ISSN 2464–4617 (print) ISSN 2464–4625 (CD-ROM)

Computer Science Research Notes

CSRN 3201

(Eds.)

Vaclav Skala
University of West Bohemia, Czech Republic

30. Jubilee International Conference in Central Europe on
Computer Graphics, Visualization and Computer Vision
WSCG 2020
Plzen, Czech Republic
May 17 – 20, 2022

Proceedings

WSCG 2022

Proceedings

ISSN 2464–4617 (print) ISSN 2464–4625 (CD-ROM)

ISSN 2464–4617 (print) ISSN 2464–4625 (CD-ROM)

Computer Science Research Notes

CSRN 3201

(Eds.)

Vaclav Skala
University of West Bohemia, Czech Republic

30. Jubilee International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision
WSCG 2020
Plzen, Czech Republic
May 17 – 20, 2022

Proceedings

WSCG 2022

Proceedings

ISSN 2464–4617 (print) ISSN 2464–4625 (CD-ROM)

Computer Science Research Notes
CSRN 3201

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

Managing Editor: Vaclav Skala

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

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

ISSN 2464-4617 (Print) ***ISSN 2464-4625*** (CD/DVD)
ISBN 978-80-86943-33-6 *(*electronic)

CSRN 3201

International Program Committee

WSCG 2020

Baranoski,G.(Canada)

Barton,M.(Spain)

Baum,D.(Germany)

Benger,W.(Austria)

Bouatouch,K.(France)

Bourke,P.(Australia)

Carmen,M.J.(Spain)

Carmo,M.B.(Portugal)

Drakopoulos,V.(Greece)

Galo,M.(Brazil)

Gdawiec,K.(Poland)

Giannini,F.(Italy)

Gunther,T.(Germany)

Hast,A.(Sweden)

Chaudhuri,D.(India)

Karim,S.A.A(Malaysia)

Klosowski,J.(United States)

Kurt,M.(Turkey)

Lee,J.K.(United States)

Liu,S.(China)

Lobachev,O.(Germany)

Max,N.(United States)

Montrucchio,B.(Italy)

Oliveira,J.F.(Portugal)

Pan,F.(China)

Pedrini,T.(Brazil)

Puig,A.(Spain)

Puppo,D.(Italy)

Raffin,S.(France)

Rodrigues,J.(Portugal)

Rojas-Sola,J.I.(Spain)

Rreshetov,A.(United States)

Savchenko,V.(Japan)

Segura,R.(Spain)

Sirakov,N.M.(United States)

Sousa,A.A(Portugal)

Thalmann,D.(Switzerland)

Tokuta,A.(United States)

Westermann,R.(Germany)

Wu,S.T.(Brazil)

Wuethrich,C.(Germany)

Zwettler,G.(Austria)

CSRN 3201

Board of Reviewers

WSCG 2022

Al-Darraji,S.(Iraq)
Baranoski,G.(Canada)
Barton,M.(Spain)
Baum,D.(Germany)
Benger,W.(Austria)
Bouatouch,K.(France)
Bourke,P.(Australia)
Burova,I.(Russia)
Cakmak,H.(Germany)
Campagnolo,L.Q(Brazil)
Carmen,M.J.(Spain)
Carmo,M.B.(Portugal)
Czapla,(Poland)
David,V.(France)
De Martino,J.M.(Brazil)
Delibasoglu,I.(Turkey)
Drakopoulos,V.(Greece)
Dziembowski,A.(Poland)
Elloumi,N.(Tunisia)
Galo,M.(Brazil)
Gdawiec,K.(Poland)
Giannini,F.(Italy)
Goncalves,A.(Portugal)
Grabska,E.(Poland)
Grajek,T.(Poland)
Gunther,T.(Germany)
Hast,A.(Sweden)
Heil,R.(Sweden)
Hu,C.(Taiwan)
Hu,P.(Belgium)
Chaudhuri,P.(India)
Jacek,K.(Poland)
Karim,S.A.A.(Malaysia)
Kerdvibulvech,C.(Thailand)
Klosowski,J.(United States)
Kuffner dos Anjos,R.(U.K.)
Kurt,M.(Turkey)
Lee,J.K.(United States)
Lefkovits,S.(Romania)
Lisowska,A.(Poland)
Liu,S.(China)
Lobachev,O.(Germany)
Marinkovic,V.(Serbia)
Marques,R.(Spain)
Max,N.(United States)
Meyer,A.(France)
Mieloch,D.(Poland)
Montrucchio,B.(Italy)
Nawfal,S.(Iraq)
Nguyen,S.(Viet Nam)
Norhaida,M.(Malaysia)
Oliveira,J.F.(Portugal)
Pagnutti,G.(Italy)
Pan,R.(China)
Papaioannou,G.(Greece)
Pedrini,H.(Brazil)
Perez,S.(Spain)
Phan,A.(Viet Nam)
Pintus,R.(Italy)
Pombinho,P.(Portugal)
Puig,A.(Spain)
Puppo,E.(Italy)
Raffin,R.(France)
Ramires Fernandes,A.(Portugal)
Reshetov,A.(United States)
Rodrigues,J.(Portugal)
Rodrigues,N.(Portugal)
Rojas-Sola,J.I.(Spain)
Romanengo,C.(Italy)
Sacco,M.(Italy)

Sardana,D.(United States)
Sarwas,G.(Poland)
Savchenko,V.(Japan)
Scaramuccia,S.(Italy)
Segura,R.(Spain)
Seracini,M.(Italy)
Shamima,Y.(United States)
Schiffner,D.(Germany)
Sintorn,E.(Sweden)
Sirakov,N.M.(United States)
Sousa,A.A.(Portugal)
Tandianus,B.(Singapore)
Tarhouni,N.(Tunisia)
Thalmann,D.(Switzerland)
Tokuta,A.(United States)
Tourre,V.(France)
Tytkowski,K.(Poland)
Westermann,R.(Germany)
Wiegreffe,D.(Germany)
Wu,S.(Brazil)
Wuethrich,C.(Germany)
Yoshizawa,S.(Japan)
Zahariev,P.(Bulgaria)
Zavala-De-Paz,J.P.(Mexico)
Zwettler,G.(Austria)

CSRN 3201

Computer Science Research Notes

WSCG 2022 Proceedings

Contents

FULL papers

SHORT papers

POSTERS

Region of interest in JPEG

David Barina Ondrej Klima

Centre of Excellence IT4Innovations
Faculty of Information Technology
Brno University of Technology
Bozetechova 1/2, Brno
Czech Republic
{ibarina,iklima}@fit.vutbr.cz

Abstract

This paper describes three methods to encode a region of interest (ROI) of arbitrary shape in the JPEG standard. The ROI is part of the image encoded with higher quality than the rest of the image. Two of the described methods are based on thresholding of DCT coefficients. The last method is based on the early termination of the entropy coding phase. All methods are fully compatible with existing JPEG decoders. Using ROI, it is possible to reduce the bitrate to only a fraction of the original value. The results in this paper show that thresholding of original DCT coefficients provides superior performance in terms of the PSNR-B index.

Keywords
JPEG, region of interest, image compression

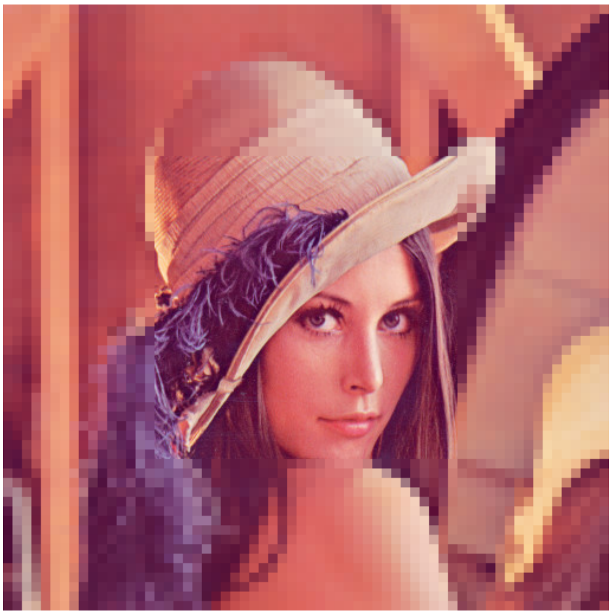


Figure 1: The image was compressed using the JPEG method. The face in the image has been selected as a region of interest.

1 INTRODUCTION

In some applications, certain areas of the image are more important than the rest. We can imagine, e.g., human faces for face recognition (Figure 1) or license plates for automatic license plate recognition system. Then it may be desirable to compress these areas with higher quality than the rest of the image. In image processing, these important parts of images are often referred to as the regions of interest (ROI).

JPEG [3] is the most widely used image format in the world. The ROI functionality is already included in some newer image compression standards, such as the JPEG 2000. However, the functionality is missing in the original JPEG format (ITU-T T.81 / ISO/IEC 10918-1). This paper describes three methods compatible with the original JPEG standard for encoding images with a ROI. The principle of these methods is to drop less important details in non-ROI areas. All three methods use only one common quantization matrix (per component). The resulting bitstream fully conforms to the JPEG standard (no modification to existing decoders is required).

The rest of the article is organized as follows. Section 2 familiarizes the reader with the necessary background for understanding the rest of this article. Section 3 presents the three proposed methods for encoding a ROI in the JPEG standard. Section 4 evaluates these methods using the PSNR-B quality assessment index. Finally, Section 5 summarizes the paper.

2 BACKGROUND

Because this article discusses the JPEG format's internalities, it is necessary to explain how this format works. This 1992 standard, commonly referred to as JPEG, is primarily intended for lossy image compression. For lossy compression, JPEG supports sequential and progressive image data transfer. The input image is processed in the YCbCr color model. Individual components can be subsampled horizontally and vertically. The chromatic components Cb and Cr are typically subsampled in one or both directions in half (4:2:2 and 4:2:0). The Y, Cb, and Cr components of the color model are further processed separately. The following text describes

https://www.doi.org/10.24132/CSRN.3201.1

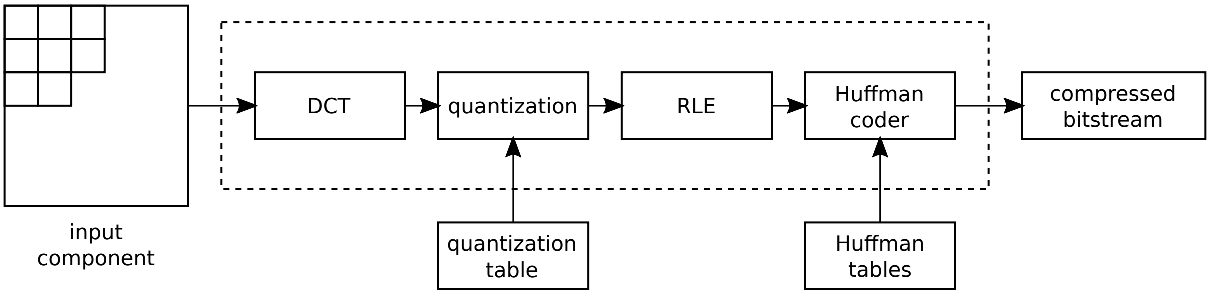


Figure 2: Simplified JPEG encoder scheme. The same scheme is applied to each image component.

the scheme shown in Figure 2. Each component is divided into blocks of 8 × 8 samples. This division is the main reason for the block artifacts in images with a high degree of compression. The 8 × 8 block is the fundamental data unit of the JPEG format. On each such block, its discrete cosine transform (DCT) [1] is then computed. The result gives 8 × 8 DCT coefficients. The DCT coefficient at position (0,0) indicates the shift from zero, and it is called the DC coefficient. The remaining 63 coefficients are referred to as the AC coefficients. They indicate the weights with which the corresponding two-dimensional cosine function is present in the block. The procedure just described is fully invertible. No data is lost. The next step is to quantize the coefficients. To do this, both the encoder and decoder must receive a so-called quantization table. This is a table with the integer values by which the corresponding DCT coefficient is to be divided during compression. Subsequently, each coefficient is rounded to the nearest integer. During decompression, the coefficients are in turn multiplied by these values. Because the coefficients' values have been rounded to integers, the reconstructed coefficient is quantized to several levels. The quantization table determines how much data is lost during compression. Higher values mean coarser quantization, i.e., more information loss. Therefore, the values in the table are created concerning the target quality and the human psychovisual model. A different table is used for the luminance component Y and the chromatic components Cb and Cr. The reason for this is the greater sensitivity of the human eye to luminance than to chromatic components. Quantized coefficients are further processed in the zig-zag scan. In the beginning, this scan processes coefficients with lower frequencies. These typically have a higher amplitude and are therefore more likely to remain non-zero even after quantization. Conversely, the coefficients at the end of the scan are more likely to be zero. A sequence of linearized coefficients is further subjected to a variant of RLE coding (only zeros are considered). The last step of the compression is Huffman or arithmetic coding. Both variants work based on supplied tables.

