, while SiW is a high-resolution dataset with live and spoof videos from 165 subjects, comprising eight different presentation attack species. The larger number of subjects, the larger number of

presentation attack species, and the higher variability in subject poses and lighting conditions, mean that SiW poses a more challenging classification problem to tackle. We also note that another advantage of SiW over other publicly available anti-spoofing datasets is its racial variety, including a sufficient number of African, Asian, Caucasian, and Indian subjects. It also has a good split between male and female subjects.

We divided the RA and SiW databases into training, validation, and testing sets with non-overlapping subjects. In the validation and testing datasets of the RA database, we included all three presentation attack species; printed photo, video, and digital photo. In the training and validation datasets of the SiW database, we included three representative attack species; printed photo, replay attack using iPhone, and replay attack using a tablet.

4.2 Evaluation Metrics

We report our results using the APCER, BPCER and ACER metrics, which are the most commonly used error metrics in face anti-spoofing, recommended by the ISO/IEC 30107-3:2023 [45] protocol for testing and reporting on biometric PAD.

The Attack Presentation Classification Error Rate (APCER) measures the performance of the system on attack images, that is, its ability to identify correctly spoof images. Unlike the most commonly used in binary classification problems False Positive Rate (FPR), to compute the (APCER), we compute misclassification rates separately over each attack species, and take the maximum. That is, APCER measures the system's performance under the most challenging type of attack, rather than under the average attack. The bona fide classification error rate (BPCER) is the misclassification rate over the bona fide samples. Finally, the ACER, which is considered a good measure of the overall performance of an algorithm, is just their average ACER=(APCER+BPCER)/2.

The definition of APCER as the maximum of the misclassification rates that are computed separately over each attack species brings to the fore an important methodological problem. When we measure the misclassification rates, do we use a single threshold for all attack species, or do we choose a different threshold for each one of them?

For example, in [6] different thresholds were used, and thus different BPCERs are reported as corresponding to each attack species, even though the dataset of bona fide presentations is one. Here, we use a single threshold for all attacks, firstly because in practice it is unrealistic to expect prior knowledge of the attack species that will inform the choice of threshold, and secondly, because we think it is closer to the spirit of the ISO definition of APCER, that is, to consider the worst case outcome over all attack species, rather than splitting the problem into smaller, easier to tackle sub-problems. In particular, we used the threshold corresponding to the Equal Error Rate (EER) on an independent validation set from the same database as the testing set.

4.3 Anoformer validation

In Tables 1 and 2 we report the results for four different classifiers over the ViT+ResNet backbone, tested on the RA and SiW databases, respectively. The one-class SVM (OC-SVM) is a widely used one-class classification method, being essentially an SVM trained with positively labelled data only, and aiming at maximising the separation of their class from the origin of the coordinate system. The second classifier we used is the Isolation Forest, which is based on decision trees and it is theoretically justified under the assumption that anomalies are "few and different". We note that while this is a very realistic assumption for the face anti-spoofing problem, it is not reflected in the usual PAD evaluation protocols that we also use here. Finally, in the last two rows of the tables, we report error rates for the Anoformer and the Anoformer with the FID metric for the computation of the anomaly score as discussed in Section 3. We notice that the combination of Anoformer with MSE in the reconstruction gives lower ACERs on both databases, and it is the configuration that we will evaluate.

Table 1: ViT + ResNet backbone with various one-class classifiers tested on RA

	ACER	APCER	BPCER
OC-SVM	.31	.26	.36
Isolation Forest	.33	.27	.40
Anoformer MSE	**.13**	**.23**	**.03**
Anoformer FID	.19	**.23**	.16

Table 2: ViT + ResNet backbone with various one-class classifiers tested on SiW.

	ACER	APCER	BPCER
OC-SVM	.31	.16	.46
Isolation Forest	.33	**.11**	.55
Anoformer MSE	**.21**	.33	**.10**
Anoformer FID	.22	.35	**.10**

Table 3 shows the results of the ablation study on the backbone of the Anoformer. The ViT + ResNet combination gives on both databases lower ACERs than ViT or ResNet alone, and notably the APCERs and BPCERs are both lower in both cases.

4.4 Performance evaluation

In Table 4, we compare the error rates of the proposed Anoformer against [34], which is a recently published

Table 3: Ablation study for the Anoformer backbone.

	RA			SiW		
	ACER	BPCER	APCER	ACER	BPCER	APCER
ViT	.16	.07	.25	.38	.42	.34
Res	.19	.07	.31	.44	.36	.52
V+R	**.13**	**.03**	**.23**	**.21**	**.10**	**.33**

anomaly detection method that reported results on the same databases and with the same error metrics as us. The results show that the Anoformer gives a lower ACER on both RA and SiW.