ROI provides an overall improvement in perceived image quality compared to conventional coding at the same bitrate. Alternatively, ROI is a way to save some bitrate on less essential parts of the image. In general, a ROI can be of arbitrary shape. However, in connection with the JPEG standard, this shape is constrained by 8 × 8 block nature of the JPEG coding. Outside of the JPEG environment, these constraints may differ. For example, in the JPEG 2000, a direct successor to the JPEG format, the ROI can be defined on a pixel basis.

The method used in the JPEG 2000 (Part 1) [2] for encoding images with a ROI is called the Maxshift method. When an image is encoded using JPEG 2000, a wavelet transform [1] of the image is initially computed. This step is followed by entropy coding of resulting coefficients, which processes the coefficients bitplane-by-bitplane. Maxshift method shifts up the wavelet coefficients so that the bits associated with the ROI are encoded in higher bit-planes than the bits associated with the background. In contrast to the methods proposed in this paper, an encoder can place those bits in the bitstream before the bits corresponding to the background.

3 REGION OF INTEREST IN JPEG

This section presents the three strategies we have chosen to implement ROI in JPEG format. For compatibility, all of these strategies must encode ROI and non-ROI blocks in a single interleaved scan with a single quantization table.

3.1 Baseline process

The JPEG transforms an input image (each component) using disjoint 8 × 8 blocks of pixels. Particularly, the nth image block is transformed from image domain s^n to frequency domain S^n using

$$S^n_{v,u} = \sum_{y=0}^{7}\sum_{x=0}^{7} s^n_{y,x}\, g_{v,u,y,x}, \tag{1}$$

where the

$$\left\{ g_{v,u,y,x} = C_v C_u \frac{1}{4} \cos\left(\frac{v\pi(2y+1)}{16}\right) \cos\left(\frac{u\pi(2x+1)}{16}\right) \right\}_{0\le v,u<8} \tag{2}$$

is set of DCT basis vectors, and

$$C_u = \begin{cases} 1/\sqrt{2} & \text{if } u = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

The next step in the process is a quantization of the coefficients. The JPEG uses uniform scalar quantization

$$S^{q,n}_{v,u} = \operatorname{round}\left(S^n_{v,u}/Q_{v,u}\right). \tag{4}$$

The quantized coefficients $S^{q,n}$ are then encoded by entropy coder using zig-zag scan $\zeta_{v,u} : \mathbf{N}_0 \times \mathbf{N}_0 \to \mathbf{N}_0$. Our implementation uses adaptive Huffman coding. The Huffman coding uses two Huffman tables per component, one for the DC coefficients $(u, v = 0)$, and the other for AC coefficients $(u, v \neq 0)$. The Huffman table for AC coefficients is indexed by the pair (zero-run length, non-zero coefficient). The special pair (0, 0) is used if all coefficients up to the end of the current block are zero. This symbol is known as the end-of-block symbol (EOB).

3.2 Coefficient thresholding

This section aims to encode a region of interest (foreground) with higher quality than the rest of the image (background). Let $\mathscr{B}$ be a set of background block indices. Let $\mathscr{F}$ be a set of foreground block indices. Note that $\mathscr{B} \cup \mathscr{F}$ covers all block indices in an image.

Our first idea was to alter the DCT coefficients. To suppress the background details, we apply the following nonlinear transform

$$\hat{S}^n_{v,u} = \begin{cases} \rho_\lambda\left(S^n_{v,u}\right) & \text{if } n \in \mathscr{B}, \\ S^n_{v,u} & \text{if } n \in \mathscr{F}, \end{cases} \tag{5}$$

where

$$\rho_\lambda(x) = \begin{cases} x & \text{if } |x| > \lambda, \\ 0 & \text{if } |x| \leq \lambda \end{cases} \tag{6}$$

is hard thresholding operator [1] with the threshold λ. The process continues with quantization

$$S^{q,n}_{v,u} = \operatorname{round}\left(\hat{S}^n_{v,u}/Q_{v,u}\right) \tag{7}$$

as usual. Effectively, this procedure uses uniform scalar dead-zone quantization [5] for background blocks. The transformed coefficients inside interval $[-\lambda, +\lambda]$ are quantized to zero. The interval $[-\lambda, +\lambda]$ is called the "dead zone". The procedure just described will hereinafter be referred to as Coefficient thresholding.

3.3 Quantized coefficient thresholding

Our second idea was to threshold quantized DCT coefficients. This method has the advantage that the entropy coder does not have to encode many symbols for background blocks (since zeros have replaced some). Formally, this step can be described as

$$S^{q,n}_{v,u} = \begin{cases} \rho_\lambda\left(\operatorname{round}\left(S^n_{v,u}/Q_{v,u}\right)\right) & \text{if } n \in \mathscr{B}, \\ \operatorname{round}\left(S^n_{v,u}/Q_{v,u}\right) & \text{if } n \in \mathscr{F}. \end{cases} \tag{8}$$

Also this procedure effectively uses uniform scalar dead-zone quantization for background blocks. However, the choice of λ threshold differs from that described in the previous section. This will from now on be called Quantized coefficient thresholding.

3.4 Cutting of coefficients

Our last idea was to modify the Huffman coding so that the EOB symbol is inserted prematurely for background blocks. In other words, we cut the coded sequence after some fixed number of coded coefficients (however, the DC coefficient is always fully encoded). This corresponds to resetting the coefficients

$$S^{q,n}_{v,u} = \begin{cases} \sigma_{\mu,v,u}\left(\operatorname{round}\left(S^n_{v,u}/Q_{v,u}\right)\right) & \text{if } n \in \mathscr{B}, \\ \operatorname{round}\left(S^n_{v,u}/Q_{v,u}\right) & \text{if } n \in \mathscr{F}, \end{cases} \tag{9}$$

where

$$\sigma_{\mu,v,u}(x) = \begin{cases} x & \text{if } \zeta_{v,u} < \mu, \\ 0 & \text{if } \zeta_{v,u} \geq \mu, \end{cases} \tag{10}$$

in the zig-zag sequence $\zeta_{v,u}$ starting at some fixed position μ. Note that the encoder could have also inserted the EOB symbol before this fixed position. The disadvantage of this procedure is the absence of high frequencies in the decoded image. The procedure will be called Cutting of coefficients.

4 EVALUATION

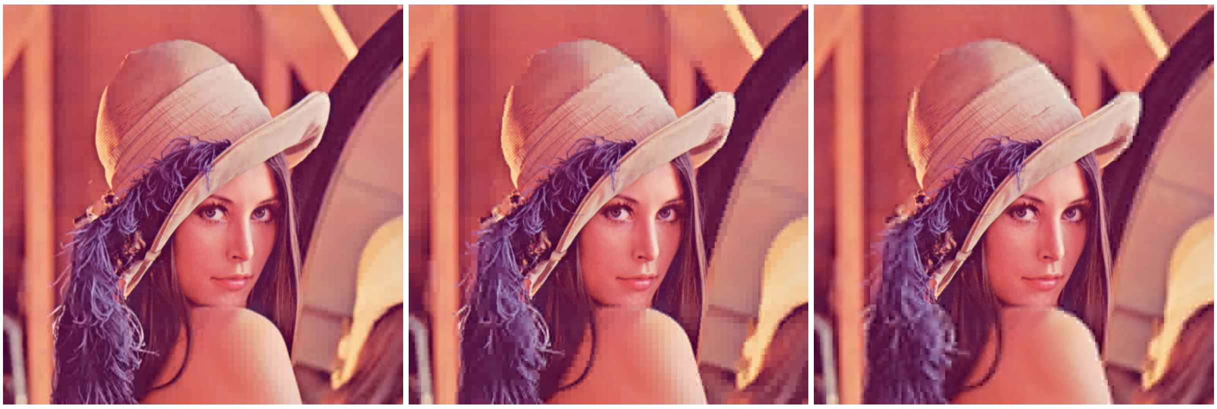
Since methods proposed in the previous section effectively increase quantization step size, blocking artifacts generally become more visible. Yim and Bovik [4] proposed a block-sensitive quality assessment index, named PSNR-B. The PSNR-B modifies PSNR by including a blocking effect factor. We use the PSNR-B index to evaluate the qualitative performance of the proposed methods. The index for reference image $\mathbf{x}$ and decompressed image $\mathbf{y}$ is defined as

$$\text{PSNR-B}(\mathbf{x}, \mathbf{y}) = 10 \log_{10} \frac{255^2}{\text{MSE-B}(\mathbf{x}, \mathbf{y})} \tag{11}$$

where

$$\text{MSE-B}(\mathbf{x}, \mathbf{y}) = \text{MSE}(\mathbf{x}, \mathbf{y}) + \text{BEF}(\mathbf{y}) \tag{12}$$

where BEF is the blocking effect factor. Further details are given in [4].



(a) (b) (c)

Figure 3: The Lenna with 3 bpp bitrate and ROI fixed at quality $q = 100$. From left: (a) Coefficient thresholding, (b) Quantized coefficient thresholding, and (c) Cutting of coefficients.

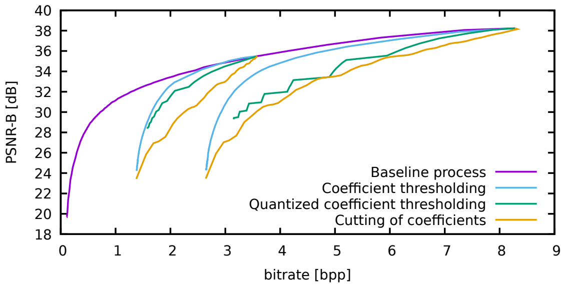


Figure 4: Results of the experiment on the Lenna image with her face used as the ROI. Two ROI quality was fixed at (i) $q = 95$ and (ii) $q = 100$.

In our evaluation, we compute quantization matrices based on different quality q—an integer in the interval $[1, 100]$. We compute the scaling factor

$$\beta = \begin{cases} 5000/q & \text{if } q < 50, \\ 200 - 2q & \text{if } q \geq 50. \end{cases} \tag{13}$$

The quantization matrices are then computed as

$$Q_{v,u} = \max\left(\min\left(\frac{\beta \cdot Q^{\text{ref}}_{v,u} + 50}{100}, 255\right), 1\right), \tag{14}$$

where Q^{ref} is the reference luminance or chrominance quantization matrix listed in the JPEG standard.

The performance of the individual methods from the previous section is demonstrated on a standard Lenna test image. Image size is 512×512 pixels. First, we compress the test image without ROI for each of the q in $[1, 100]$. This gives us an idea of what quality we could achieve at the most using ROI. In the next step, Lenna's face was used as the ROI (size 256×256 pixels, thus the ROI occupies exactly 25 % of the image area). For the experiment, we fixed the ROI quality to (i) $q = 95$ and (ii) $q = 100$. Next, we tested all permissible parameterizations of all three methods from the previous section. This gives us three PSNR-B dependences on the bitrate for (i) and three for (ii). All should be dominated by the curve from the beginning of the experiment. The result is shown in Figure 4. The (i) is shown in the left part and (ii) in the right part of the figure. It is clear that the Coefficient thresholding method dominates the other two methods. Furthermore, we see that using this method, we can conveniently halve the bitrate while maintaining full quality in the ROI. The Cutting of coefficients method proves to be the worst of the three. The extreme case of this experiment is shown in Figure 1. In this case, we kept only the DC coefficient in the non-ROI area. Methods discussed in this paper show the same behavior also on other images with a different ROI. The smaller the ROI area to the rest of the image, the more bitrate we can save.

To give an idea of the artifacts caused by the particular methods, the experiment for $q = 100$ and 3 bpp is shown in Figure 3. The original image at $q = 100$ without ROI has an 8.279 bpp bitrate.

We have released the software (complete implementation of the JPEG baseline encoder and decoder) used in this article as open source.[1] We want to emphasize that the JPEG files created in this way are fully compatible with existing decoders.

5 CONCLUSIONS

This paper proposed three novel methods for encoding a region of interest (ROI) in the original JPEG standard. Two of the methods are based on the thresholding of DCT coefficients in the non-ROI blocks. Similarly, the third method is based on the early termination of the entropy coding. The proposed methods allow for a reduction of the bitrate to only a fraction of the original value. The methods are fully compatible with existing JPEG decoders, and the resulting bitstream fully complies with the JPEG standard. The software used in this paper was released as open-source.

[1] https://github.com/xbarin02/jpeg

Acknowledgements

This work has been supported by the Technology Agency of the Czech Republic (TA CR) project Safe and Secure Traffic Systems of New Generation (no. VB01000048).

REFERENCES

[1] Mallat, S. *A Wavelet Tour of Signal Processing: The Sparse Way. With contributions from Gabriel Peyre*. Academic Press, 3 edition, 2009. ISBN 9780123743701.

[2] Taubman, D. S. and Marcellin, M. W. *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Springer, 2004. ISBN 978-1-4613-5245-7. doi: 10.1007/978-1-4615-0799-4.

[3] Wallace, G. K. The JPEG Still Picture Compression Standard. *Communications of the ACM*, 34(4):30–44, Apr. 1991. ISSN 0001-0782. doi: 10.1145/103085.103089.

[4] Yim, C. and Bovik, A. C. Quality assessment of deblocked images. *IEEE Transactions on Image Processing*, 20(1):88–98, 2011. ISSN 1941-0042. doi: 10.1109/TIP.2010.2061859.

[5] Yu, J. Advantages of uniform scalar dead-zone quantization in image coding system. In *2004 International Conference on Communications, Circuits and Systems (IEEE Cat. No.04EX914)*, volume 2, pages 805–808, 2004. doi: 10.1109/ICCCAS.2004.1346303.

Performance Assessment of Diffusive Load Balancing for Distributed Particle Advection

Ali Can Demiralp
RWTH Aachen
Kopernikusstrasse 6
52074, Aachen, Germany

Dirk Norbert Helmrich
Forschungszentrum Jülich GmbH
Wilhelm-Johnen-Strasse
52428, Jülich, Germany

Joachim Protze
RWTH Aachen
Kopernikusstrasse 6
52074, Aachen, Germany

Torsten Wolfgang Kuhlen
RWTH Aachen
Kopernikusstrasse 6
52074, Aachen, Germany

Tim Gerrits
RWTH Aachen
Kopernikusstrasse 6
52074, Aachen, Germany

ABSTRACT

Particle advection is the approach for extraction of integral curves from vector fields. Efficient parallelization of particle advection is a challenging task due to the problem of load imbalance, in which processes are assigned unequal workloads, causing some of them to idle as the others are performing compute. Various approaches to load balancing exist, yet they all involve trade-offs such as increased inter-process communication, or the need for central control structures. In this work, we present two local load balancing methods for particle advection based on the family of diffusive load balancing. Each process has access to the blocks of its neighboring processes, which enables dynamic sharing of the particles based on a metric defined by the workload of the neighborhood. The approaches are assessed in terms of strong and weak scaling as well as load imbalance. We show that the methods reduce the total run-time of advection and are promising with regard to scaling as they operate locally on isolated process neighborhoods.

Keywords

Particle Advection, Distributed Algorithms, Load Balancing

1 INTRODUCTION

Particle advection is an important method for analysis and visualization of vector fields. The idea is to place a set of (often massless) particles within the vector field, and integrate them over time using a numerical integration scheme such as the Runge-Kutta family of methods. Aside from its primary use for estimation of integral curves in vector data, it serves as a basis for various feature extraction methods such as Lagrangian Coherent Structures [HY00].

Modern vector datasets vary from several gigabytes to the upper end of the terabyte range. Considering the size of the data, and the number of particles necessary to represent the domain, parallel compute becomes essential and is in fact a standard tool for particle advection today [PCG+09, PRN+11].

Parallel particle advection suffers from the problem of load imbalance, in which the vector field blocks and/or the particle sets are distributed in an unfair way, leading to some processes idling as others are performing compute. This is economically undesirable as the idle processes waste core-hours, yet are still paid for by the application user in the form of allocated cluster resources or currency. It is also desirable to have low advection times in order to enable real-time adjustments to the particle set and advection parameters. Due to these aspects, many recent approaches focus on the problem of load balancing [ZGYP18, ZGH+18, BPNC19, BPC21].

Although a variety of load balancing approaches already exist, they invariably come with trade-offs. The approach of [PCG+09] offers decent performance, yet involves a task dispatching model that is challenging to implement in a distributed environment. The approach in [ZGYP18] involves transmission of the vector field across processes which may cause significant communication overhead. The approach presented in [ZGH+18] involves a parallel K-D tree construction at each round. The approach of [BPNC19] is shown to maximize resource usage, however may involve a set of serial communications until work is delivered to the requesting process.

https://www.doi.org/10.24132/CSRN.3201.2

In this work, we present a hybrid-parallel particle advection system, coupled with a local load balancing method based on diffusive load balancing [Boi90, Cyb89]. The processes load the blocks of their first neighbors in addition to their own. This partial redundancy enables efficient application of nearest-neighbor family of diffusive load balancing algorithms based on the framework defined by Cybenko [Cyb89]. We introduce two scheduling methods, considering the requirements of the specific problem of particle advection.

2 RELATED WORK

The approach presented in [PCG+09] is an earlier application of hybrid parallelism to particle advection. The processes are divided into groups, each containing a master load balancing process which dynamically distributes the data and the particle subsets to the rest. The method overcomes the scaling issues of a central load balancer, by introducing multiple centers.

The recent approach presented in [ZGYP18] dynamically adjusts the blocks assigned to the processes at each round of advection. The method constructs a dynamic access dependency graph (ADG) from the number of transmissions between neighboring processes. This information is combined with the particle sets to resize and relocate the blocks assigned to the processes.

The dynamic K-D tree decomposition presented in [ZGH+18] is an example of a hybrid approach, combining static data parallelism with dynamic task parallelism. In this method, each process statically loads a larger data block than the region it is assigned to. At each advection round, a constrained K-D tree is constructed based on the current particle set, whose leaves are used to resize the regions of the processes, up to the boundary of their data blocks.

The approach presented in [BPNC19] adapts the lifeline-based load balancing method to the context of parallel particle advection. The method builds on work requesting [MCHG13], in which processes with low load ask processes with higher load to share (often half of) their load. In contrast to random selection of the process to request work from, this approach first constructs a lifeline graph of processes. The requests are not random, and made to the adjacent nodes in the graph, which may forward the request recursively to their neighbors.

3 METHOD

In this section, we first provide an overview of baseline hybrid-parallel particle advection and a review of the mathematical framework for diffusive load balancing as defined in [Cyb89]. We then introduce two methods that are based on diffusive load balancing.

3.1 Hybrid-Parallel Particle Advection

The approach to achieve hybrid parallelism is to parallelize over data among the processes, and parallelize over the tasks among the threads of each process. Within the context of particle advection, this corresponds to distribution of the vector field blocks as well as the related particle subsets to the processes, and integrating the latter using an array of threads.

The following pseudocode outlines the kernel ran by each process. We also provide two points of entry for modular implementation of the load balancing algorithms which will be discussed in the next section.

Algorithm 1: The advection kernel.

```
function Advect $(p, v)$;
Input  : Particles $p$ and vector fields $v$
Output : Advected particles $p'$ and integral curves $i$
while !check_completion(p) do
    load_balance_distribute(p);
    r = compute_round_info(p, i);
    allocate_integral_curves(p, i, r);
    integrate(v, p, p', i, r);
    load_balance_collect(r);
    out_of_bounds_distribute(p, r);
end
```

The outermost condition, *check_completion*, consists of two global collective operations, a gather for retrieving the current active particle counts of the processes, and a broadcast that consists of a single Boolean evaluating to true only when all gathered particle counts are zero.

The *load_balance_distribute* is a function that provides the active particles (as well as the index of the process), and expects the implementer to freely call global or neighborhood collective operations to balance the load of the process for the upcoming round.

The function *compute_round_info* generates the metadata to conduct an advection round, setting the range of particles, tracking vertex offsets and strides for the integral curves, as well as creating a mapping of out-of-bound particles to the neighboring processes.

The *allocate_integral_curves* (re-)allocates the vertex array of the integral curves, expanding it by the number of particles to trace this round times the longest iteration among those particles. Despite potentially wasteful usage of memory, one allocation per round is significantly more efficient than using resizable nested vectors of points. Besides, it is possible to prune the curves based on a reserved vertex either at the end of each round or after the advection kernel has finished, which may also be done in parallel using the execution policies recently introduced to the standard template library.

The *integrate* iterates through the particles of the round, sampling the corresponding vector field at the position of the particle, and integrates the curve based on the interpolated vector using a scheme such as the Runge-Kutta family of methods.

The *load_balance_collect* is an optional function providing a way to recall the particles that have been transmitted to other processes during load balancing. In the case of diffusive load balancing, particles that have gone out-of-bounds after being load balanced to a neighboring process have to be returned to the original process prior to the out-of-bounds distribution stage. This is due to the fact that the neighboring process does not have any information on or topological connection to the neighbors of the original process except itself.

The *out_of_bounds_distribute* is the final stage, in which particles reaching the boundary of a neighbor during the round are transmitted to the corresponding neighboring processes using local collectives.

3.2 Load Balancing

A set of processes arranged in a Cartesian grid may be interpreted as a connected graph where the nodes are the processes and the edges are the topological connections between the processes. Assume the processes are labelled from 0 to n. Define w^t as an n-vector quantifying the work distribution, so that w_i^t is the amount of work to be done by process i at time t. The diffusion model for dynamic load balancing as described by Cybenko in [Cyb89] has the following form:

$$w_i^{t+1} = w_i^t + \sum_j \alpha_{ij}(w_j^t - w_i^t) + \eta_i^{t+1} - c \qquad (1)$$

where α_{ij} are coupled scalars, evaluating to non-zero only when process i and j are topologically connected. Within a Cartesian grid, this corresponds to the immediate neighbors excluding the diagonals. The sum implies that the processes i and j compare their workloads at time t and transmit $\alpha_{ij}(w_j^t - w_i^t)$ pieces of work among each other. The work is transmitted from j to i if the quantity is positive, and vice versa if negative. The term η_i^{t+1} corresponds to new work generated at process i at time t. The term c describes the amount of work performed by the process between time t and $t+1$.

In the next section, we propose a method to compute α_{ij} dynamically based on several local averaging operations, guaranteeing an improved distribution of the workload. A central requirement for the following algorithms for computation of α_{ij} is the availability of information from the first neighbors. Each process is required to additionally have access to the data blocks of its neighbors. We furthermore communicate the local workload for the next round of particle advection to all neighbors dynamically at the beginning of each load balancing step.

Algorithm 2: The lesser mean assignment.

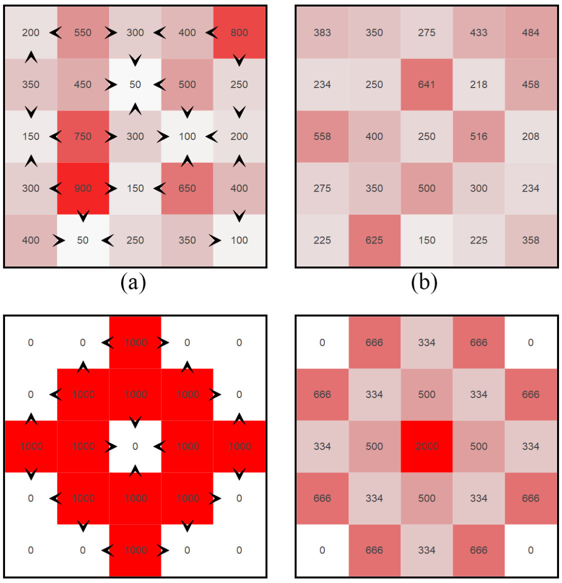
```
function LMA (load, neighbor_loads);
Input  : Local workload amount load, and vector
         of neighbors' workload amounts
         neighbor_loads.
Output : Vector of workload amounts to be sent to
         each neighbor outgoing_loads.
contributors =
  vector<bool>(neighbor_loads.size());
mean = load;
do
    contributors.fill(false);
    sum = load;
    count = 1;
    for i=0; i < neighbor_loads.size(); i++ do
        if neighbor_loads[i] < mean then
            contributors[i] = true;
            sum += neighbor_loads[i];
            count++;
        end
    end
    mean = sum / count;
while There are neighbors above mean in
  contributors;

outgoing_loads =
  vector<uint>(neighbor_loads.size());
outgoing_loads.fill(0);
for i=0; i < neighbor_loads.size(); i++ do
    if contributors[i] == true then
        outgoing_loads[i] = mean -
          neighbor_loads[i];
    end
end
return outgoing_loads;
```

3.2.1 Lesser Mean Assignment

Lesser mean assignment (LMA), detailed in Algorithm 2 and illustrated in Figure 1, attempts to equalize the workload of the local process and its lesser loaded neighbors. Upon the transmission of neighbor workloads, the mean of the local process and any neighbor with less load is computed. The mean is revised iteratively, removing any neighbors contributing to it which have more workload than it. Finally, we transmit the difference of each contributing neighbor from the mean to the corresponding processes.