Table 4: Performance comparison against [34]

		ACER	APCER	BPCER
RA	[34]	.21	.25	.17
RA	ours	**.13**	**.23**	**.03**
SiW	[34]	.23	**.23**	.23
SiW	ours	**.21**	.33	**.10**

Finally, in Table 5, we report cross-database testing results for the Anoformer with threshold-specific metrics. As expected cross-database testing gives significantly higher ACERs. However, we note that this is mostly due to the higher APCERs.

Table 5: Intra- and cross-database testing of the Anoformer with threshold-specific metrics.

	ACER	BPCER	APCER
RA/RA	.13	.03	.23
SiW/RA	.26	.03	.50
RA/SiW	.27	.13	.42
SiW/SiW	.21	.10	.33

5 CONCLUSION

We proposed Anoformer, an anomaly detection model for PAD, with the pre-trained transformer ViT and the deep CNN Resnet in the backbone, and a one-class trained convolutional decoder for reconstruction. Our experimental results show that the performance of the model is competitive with the current state of the art in anomaly detection for generalised face anti-spoofing.

In the future we would like to test the Anoformer on more databases, and even use test bona fide image from outside the specialised PAD databases. Our aim would be to further demonstrate the generalisation power of anomaly detection.

6 REFERENCES

[1] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou, and G. Zhao, "Searching central difference convolutional networks for face anti-spoofing," in *Proc. CVPR*, 2020. [Online]. Available: 10.1109/CVPR42600.2020.00534

[2] S. R. Arashloo, J. Kittler, and W. Christmas, "An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol," *IEEE Access*, vol. 5, pp. 13 868–13 882, 2017.

[3] H. Feng, Z. Hong, H. Yue, Y. Chen, K. Wang, J. Han, J. Liu, and E. Ding, "Learning generalized spoof cues for face anti-spoofing," *arXiv:2005.03922*, 2020.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv:2010.11929*, 2020.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778. [Online]. Available: 10.1109/CVPR.2016.90

[6] A. George and S. Marcel, "On the effectiveness of vision transformers for zero-shot face anti-spoofing," in *Proc. IJCB*, 2021, pp. 1–8. [Online]. Available: 10.1109/IJCB52358.2021.9484333

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv:1409.0473*, 2014.

[8] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv:1811.12231*, 2018.

[9] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol, "Face recognition using hog–ebgm," *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1537–1543, 2008.

[10] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bi-linear discriminative model," in *Proc. ECCV*, 2010, pp. 504–517.

[11] B. Peixoto, C. Michelassi, and A. Rocha, "Face liveness detection under bad illumination conditions," in *Proc. ICIP*, 2011, pp. 3557–3560.

[12] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *Proc. IJCB*, 2011, pp. 1–7.

[13] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *Proc. BIOSIG*, 2012, pp. 1–7.

[14] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing based on color texture analy-

sis," in *Proc. ICIP*, 2015, pp. 2636–2640.

[15] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *arXiv:1408.5601*, 2014.

[16] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," in *Proc. IJCB*, 2017, pp. 319–328.

[17] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Proc. CVPR*, 2018.

[18] A. Jourabloo, Y. Liu, and X. Liu, "Face despoofing: Anti-spoofing via noise modeling," in *Proc. ECCV*, 2018, pp. 290–306.

[19] C. Nagpal and S. R. Dubey, "A performance evaluation of convolutional neural networks for face anti spoofing," in *Proc. IJCNN*, 2019, pp. 1–8.

[20] Z. Wang, Q. Wang, W. Deng, and G. Guo, "Face anti-spoofing using transformers with relation-aware mechanism," *IEEE Trans. BBIS*, vol. 4, no. 3, pp. 439–450, 2022. [Online]. Available: 10.1109/TBIOM.2022.3184500

[21] Z. Yu, J. Wan, Y. Qin, X. Li, S. Z. Li, and G. Zhao, "Nas-fas: Static-dynamic central difference network search for face anti-spoofing," *IEEE Trans. PAMI*, vol. 43, no. 9, pp. 3005–3023, 2020.

[22] R. Cai, H. Li, S. Wang, C. Chen, and A. C. Kot, "Drl-fas: A novel framework based on deep reinforcement learning for face anti-spoofing," *IEEE Trans. IFS*, vol. 16, pp. 937–951, 2020.