The method could be interpreted as a unidirectional flow of workload from more to less loaded processes. In contrast to the original load balancing approach by Cybenko [Cyb89], which fixes the diffusion parameter to $1 - \frac{2}{dimensions+1}$ of the difference from each neighbor-



(a) (b)

(c) (d)

Figure 1: An illustration of particles (loads) assigned to a grid of processes. The lesser mean assignment balances the state (a) to state (b). Certain cases in which a lesser loaded process is surrounded by greater loaded processes such as (c) may lead to over-balancing as seen in (d).

ing process, the method scans the whole neighborhood for an optimal local distribution.

Due to unconditional flow of load from greater loaded neighbors, the lesser loaded processes potentially suffer from an issue of *over-balancing*, in which they receive a large sum of particles when surrounded by greater loaded neighbors. Since second-neighbor information is not available, the greater loaded processes are unable to take the neighborhood of the lesser loaded processes into account. The issue potentially results in cases in which the global maximum load increases, as seen in the second row of Figure 1.

3.2.2 Greater-Limited Lesser Mean Assignment

Greater-limited lesser mean assignment (GL-LMA) is detailed in Algorithm 3 and illustrated in Figure 2. It introduces a preprocessing step to LMA, utilizing the higher loaded neighbor information in an effort to prevent the over-balancing effect.

The processes start by computing the mean of their own load and any neighbor with *greater* load. The mean is revised iteratively, omitting neighbors with lesser load, ensuring that the local process is the only one with less load than it. The processes then define their total quota, that is the total number of particles they would accept from their neighbors, as the difference between the greater mean and their local load. Each neighbor is assigned a part of this quota, proportional to their contribution to the greater mean.

Algorithm 3: The greater limited lesser mean assignment.

```
function GLLMA ($load, neighbor\_loads$);
Input  : Local workload amount $load$, and vector
         of neighbors' workload amounts
         $neighbor\_loads$.
Output : Vector of workload amounts to be sent to
         each neighbor $outgoing\_loads$.
$contributors$ =
  vector<bool>($neighbor\_loads$.size());
$mean = load$;
do
    $contributors$.fill(false);
    $sum = load$;
    $count$ = 1;
    for $i$=0; $i$ < $neighbor\_loads$.size(); $i$++ do
        if $neighbor\_loads[i] > mean$ then
            $contributors[i]$ = true;
            $sum$ += $neighbor\_loads[i]$;
            $count$++;
        end
    end
    $mean = sum / count$;
while There are neighbors below mean in
  $contributors$;

$total\_quota = mean - load$;
$quotas$ = vector<uint>($neighbor\_loads$.size());
$quotas$.fill(0);
for $i$=0; $i$ < $neighbor\_loads$.size(); $i$++ do
    if $contributors[i]$ == true then
        $quotas[i] = total\_quota \cdot$
          $neighbor\_loads[i]/(sum - load)$;
    end
end
neighborhood_all_gather($quotas$);

$outgoing\_loads = LMA(load, neighbor\_loads)$;
return $pairwise\_minimum(outgoing\_loads,$
  $quotas)$;
```

The quotas are transmitted to the associated neighbors, through a local collective operation that we factor in as *neighborhood_all_gather* in Algorithm 3. This function may be realized using *MPI_Neighbor_allgather* or implemented manually as a series of local communications if MPI 3.0 is not available. Upon the transmission of the quotas, the processes apply a round of LMA, but limit the amount of outgoing particles by the quotas received from the neighbors.

The approach requires a third local collective operation in addition to the load and particle transmission stages, yet effectively prevents the *over-balancing* effect LMA is prone to. In an overview, each process first sets quo-

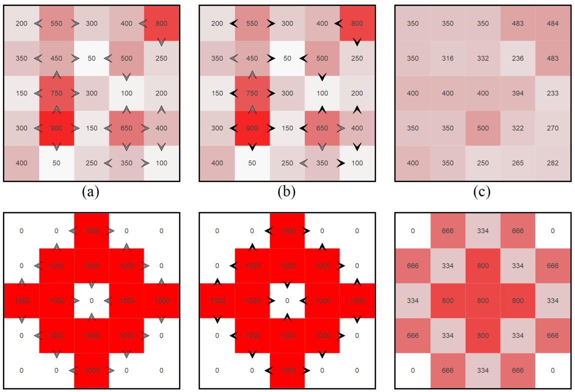


Figure 2: The greater-limited lesser mean assignment first declares quotas for the greater loaded neighbors, limiting the flow as in (a, d), followed by lesser mean assignment, balancing from (b, e) to (c, f). Cases that lead to over-balancing are averted by the initial quota process as seen in the second line, still providing a 20% boost to round run-time.

tas for its greater loaded neighbors, and then sends particles to its lesser loaded neighbors. The approach indirectly makes second neighbor information available to the processes, since they are now partially aware of the neighborhood of their lesser loaded neighbors through the quota received from them.

4 PERFORMANCE

This section details the experiments we have conducted to measure the runtime performance and scaling of the presented load balancing algorithms. We first enumerate and describe the variables of the particle advection pipeline in Section 4.1, and then construct a series of experiments by systematically varying them in Section 4.2. The metrics accompanying the measurements are detailed in Section 4.3.

4.1 Variables

4.1.1 Number of Processes

The number of processes is an axis of measurement, and is defined in terms of compute nodes rather than cores in this work. The process count also controls the number of partitions in a data-parallel sense; each process is responsible for one partition in the baseline approach, and for seven partitions (in 3D) in the diffusive load balancing approaches. Increasing the number of nodes refines the domain partitioning, which accelerates the compute.

4.1.2 Load Balancing Algorithm

A central variable is the load balancing algorithm. The pipeline supports four methods which are no diffusion, constant diffusion, lesser mean assignment and greater-limited lesser mean assignment. Each experiment is performed on all four methods regardless of measurement context, in order to quantify the benefits and limitations of the presented algorithms with respect to the baseline.

4.1.3 Dataset

The dataset is a variable since the complexity of the input vector field impacts the performance of particle advection [PCG+09]. We have obtained three datasets covering the domains of astrophysics, thermal hydraulics and nuclear fusion from the Research Group on Computing and Data Understanding at eXtreme Scale at the University of Oregon. Each dataset contains features which lead to distinct particle behavior as seen in Figure 3. The datasets are identical to the ones presented in [BPNC19], which we refer the reader to for further detail.

4.1.4 Dataset Size

The size of the dataset is a separate variable. It is in a linear relationship with the size of the partitions and hence influences performance. We scale each dataset to 1024^3, 1536^3, 2048^3 voxels, leading to 12.8GB, 43.4GB and 103GB floating-point triplets respectively.

4.1.5 Seed Set Distribution

The seed set distribution is another variable, as it has a direct impact on load imbalance. The particles are uniformly generated within an axis-aligned bounding box (AABB) located at the center of the dataset. Scaling the AABB enables concentrating the particles towards the center and vice versa. We limit the distributions to uniform scaling by 0.25, 0.5 and 1.0 (the complete dataset).

4.1.6 Seed Set Size

The seed set size is the final variable, which corresponds to the amount of particles/work. It is controlled through *stride*, a vector parameter describing the distance between adjacent particles in each dimension. A stride of $[1,1,1]$ implies one particle per voxel. We limit the strides to $[8,8,8]$, $[8,8,4]$, $[8,4,4]$, $[4,4,4]$, a sequence which may be interpreted as cumulatively doubling the amount of work across Z, Y and X.

The rest of the variables are fixed for all experiments: The integrator is set to Runge-Kutta 4, with a constant step size of 0.001. Maximum iterations per particle is set to 1000. Note that these settings are independent of the dataset size as long as the domain is scaled to the $[0,1]$ range in each dimension. Despite being adjustable, particles-per-round are set to 10 million in order to reduce benchmark combinations.

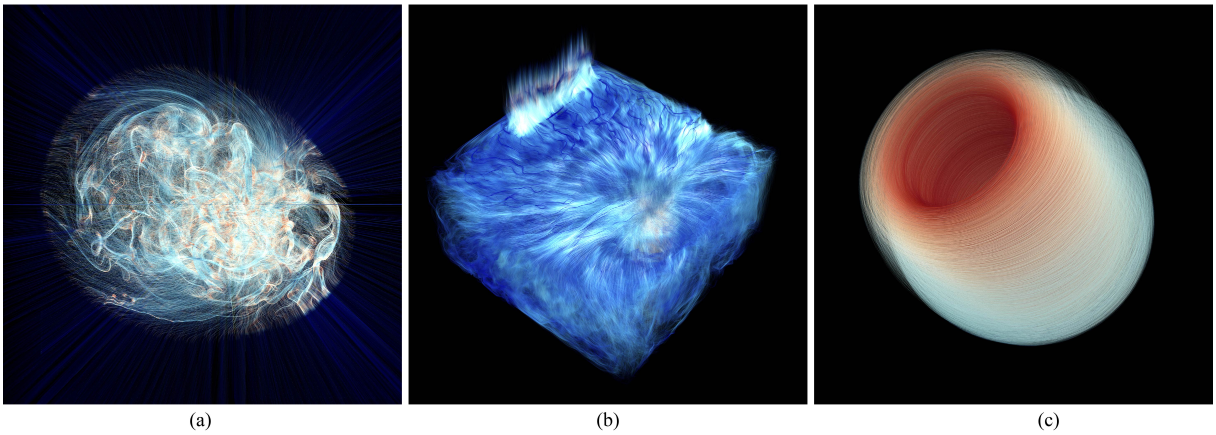


Figure 3: The astrophysics dataset (a) is a simulated magnetic field around a core-collapse supernova, with complex trajectories concentrated near the center. The thermal hydraulics dataset (b) is a simulation featuring two sources pumping water at different temperatures into a rigid box. The nuclear fusion dataset (c) is a simulation of the magnetic field within a Tokamak device, consisting of many orbital trajectories. All datasets comprise several GBs, streamlines are extracted by our approach and ray traced via [WJA^{+}17]. Color indicates vector magnitude.

4.2 Experiments

We assess the strong and weak scaling of the system, applying each algorithm to each dataset. Furthermore, we record and present the per-round load imbalance of the algorithms in a fixed setting. We finally scan the parameter space, assessing the impact of each variable in isolation.

4.2.1 Strong Scaling

The number of processes is varied among 16, 32, 64, 128. Every other variable is fixed: The dataset sizes are set to 1024^3, the seed set distribution spans the whole domain (uniform scaling of the AABB by 1.0) and the stride is $[4,4,4]$, leading to 16.78 million particles. The experiment is performed independently for each algorithm and dataset, leading to $nodes \cdot algorithms \cdot datasets = 4 \cdot 4 \cdot 3 = 48$ measurements. We record the total duration of particle advection for each process.

4.2.2 Weak Scaling

The number of processes is varied among 16, 32, 64, 128 simultaneously with stride, which is varied among $[8,8,8]$, $[8,8,4]$, $[8,4,4]$, $[4,4,4]$. Both the number of processes and the amount of work is doubled, as the stride leads to 2^{21}, 2^{22}, 2^{23} and 2^{24} particles respectively. The rest of the variables are fixed, identical to the strong scaling benchmarks: The dataset sizes are 1024^3 and the particle distribution spans the whole dataset. The experiment is performed independently for each algorithm and dataset, leading to $nodes \cdot algorithms \cdot datasets = 4 \cdot 4 \cdot 3 = 48$ additional measurements. We record the total duration of particle advection for each process.

4.2.3 Load Balancing

The number of processes is fixed to 16. The dataset sizes are set to 1024^3. The seed set distribution contains half of the domain in each dimension (uniform scaling of the AABB by 0.5) and the stride is set to $[4,4,4]$, yielding 16.78 million particles. The particles are concentrated near the center of the dataset, leading to significant load imbalance from the start. The experiment is performed independently for each algorithm and dataset, leading to $algorithms \cdot datasets = 4 \cdot 3 = 12$ measurements. We record the duration of each stage of each round per-process.

4.2.4 Parameter Space

The number of processes is varied among 16, 32, 64, 128 in addition to one of: dataset complexity, dataset size, seed distribution, seed stride. Each of the latter variables have at least three settings described in Section 4.1, which are tested in isolation. Note that when varying dataset size, we proportionally vary stride in order to keep the particle count constant. When fixed, the dataset is set to astrophysics, the dataset size is set to 1024^3, the seed distribution is set to 1.0 and the seed stride is set to $[4,4,4]$. The experiment is performed for each algorithm, leading to $nodes \cdot algorithms \cdot (variables \cdot variable\_options) = 4 \cdot 4 \cdot (4 \cdot 3) = 192$ measurements. We record the total duration of particle advection for each process. Note that the dataset complexity configurations are equivalent to the strong scaling tests, yet are recorded separately and included for comparison to other parameters.

4.3 Metrics

For each scaling measurement we compute the speedup from the time measurements. The speedup is always

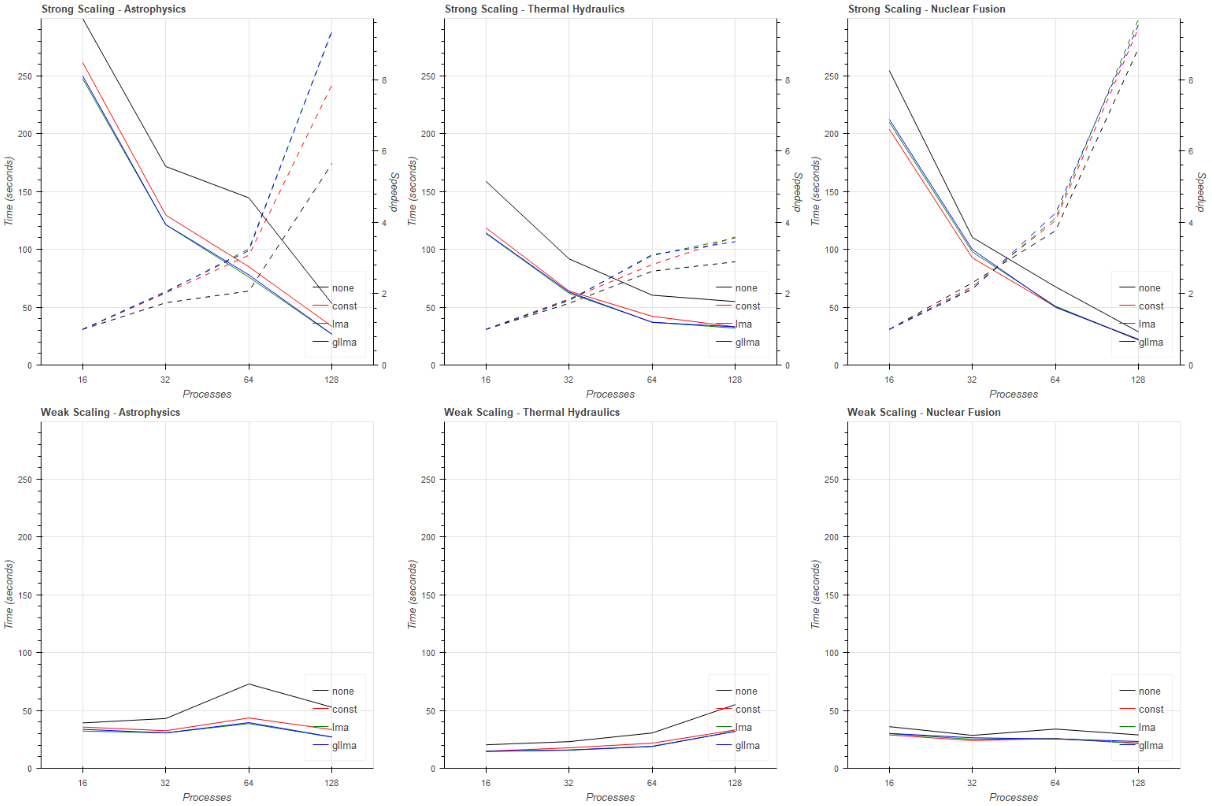


Figure 4: Strong (top) and weak (bottom) scaling benchmarks. Black lines are no diffusion, red lines are constant diffusion, green lines are LMA, blue lines are GL-LMA. Datasets vary from left to right; astrophysics, thermal hydraulics, nuclear fusion. Solid lines are total advection times, dashed lines are speedups.

defined in terms of nodes, rather than cores. It is furthermore relative to the first measurement, since the problem sizes often exceed the capabilities of the serial application. That is, for the strong scaling benchmarks ran on $N_i < ... < N_j$ nodes, speedup is defined as:

$$S(N) = \frac{T(N_i)}{T(N)}$$

where $T(N)$ is the duration of the application with N nodes.

For the load balancing measurements, we compute the load imbalance factor in each round as:

$$LIF = \frac{load_{max}}{load_{avg}}$$

which is equal to 1 when all processes have an equal amount of work. The metric denotes the distance of the current load distribution from the optimal state where each process has exactly average workload.

5 RESULTS & DISCUSSION

This section presents the results accompanied by a comparative discussion on the performance of the four algorithms under varying conditions.

The performance benchmarks are ran on the RWTH Aachen CLAIX-2018 (c18m) compute cluster, which offers up to 1024 nodes containing 2x24 Intel Skylake Platinum 8160 cores, along with 192GB of memory per node.

The tests are conducted using 16 to 128 nodes. The nodes are exclusively reserved for the application, in order to eliminate any effects due to resource consumption of other processes. Each node is configured to run one instance of the MPI application, which saturates all 48 cores and the complete memory. The number of cores and memory per node are fixed throughout the tests.

5.1 Strong Scaling

The strong scaling benchmarks are presented as line plots mapping node counts to total advection times in the first row of Figure 4. The plots show asymptotic behavior, with the diffusive approaches consistently outperforming the baseline. The LMA and GL-LMA display nearly identical performance implying that overbalancing does not occur. Both approaches outperform constant diffusion in 10 out of 12 cases. The exceptions are the 16 and 32 node runs on the nuclear fusion dataset, which we attribute to the coarseness of domain

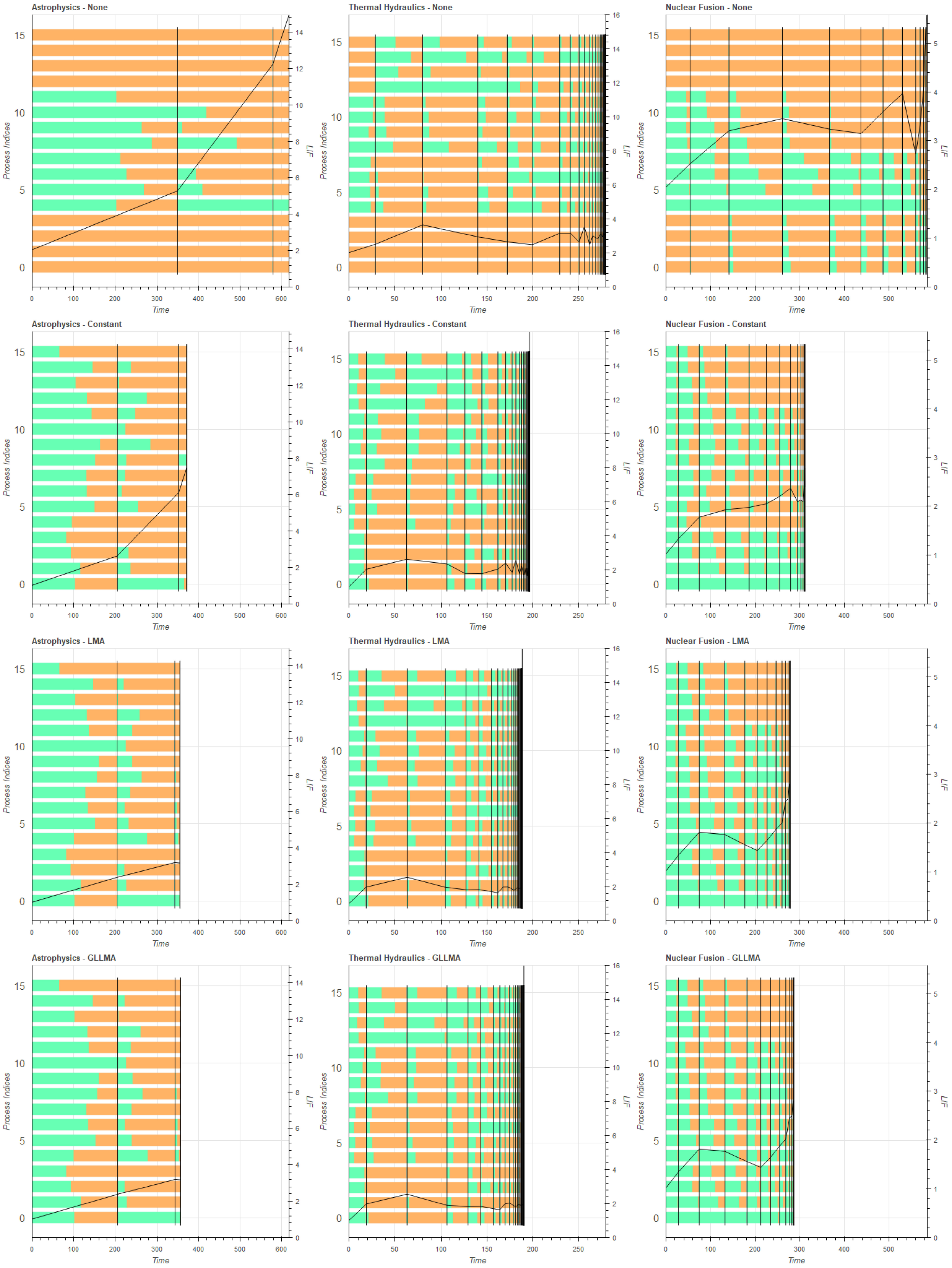


Figure 5: Gantt charts of round times per process. Green is active compute, orange is idle. Datasets vary from left to right; astrophysics, thermal hydraulics, nuclear fusion. Load balancing algorithms vary from top to bottom; no diffusion, constant diffusion, LMA, GL-LMA. Load imbalance factors are overlaid as line plots.

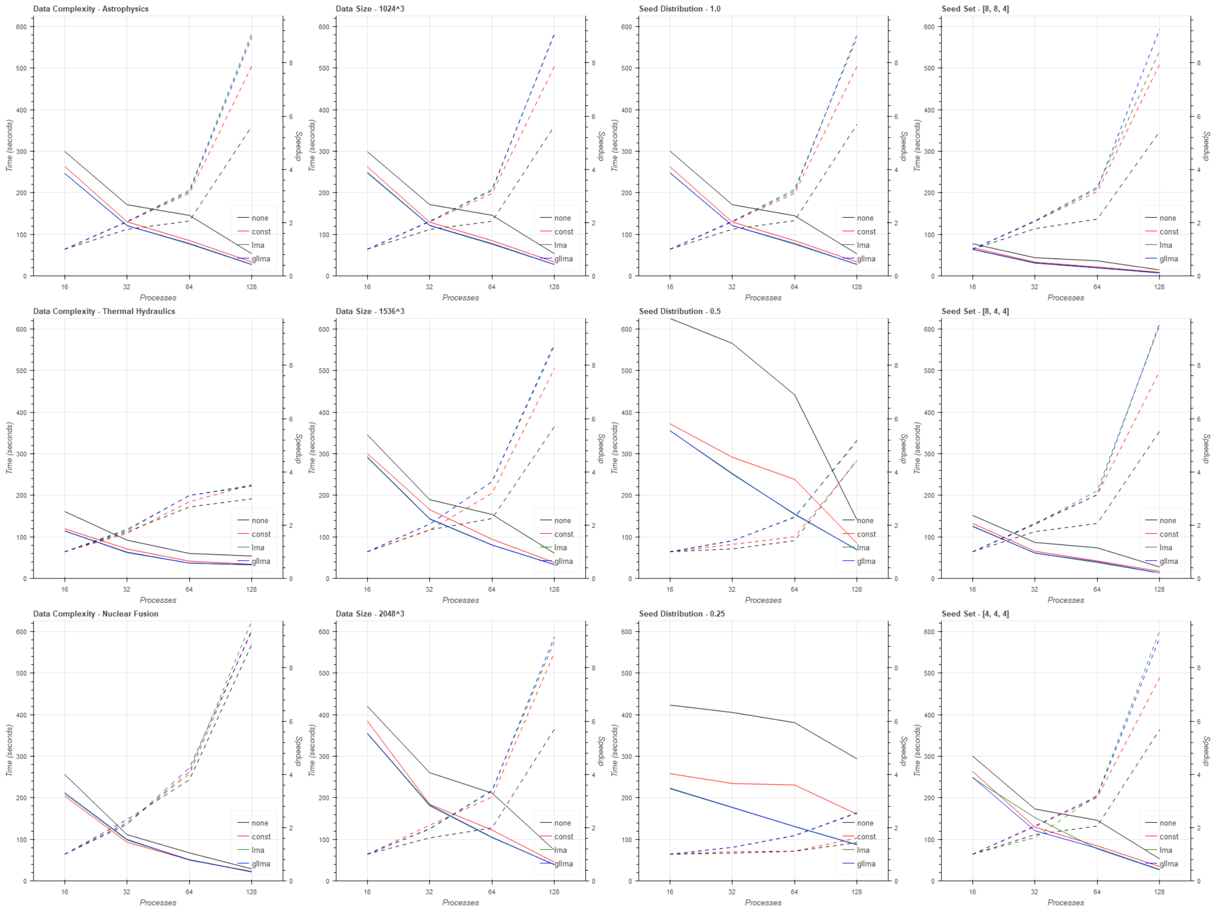


Figure 6: Parameter space benchmarks. From left to right; dataset type, dataset size, seed distribution, seed set size. Black lines are no diffusion, red lines are constant diffusion, green lines are LMA, blue lines are GL-LMA. Solid lines are total advection times, dashed lines are speedups.

partitioning at low node counts. The total advection times converge with increasing node counts.