[23] K.-Y. Zhang, T. Yao, J. Zhang, Y. Tai, S. Ding, J. Li, F. Huang, H. Song, and L. Ma, "Face anti-spoofing via disentangled representation learning," in *Proc. ECCV*, 2020, pp. 641–657.

[24] Z. Yu, X. Li, J. Shi, Z. Xia, and G. Zhao, "Revisiting pixel-wise supervision for face anti-spoofing," *IEEE Trans. BBIS*, vol. 3, no. 3, pp. 285–295, 2021.

[25] Z. Wang, Z. Yu, C. Zhao, X. Zhu, Y. Qin, Q. Zhou, F. Zhou, and Z. Lei, "Deep spatial gradient and temporal depth learning for face anti-spoofing," in *Proc. CVPR*, 2020, pp. 5042–5051. [Online]. Available: 10.1109/CVPR42600.2020.00509

[26] A. Liu, Z. Tan, J. Wan, Y. Liang, Z. Lei, G. Guo, and S. Z. Li, "Face anti-spoofing via adversarial cross-modality translation," *IEEE Trans. IFS*, vol. 16, pp. 2759–2772, 2021.

[27] M. Fang, N. Damer, F. Kirchbuchner, and A. Kuijper, "Learnable multi-level frequency decomposition and hierarchical attention mechanism for generalized face presentation attack detection," in *Proc. WCACV*, 2022, pp. 3722–3731.

[28] O. Nikisins, A. Mohammadi, A. Anjos, and S. Marcel, "On effectiveness of anomaly detection approaches against unseen presentation attacks in face anti-spoofing," in *Proc. ICB*, 2018, pp. 75–81. [Online]. Available: 10.1109/ICB2018.2018.00022

[29] F. Xiong and W. AbdAlmageed, "Unknown presentation attack detection with face rgb images," in *Proc. BTAS*, 2018, pp. 1–9.

[30] D. Pérez-Cabo, D. Jiménez-Cabello, A. Costa-Pazo, and R. J. López-Sastre, "Deep anomaly detection for generalized face anti-spoofing," in *Proc. CVPR*, 2019, pp. 0–0.

[31] S. Fatemifar, S. R. Arashloo, M. Awais, and J. Kittler, "Spoofing attack detection by anomaly detection," in *Proc. ICASSP*, 2019, pp. 8464–8468.

[32] S. R. Arashloo, "Unseen face presentation attack detection using sparse multiple kernel fisher null-space," *IEEE Trans. CSVT*, vol. 31, no. 10, pp. 4084–4095, 2020.

[33] S. R. Arshloo, "Matrix-regularized one-class multiple kernel learning for unseen face presentation attack detection," *IEEE Trans. IFS*, vol. 16, pp. 4635–4647, 2021.

[34] Y. Baweja, P. Oza, P. Perera, and V. M. Patel, "Anomaly detection-based unknown face presentation attack detection," in *Proc. IJCB*, 2020, pp. 1–9.

[35] L. Abduh and I. Ivrissimtzis, "Training dataset construction for anomaly detection in face anti-spoofing," in *Proc. CGVC*, 2021.

[36] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. ICCV*, 2021, pp. 10 012–10 022.

[37] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti, "Vt-adl: A vision transformer network for image anomaly detection and localization," in *Proc. ISIE*, 2021, pp. 01–06.

[38] Y. Lee and P. Kang, "Anovit: Unsupervised anomaly detection and localization with vision transformer-based encoder-decoder," *IEEE Access*, vol. 10, pp. 46 717–46 724, 2022.

[39] P. Mukherjee, C. K. Roy, and S. K. Roy, "Ocformer: One-class transformer network for image classification," *arXiv:2204.11449*, 2022.

[40] A. Liu and Y. Liang, "Ma-vit: Modality-agnostic vision transformers for face anti-spoofing," in *Proc. IJCAI*, 2022, pp. 1180–1186.

[41] H.-P. Huang, D. Sun, Y. Liu, W.-S. Chu, T. Xiao, J. Yuan, H. Adam, and M.-H. Yang, "Adaptive transformers for robust few-shot cross-domain face anti-spoofing," *arXiv:2203.12175*, 2022.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical

image database," in *Proc. CVPR*, 2009, pp. 248–255.

[43] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. ICML*. PMLR, 2021, pp. 10 347–10 357.

[44] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[45] Biometrics IJS, "Iso/iec 30107-3: 2023. information technology - biometric presentation attack detection - part 3: Testing and reporting," *International Organization for Standardization*, 2023.