The baseline yields a relative speedup of 2.07 for the 64 node run on the astrophysics dataset, significantly lower than the diffusive approaches which provide a minimum of 3.11. Irregularities regarding this configuration also appear in the latter tests, which may indicate a relationship to the domain partitioning for the given node count.

5.2 Weak Scaling

The weak scaling benchmarks map node count - stride pairs to total advection times as presented in the second row of Figure 4. The astrophysics set leads to non-constant behavior with the 64 node configuration, although this effect is largely suppressed by the diffusive load balancing approaches.

The thermal hydraulics dataset displays an increase in total runtime with increasing nodes and work, becoming apparent in the 128 node configuration. The dataset penalties static domain partitioning, as it contains long trajectories covering the whole domain. Refining the domain subdivision leads to increased communication across nodes, which negatively affect runtime, a pattern which is also seen in Figure 5. The nuclear fusion dataset displays constant behavior.

5.3 Load Balancing

The load balancing benchmarks are presented as Gantt charts mapping per-stage advection times to the nodes in Figure 5. Note that due to the seed distribution being set to the center 0.5 of the dataset, the outer processes are expected to initially idle. The GL-LMA nearly halves the total time advection in the astrophysics dataset, bringing 617 seconds down to 356. Similar results are observed for the thermal hydraulics and nuclear fusion datasets.

The LMA outperforms GL-LMA in cases where many short rounds are involved. The per-round overhead of the latter, along with the limits it applies, leads to longer total advection times.

5.4 Parameter Space

The parameter space experiments are presented as line plots in Figure 6 and are in agreement with the strong scaling benchmarks. The LMA and GL-LMA achieve the lowest advection time in 46 of the 48 measurements, along with the highest speedups for all 16 configurations.

The astrophysics and nuclear fusion datasets appear to greatly benefit from increased process counts, yielding relative speedups of 8.9 and 9.37 for the 128 node setting whereas this is limited to 3.45 for thermal hydraulics. The dataset contains empty regions above and below the box, which leads to uneven partitioning in the associated axis.

Changes to dataset and seed set size display a constant response to total advection time, yet seed distribution appears to have a great effect as seen in the third column. LMA and GL-LMA are more resilient to increase in locality, compared to the baseline and constant case. Overbalancing occurs in the 32 node setting of the densest seed set distribution, yet is averted by GL-LMA as seen in the fourth column.

6 CONCLUSION

We have presented a distributed particle advection system along with a local load balancing method based on partial data replication among process neighborhoods. Two local scheduling methods based on diffusive load balancing have been applied to the problem of parallel particle advection. The performance has been assessed through strong and weak scaling benchmarks as well as load imbalance metrics. The results are shown to improve total runtime and are promising regarding scaling, since the method exclusively operates on local neighborhoods of processes, avoiding any global communication.

7 ACKNOWLEDGEMENTS

The authors gratefully acknowledge the computing time granted by the NHR4CES Resource Allocation Board and provided on the supercomputer CLAIX at RWTH Aachen University as part of the NHR4CES infrastructure. The calculations for this research were conducted with computing resources under the project rwth0432.

8 REFERENCES

[Boi90] J. E. Boillat. Load balancing and poisson equation in a graph. *Concurrency: Pract. Exper.*, 2(4):289–313, November 1990.

[BPC21] R. Binyahib, D. Pugmire, and H. Childs. HyLiPoD: Parallel Particle Advection Via a Hybrid of Lifeline Scheduling and Parallelization-Over-Data. In Matthew Larsen and Filip Sadlo, editors, *Eurographics Symposium on Parallel Graphics and Visualization*. The Eurographics Association, 2021.

[BPNC19] R. Binyahib, D. Pugmire, B. Norris, and H. Childs. A lifeline-based approach for work requesting and parallel particle advection. In *2019 IEEE 9th Symposium on Large Data Analysis and Visualization (LDAV)*, pages 52–61, Oct 2019.

[Cyb89] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7(2):279–301, 1989.

[HY00] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D: Nonlinear Phenomena*, 147(3):352–370, 2000.

[MCHG13] C. Müller, D. Camp, B. Hentschel, and C. Garth. Distributed parallel particle advection using work requesting. In *2013 IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV)*, pages 1–6, Oct 2013.

[PCG$^+$09] D. Pugmire, H. Childs, C. Garth, S. Ahern, and G. H. Weber. Scalable computation of streamlines on very large datasets. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, pages 1–12, Nov 2009.

[PRN$^+$11] T. Peterka, R. Ross, B. Nouanesengsy, T. Lee, H. Shen, W. Kendall, and J. Huang. A study of parallel particle tracing for steady-state and time-varying flow fields. In *2011 IEEE International Parallel Distributed Processing Symposium*, pages 580–591, May 2011.

[WJA$^+$17] I. Wald, G. Johnson, J. Amstutz, C. Brownlee, A. Knoll, J. Jeffers, J. Günther, and P. Navratil. Ospray - a cpu ray tracing framework for scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):931–940, Jan 2017.

[ZGH$^+$18] J. Zhang, H. Guo, F. Hong, X. Yuan, and T. Peterka. Dynamic load balancing based on constrained k-d tree decomposition for parallel particle tracing. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):954–963, Jan 2018.

[ZGYP18] J. Zhang, H. Guo, X. Yuan, and T. Peterka. Dynamic data repartitioning for load-balanced parallel particle tracing. In *2018 IEEE Pacific Visualization Symposium (PacificVis)*, pages 86–95, April 2018.

An improved simple feature set for face presentation attack detection

Anna Denisova[1,2]

[1]Samara National Research University
Moskovskoye shosse 34,
443086, Samara, Russia
[2]Image Processing Systems Institute – Branch of the Federal Scientific Research Centre
“Crystallography and Photonics” of
Russian Academy of Sciences
Molodogvardeiskaya st. 151
443001, Samara, Russia

denisova_ay@geosamara.ru

ABSTRACT

Presentation attacks are weak points of facial biometrical authentication systems. Although several presentation attack detection methods were developed, the best of them require a sufficient amount of training data and rely on computationally intensive deep learning based features. Thus, most of them have difficulties with adaptation to new types of presentation attacks or new cameras. In this paper, we introduce a method for face presentation attack detection with low requirements for training data and high efficiency for a wide range of spoofing attacks. The method includes feature extraction and binary classification stages. We use a combination of simple statistical and texture features and describe the experimental results of feature adjustment and selection. We validate the proposed method using WMCA dataset. The experiments showed that the proposed features decrease the average classification error in comparison with the RDWT-Haralick-SVM method and demonstrate the best performance among non-CNN-based methods.

Keywords

Presentation attack detection, feature extraction, depth map, thermal data, infrared data, WMCA, SVM, RDWT-Haralick-SVM, MC-CNN.

1. INTRODUCTION

Recent research in face recognition systems vulnerability revealed high Impostor Attack Presentation Match Rates (IAPMR) for almost all kinds of face recognition systems [Bha19]. This fact demonstrates the crucial need of the development of presentation attack detection (PAD) mechanisms to protect face recognition systems. The general scenario of presentation attack is to demonstrate a fake image of the bona fide person to the authentication system. Presentation attack instruments may include printed photos, masks or screen photos that can be easily reproduced due to high accessibility of the target person’s photos in social networks and other open data sources.

To prevent face recognition systems from presentation attacks five basic approaches were developed. The motion based approach exploits an additional analysis of face movements in the video sequence to extract liveness information [Anj11; Fre12; Liu09]. For example, the eye blinking in the video distinguishes the real person’s image from printed photos. The motion based PAD methods suffer from long time of data registration and require in most cases additional actions from the user. The second approach is texture based methods [Pen18; Pen20; Du21]. Texture based methods suppose that the real face image has significantly different textural properties from the fake image reproduced using printing or screen devices. However, this kind of method may not be effective in the case of more complex attacks such as 3D masking. The image quality based methods belong to the third approach that calculates different quality measures for fake and real images [Gal14; Wen15]. Usually low cost spoofing devices lead to image quality degradation that is monitored by this group of methods. Nevertheless, high quality spoofing cannot be detected by these methods. The fourth approach is

https://www.doi.org/10.24132/CSRN.3201.3

multimodal spoofing detection. The main modality used for PAD is RGB color images. The other modalities such as infrared images, depth maps or thermal images provide useful additional information for liveness detection and can improve the PAD detection for many kinds of sophisticated attacks as well as for the novel PAD instruments [Wan20; Abd21; Kow20]. For example, 3D-latex and silicone masks can be easily detected using thermal data due to the difference in thermal characteristics of the artificial materials in comparison to the real faces [Kow20]. The last approach, that is worth mentioning, is deep learning. Deep learning algorithms are based on artificial neural networks and use tones of images for training. They can be used in both multimodal and single modal cases. At the current time, deep learning methods provide the best quality of spoofing detection [Wan19; Geo19; Bre19]. However, the need for large training datasets makes these methods inconvenient for launching on new devices when it is impossible to collect a lot of training data.

In this article, we propose a multimodal PAD method based on simple handcrafted features that can be easily calculated for new imaging devices. The features obtained for each particular sensor are concatenated and then processed together by the classifier. The method handles the proposed features using one of the classical binary classification methods such as linear support vector machines classifier (linear SVM) and random forest classifier (RF). These classification methods allow training with small training sets. Thereby the method combines the universality of feature extraction for new modalities and the simplicity of training provided by classical binary classification approaches.

We compared our method with two algorithms such as RDWT-Haralick-SVM and MC-CNN which have shown the best spoofing detection performance according to [Geo19]. The RDWT-Haralick-SVM algorithm leans on linear SVM classifier and texture features. MC-CNN is a deep learning algorithm that uses an artificial neural network for feature extraction and classification. Our method outperformed the RDWT-Haralick-SVM algorithm and gave closer results to MC-CNN. Although MC-CNN demonstrates higher performance, it uses an extremely large dataset for training whereas the proposed method can be trained only with hundreds of images. Thereby, the proposed algorithm is more suitable in cases when training images are hard to obtain.

The rest of the paper is organized as follows. Section 2 describes the proposed feature set. Section 3 provides the experimental results of feature set adjustment and comparison of the proposed method with the baseline algorithms such as RDWT-Haralick-SVM and MC-CNN.

2. PROPOSED METHOD

2.1 Feature Extraction

There are several input images $X, Y_1, Y_2, ..., Y_N$ of the target face. The image X is a grayscale image obtained by RGB camera and $Y_1, Y_2, ..., Y_N$ are the images obtained by different additional sensors (sensor images), for example, depth map or infrared image. All images are geometrically consistent and have the same size $I \times J$ pixels. Input images are linearly contrasted in the range $[0, 255]$. The features are calculated in two stages. Firstly, we calculate feature vector $f_n \in R^M$ for a pair of grayscale and sensor image X, Y_n. Finally, all feature vectors are concatenated $\left(f_1^T, f_2^T, ..., f_N^T\right)^T \in R^{NM}$.

There are two key ideas behind the feature extraction process. First, if there is no attack the images X and Y_n have to demonstrate the same face contours. Second, if there is an attack the images X and Y_n have to significantly differ from each other in both statistical and textural aspects. Using these ideas we constructed our feature set from 6 basic feature groups which were examined in our paper [Den21]. Even with these feature groups we obtained better spoofing detection. However, texture features are a very powerful instrument of image analysis and, in this paper, we add Haralick features as the seventh basic feature group. The basic feature groups are presented in Table 1. Following by our intuitions we consider some additional feature groups as well. These features are optional and may be included or omitted depending on experimental results. The list of additional feature groups is provided in Table 2.

The first basic feature group (Var) includes the variance of the sensor image. To compute the second basic feature group (AvgArr) we divide the sensor image on $K \times K$ regions and calculate the average brightness for each region.

The third basic group (GradYH) is a histogram bins $H_{Yn}(q), q = 1, ..., Q$ of the gradient amplitude G_{Yn} for sensor image. We take into account only values of the gradient amplitude from 0 to 30 and the number of bins is supposed to be $Q = 20$. This group of feature group allows us to distinguish between print and replay attacks when the sensor images do not have contours of the face and, therefore, demonstrate low gradient amplitude values.

The fourth basic group (CorrY) includes the coefficients of sensor image correlation. If $B_n(p_1, p_2)$ is an autocorrelation function of sensor image Y_n, the values of $B_n(p_1, p_2)$ for particular

p_1, p_2 are the features values. In this paper, we consider correlation in four directions $B_n(-1,0)$, $B_n(1,0)$, $B_n(0,1)$ and $B_n(0,-1)$. Usually, the correlation for fake images is higher than for bona fide images.

Notation	Images needed for computation	Number of features in group
Var	Y_n	1
AvgArr	Y_n	$K^2 = 9$
GradYH	Y_n	$Q = 20$
CorrY	Y_n	4
GradXH	X, Y_n	$Q = 20$
CorrXY	X, Y_n	1
H13	Y_n	208

Table 1. Basic feature groups

Notation	Images needed for computation	Number of features in group
StatY	Y_n	4
HistY	Y_n	$Q = 20$
GradDirYH	Y_n	$Q = 20$
GradDirXH	X, Y_n	$Q = 20$

Table 2. Additional feature groups

The fifth basic feature group (GradXH) is a histogram $H_{X(n)}(q), q = 1,...,Q$ of the gradient amplitude G_X of the grayscale image X in contour points of the sensor image Y_n. The contour points $i, j \in \Omega_{Y(n)}$ are the points detected by Canny detector in the image Y_n. The histogram $H_{X(n)}(q)$ is computed only for values $G_X(i,j), (i,j) \in \Omega_{Y(n)}$. We compute the histogram $H_{X(n)}(q)$ only for the range from 0 to 128 because these values of the gradient amplitude are more informative and correspond to local brightness variations within the face area. The number of bins is 20. These features aim to evaluate the correspondence between the facial contours on the sensor and grayscale images.

The sixth basic feature group (CorrXY) is the mutual correlation of the gradients G_X and $G_{Y(n)}$.

The seventh basic feature group (H13) is a concatenation of 13 Haralick features obtained for different parts of the sensor image. We divide the sensor image Y_n into $W \times W$ regions and compute 13 Haralick features for each region: Angular Second Moment, Correlation, Inverse Difference Moment, Sum Variance, Entropy, Difference Entropy, Contrast, Sum of Squares: Variance, Sum Average, Sum Entropy, Difference Variance, Info. Measure of Correlation 1 and Info. Measure of Correlation 2. The names of features correspond to the ones in the article [Har73]. In this paper we used *W=4*.

As for additional feature groups, they were designed intuitively to improve the classification performance. We added the features explaining the statistical properties of the sensor image and the gradient phase based features. The feature group StatY is the mean, the median, the range and the maximum value obtained for the sensor Y_n image. The feature group HistY is the histogram $H_{Y(n)}(q), q = 1,...,Q$ of the sensor image Y_n. For this histogram we use the full range of pixel values from 0 to 255 and $Q = 20$. The next feature group (GradDirYH) is a histogram $HDir_{Yn}(q), q = 1,...,Q$ of the gradient phase $GDir_{Yn}$ for the sensor image. The histogram range is from -180 degrees to 180 degrees. The number of bins remained the same $Q = 20$. The last feature group is the gradient phase histogram (GradDirXH) $HDir_{X(n)}(q), q = 1,...,Q$ for image X in contour points of the sensor image Y_n. This feature group is calculated in a similar way that is for GradXH feature group. The only difference is using the gradient phase instead of the amplitude to build the histogram. The range of the histogram bins varies from -180 degrees to 180 degrees. The number of bins is equal to 20.

The proposed features have several advantages. First of all, they are simple in computation. Secondly, they are enough effective even for images with degraded quality when the sensor image Y_n has a lower resolution than the grayscale image X.

2.2 Classification

We consider the presentation attack detection as a binary classification problem. The bona fide presentations are classified in class 1. The fake images are classified in class 0. To perform classification we use one of three algorithms: linear SVM (Support Vector Machines with linear kernel) [Chr00] and RF (Random Forest) [Kul12]. These algorithms are able to work with small training sets containing several hundreds of images.

To find the parameters of each algorithm we produced a grid search with optimization of the following standard measures: attack presentation classification error rate (APCER), bona fide presentation classification error rate (BPCER) and average classification error rate (ACER). The APCER corresponds to False Positive Rate. BPCER is the same as False Negative Rate. ACER is the average of APCER and BPCER.

For linear SVM, we tested penalty parameter C in the range [0,1] with step 0.0001 and the cost matrix A. The cost matrix was chosen as one of three matrixes:

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, A_2 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, A_3 = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix},$$

where the cost of bona fide class is 1, 2 and 4.

For the RF algorithm, we optimized the tree number N and cost matrix A. We tested N from 10 to 200 with the step 10.

The cost matrixes and algorithm parameters' ranges were selected during the preliminary experiments.

3. EXPERIMENTAL RESEARCH

3.1 Dataset

We evaluated our method using the Wide Multi-Channel Presentation Attack Database (WMCA) [Geo19]. WMCA includes 1679 videos which contain 347 bona fide presentations and 1332 attacks. The attacks are grouped in the following categories:

– "fake head" when the heated mannequin was used instead of real face;

– "print attack" when printed photo of the target person is demonstrated to the camera;

– "replay" when the screen with the person's portrait is demonstrated to the camera;

– "rigid mask" when the person wears the rigid mask made in a handcrafted manner from rigid materials;

– "flexible mask" when the person wears the soft silicon mask;

– "paper mask" when the person wears a handcrafted paper mask;

– "glasses" when the person wears paper or funny eyes glasses.

WMCA database incorporates data captured in four modalities: RGB color data, depth map, infrared data and thermal data. The first three modalities were captured using Intel RealSense SR300. Thermal data were collected using Seek Thermal Compact PRO. Each image in the dataset is presented in four channels color image intensity (C), depth (D), infrared image (I) and thermal data (T). The images are geometrically aligned and have 128 × 128 pixels in size. The examples of four image channels are presented in Figure 1.

In [Geo19] the authors used approximately 50 frames per video for data augmentation that is 83950 images in total. We will refer to this case as the full dataset. In our case, we do not need an augmentation to train classifiers that is why we used only one frame per video that is 1679 images in total. We will refer to this case as the basic dataset. Most of our experiments are provided for the basic dataset.

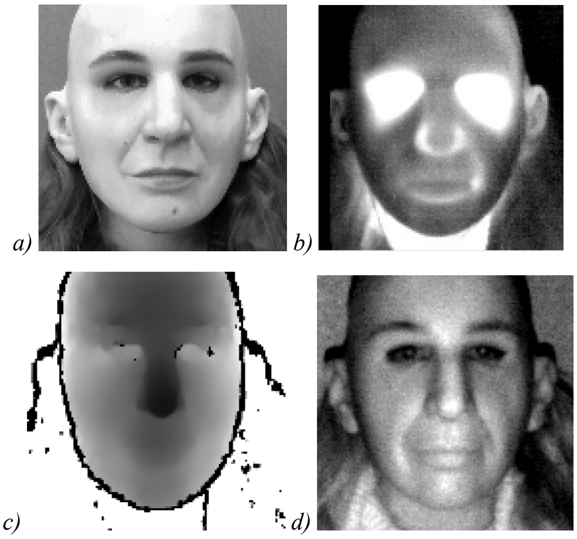


Figure 1. The example of data in CDIT channels for fake head attack: a) color image intensity, b) thermal data, c) depth, d) infrared

3.2 Baseline Methods

We focus on the two methods that demonstrated the best results in the paper [Geo19]. The first one is RDWT-Haralick-SVM method proposed in [Ewa20]. This method is based on RDWT-Haralick features and linear SVM classification. To compare our method with RDWT-Haralick-SVM in the same conditions we reimplemented this method. The final feature vector in our implementation is a concatenation of RDWT-Haralick features obtained for each image channel separately. The RDWT-Haralick features for one image channel are obtained in the same way as described in [Geo19].

The second baseline method is MC-CNN developed in [Geo19]. This method is based on LightCNN deep learning model proposed in [Wu18]. The MC-CNN extends pre-trained LightCNN for four channels and applies additional training to some layers of the model. A detailed description of the MC-CNN architecture is given in [Geo19].

3.3 Experiment protocol

In this paper, we follow the grandtest protocol described in [Geo19]. We divide datasets into training, test and validation (development) sets in the same manner. Thus, training, test and validation sets do not have common bona fide presentations and demonstrate the almost equal distribution of presentation attack instruments.

We provide two series of experiments. In the first series, we evaluate only the basic feature set and define the basic performance of the method in different conditions. In the second series, we test our intuitions on further performance improvement using additional features and adjusting some basic features.

3.4 Channel Selection

In the first experiment, we assessed the optimal channel set. We tested the algorithm using basic dataset. The results are shown in Table 3. The best performance is demonstrated for all channels (CDIT) and corresponds to an ACER value equal to 2.91%. It is less than the ACER=3.44% obtained by RDWT-Haralick-SVM on the full dataset. Thus, the proposed basic feature set improves the classification results obtained for CDIT channels even for small training sets. Further, we provide the results only for CDIT channel configuration.

3.5 Classifier selection

To investigate classifiers' performance and compare the results with the baseline methods we carried out experiments with the basic and full datasets. The results for the basic dataset are shown in Table 4. The proposed method demonstrates the best results for linear SVM classifier (ACER=2.91%). The RDWT-Haralick-SVM achieves ACER bigger in 1.43 times. Therefore, the proposed method in combination with linear SVM improves spoofing detection for the small training sets in comparison to RDWT-Haralick-SVM.

The classification results for the full dataset are given in Table 5. The proposed algorithm achieves the best result with ACER=1.18% using linear SVM and it still outperforms RDWT-Haralick-SVM (ACER=3.44%) but loses to MC-CNN (ACER=0.3%). Nevertheless, we suppose that MC-CNN is impossible to train in the case of small datasets including hundreds of images whereas the proposed algorithm is successfully trained in this case.

3.6 Additional Feature Set Testing and Basic Feature Set Parameters Adjustment

The second series of experiments aimed to check the following our intuitions about feature improvement:

1) RGB data and gradient phase can improve the classification results. These two heuristic scenarios correspond to the cases when RGB channels are considered as additional independent modalities and when the gradient phase histograms are taken into account as additional features GradDirYH and GradDirXH (see Table 2).

Channels	Optimal hyperparameters		Validation set		Test set		
	C	A	APCER, %	BPCER, %	APCER, %	BPCER, %	ACER, %
CDIT	**0.0055**	**A_3**	**3.34**	**1.85**	**4.07**	**1.74**	**2.91**
CDT	0.0119	A_3	4.90	0	7.01	6.96	6.99
CTI	0.0111	A_3	4.45	0.93	5.20	1.74	3.47
CDI	0.0015	A_3	11.58	7.41	9.28	0	4.64
CT	0.0439	A_3	7.13	0	10.86	1.74	6.30
CD	0.0319	A_3	10.02	13.89	12.90	6.09	9.49
CI	0.0856	A_3	6.46	3.70	6.11	8.70	7.40

Table 3. Classification errors for the test set for different input channels (basic dataset, linear SVM)

Method	Optimal hyperparameters	APCER, %	BPCER, %	ACER, %
Basic feature set + SVM	**C = 0.0055, A_3**	**4.07**	**1.74**	**2.91**
Basic feature set + RF	N=80, A_2	5.88	7.83	6.85
RDWT-Haralick+SVM	C=0.0002, A_3	8.37	0	4.18

Table 4. Classification errors for the test set in the case of CDIT data and basic dataset of 1679 images

Method	Optimal hyperparameters	APCER, %	BPCER, %	ACER, %
Basic feature set + SVM	C = 1.1327, A_3	0.11	2.24	1.18
Basic feature set + RF	N=80, A_2	3.23	6.50	4.87
RDWT-Haralick+SVM [Geo19]	-	6.39	0.49	3.44
MC-CNN [Geo19**]**	**-**	**0.60**	**0.0**	**0.30**

Table 5. The classification errors for the test set in the case of CDIT data and full dataset of 83950 images

2) the optimal number of regions used for AvgArr feature calculation can be selected. The feature AvgArr is composed of statistics obtained for $K \times K$ regions of the image. However, the preliminary test of the algorithm included only results for $K = 3$ while the other values of K may provide better results.

3) face region masking can improve the results. The face region and the background probably contain different information about spoofing methods. Thus, it is possibly better to calculate features in these regions separately.

We checked these three hypotheses for basic and extended feature sets. The last one includes basic features (see Table 1) and two additional features HistY and StatY (see Table 2). The experiments were carried out using basic dataset (1679 images) and linear SVM classifier.

Table 6 shows the results of our first intuition. As we can see from Table 6, the RGB channels do not improve the classification results. On the contrary, the gradient phase histograms improved the average classification error by almost 1%. Besides the extended feature set demonstrated a little bit better result (ACER=2.04%) than the basic feature set (ACER=2.37).

Table 7 demonstrates the results for the second intuition. The experiments showed that the optimal value of K is 8. Moreover, the extended feature set demonstrated a significant decrease in ACER (1.69%) for optimal K.

Table 8 demonstrates the results for different ways of masking. The separation of an image on the face and non-face regions achieves the best classification error. Besides, the result obtained for the extended feature set (ACER=1.70%) is better than the one for the basic feature set (ACER=2.36%).

To summarize, we defined three strategies for feature improvement. The first strategy is the use of gradient phase histograms. The second strategy is the use of $K = 8$ for AvgArr computation. And the last one is to separate feature extraction for face and non-face regions by means of masking.

Features	Channels	Optimal hyperparameters	ACER%	APCER%	BPCER%
Basic	CDIT	C= 0.0055, A_3	2.91	4.07	1.74
Extended	CDIT	C= 0.0032, A_3	3.00	3.39	2.61
Basic	CDIT+RGB	C= 0.0060, A3	4.26	6.79	1.74
Extended	CDIT+RGB	C= 0.0026, A3	3.81	5.88	1.74
Basic + GradDirYH + GradDirXH	CDIT	C = 0.0218, A3	2.37	4.75	0
Extended + GradDirYH + GradDirXH	CDIT	C = 0.0019, A3	2.04	4.07	0

Table 6. Classification errors for the test set and for the first hypothesis (linear SVM, basic dataset)

Features	K	Channels	Optimal hyperparameters	ACER%	APCER%	BPCER%
Basic	2	CDIT	C=0.0058, A3	3.88	4.30	3.48
Basic	3	CDIT	C=0.0055, A_3	2.91	4.07	1.74
Basic	4	CDIT	C=0.0057, A3	3.66	3.85	3.48
Basic	**8**	**CDIT**	**C=0.0178, A3**	**2.24**	**3.62**	**0.87**
Extended	2	CDIT	C=0.0035, A3	3.00	3.39	2.61
Extended	3	CDIT	C=0.0032, A_3	3.00	3.39	2.61
Extended	4	CDIT	C=0.0035, A3	3.00	3.39	2.61
Extended	**8**	**CDIT**	**C=0.0036, A3**	**1.69**	**3.39**	**0**

Table 7. Classification errors for the test set and for the second hypothesis (linear SVM, basic dataset)

Features	Mask	Optimal hyperparameters	ACER%	APCER%	BPCER%
Extended	One elliptical face-mask	C=0.0046, A_3	2.92	4.98	0.87
Extended	**Two masks for face and non-face regions**	**C=0.0020, A3**	**1.70**	**3.39**	**0**
Extended	No mask	C=0.0032, A3	3.00	3.39	2.61
Basic	One elliptical face-mask	C=0.0035, A3	2.38	4.75	0
Basic	**Two masks for face and non-face regions**	**C=0.0050, A3**	**2.36**	**3.85**	**0.87**
Basic	No mask	C=0.0055, A3	2.91	4.07	1.74

Table 8. Classification errors for the test set and for third hypothesis (linear SVM, basic dataset)

Features	Gradient phase histograms	AvgArr, K	Optimal hyperparameters	ACER%	APCER%	BPCER%
Basic	+	**8**	**C=0.0121, A_3**	**2.26**	**4.52**	**0**
Extended	+	8	C=0.0054, A_3	1.81	3.62	0
Basic	-	3	C=0.0055, A_3	2.91	4.07	1.74
Extended	-	3	C=0.0032, A_3	3.00	3.39	2.61
Basic	+	3	C=0.0218, A3	2.37	4.75	0
Extended	+	3	C=0.0019, A3	2.04	4.07	0
Basic	-	8	C=0.0178, A3	2.24	3.62	0.87
Extended	**-**	**8**	**C=0.0036, A3**	**1.69**	**3.39**	**0**

Table 9. Classification errors for the test set in the case of first and second feature improvement strategies (CDIT channels, basic dataset of 1679 images)

Features	Gradient phase histograms	AvgArr, K	Masking	Optimal hyperparameters	ACER%	APCER%	BPCER%
Basic	+	8	Two masks for face and non-face regions	C=0.0025, A3	1.69	3.39	0
Extended	-	8	Two masks for face and non-face regions	C=0.0019, A3	1.58	3.17	0

Table 10. Classification errors for the test set in the case of best feature improvement strategies (CDIT channels, basic dataset of 1679 images)

Table 9 provides the results for the simultaneous use of first two strategies. For extended feature set, the best choice is to use the second strategy only. For basic dataset, the best choice is using both first and second strategies simultaneously.

The further evaluation of the best strategies is given in Table 10. In both cases, simultaneous use with the third strategy provides the decrease in classification error.

In the end, we may conclude that all of the explored intuitions lead us to classification performance improvement. The best strategies shown in Table 10 demonstrate a decrease of classification error almost 1.84 times in comparison to the basic feature set with the default parameters.

4. CONCLUSION

In this paper, we propose a method for multimodal face presentation attack detection. The method is based on simple statistical and texture feature extraction and classical binary classification methods. The main advantages of the proposed method are the universality in terms of sensors used to obtain data and the simplicity of training on small training sets. Therefore, the main use cases of the proposed method are when the training data are limited or hard to achieve and when the new sensor is introduced into the PAD system.

We consider the basic method implementation including the basic set of features and one of two classification algorithms linear SVM and RF and the additional feature set with some heuristic strategies of feature improvement. The proposed method was verified using the WMCA database. The comparison of basic method implementation with two baseline methods showed that the proposed method gives better results than the RDWT-Haralick-SVM method, however, it loses to MC-CNN method in the case of large training dataset. Nevertheless, in the case of 50 times smaller training dataset, we suppose that the MC-CNN method cannot be successfully trained in contrast to the proposed method, which was successfully trained and demonstrated ACER=2.91% in this case. Additional experiments with heuristic strategies of feature improvement allowed us to decrease the classification error almost 1.84 times in comparison with the best basic method implementation.

5. ACKNOWLEDGMENTS

The reported study was funded by RFBR, project number 19-29-09045.

6. REFERENCES

[Abd21] Abdullakutty, F., Elyan, E., and Johnston, P. A review of state-of-the-art in Face Presentation Attack Detection: From early development to advanced deep learning and multi-modal fusion methods. Information fusion 75, pp. 55-69, 2021. DOI: 10.1016/j.inffus.2021.04.015.

[Anj11] Anjos, A., and Marcel, S. Counter-measures to photo attacks in face recognition: a public database and a baseline Biometrics (IJCB). 2011 international joint conference on Biometrics

(IJCB), pp. 1-7, 2011. DOI: 10.1109/IJCB.2011.6117503.

[Bha19] Bhattacharjee, S., Mohammadi, A., Anjos, A., and Marcel, S. (2019). Recent advances in face presentation attack detection. Handbook of Biometric Anti-Spoofing, pp. 207-228, 2019. DOI: 10.1007/978-3-319-92627-8_10.

[Bre19] Bresan, R., Pinto, A., Rocha, A., Beluzo, C., and Carvalho, T. (2019). Facespoof buster: a presentation attack detector based on intrinsic image properties and deep learning. arXiv preprint arXiv:1902.02845, 2019. 10.48550/arXiv.1902.02845.

[Chr00] Christianini, N. and Shawe-Taylor, J. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge, UK: Cambridge University Press, 2000.

[Den21] Denisova, A. and Fedoseev, V. Presentation Attack Detection in Facial Authentication using Small Training Dataset Obtained by Multiple Devices. 2021 International Conference on Information Technology and Nanotechnology (ITNT), pp. 1-5, 2021. DOI: 10.1109/ITNT52450.2021.9649390.

[Du21] Du, Y., Qiao, T., Xu, M., and Zheng, N. (2021). Towards Face Presentation Attack Detection Based on Residual Color Texture Representation. Security and Communication Networks, pp. 6652727, 2021. DOI: 10.1155/2021/6652727.

[Ewa20] Ewald, K.E., Zeng, L., Zhengyao, Mawuli, C.B., Abubakar, H.S. and Victor, A. Applying CNN with Extracted Facial Patches using 3 Modalities to Detect 3D Face Spoof. 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), pp. 216–221, 2020. DOI: 10.1109/ICCWAMTIP51612.2020.9317329.

[Fre12] de Freitas Pereira, T., Anjos, A., De Martino, J. M., and Marcel, S. LBP− TOP based countermeasure against face spoofing attacks. Asian Conference on Computer Vision, pp. 121-132, 2012. DOI: 10.1007/978-3-642-37410-4_11.

[Gal14] Galbally, J., Marcel, S. and Fierrez, J. Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition. IEEE transactions on image processing 23, No. 2, pp. 710–724, 2014. DOI: 10.1109/TIP.2013.2292332.

[Geo19] George, A., Mostaani, Z., Geissenbuhler, D., Nikisins, O., Anjos, A., and Marcel, S. Biometric face presentation attack detection with multi-channel convolutional neural network. IEEE Transactions on Information Forensics and Security 15, pp. 42-55, 2019. DOI: 10.1109/TIFS.2019.2916652.

[Har73] Haralick, R., Shanmugam, K., Dinstein, I. Textural features for image classification. IEEE TSMC 3, No. 6, pp. 610-621, 1973. DOI: 10.1109/TSMC.1973.4309314.

[Kow20] Kowalski, M. A Study on Presentation Attack Detection in Thermal Infrared. Sensors 20, No. 14, pp. 3988, 2020. DOI: 10.3390/s20143988.

[Kul12] Kulkarni, V.Y. and Sinha, P.K. Pruning of random forest classifiers: A survey and future directions. 2012 International Conference on Data Science & Engineering (ICDSE), pp. 64-68, 2012. DOI: 10.1109/ICDSE.2012.6282329.

[Liu09] Liu, C. Beyond Pixels: Exploring New Representations and Applications for Motion Analysis. Ph.D. Dissertation. Citeseer, 2009.

[Pen18] Peng, F., Qin, L. and Long, M. Face presentation attack detection using guided scale texture. Multimedia Tools and Applications 77, No. 7, pp. 8883-8909, 2018. DOI: 10.1007/s11042-017-4780-0.

[Pen20] Peng, F., Qin, L. and Long, M. Face presentation attack detection based on chromatic co-occurrence of local binary pattern and ensemble learning. Journal of Visual Communication and Image Representation 66, pp. 102746, 2020. DOI: 10.1016/j.jvcir.2019.102746.

[Wan19] Wang, G., Lan, C., Han, H., Shan, S., and Chen, X. Multi-modal face presentation attack detection via spatial and channel attentions. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2019.

[Wan20] Wan, J., Guo, G., Escalera, S., Escalante, H. J., and Li, S. Z. Multi-Modal Face Presentation Attack Detection. Synthesis Lectures on Computer Vision 9, No. 1, pp. 1-88, 2020. DOI: 10.2200/S01032ED1V01Y202007COV017.

[Wen15] Wen, D., Han, H. and Jain, A. K. Face spoof detection with image distortion analysis. IEEE Transactions on Information Forensics and Security 10, No. 4, pp. 746–761, 2015. DOI: 10.1109/TIFS.2015.2400395.

[Wu18] Wu, X., He, R., Sun, Z. and Tan T. A Light CNN for deep face representation with noisy labels. IEEE Transactions on Information Forensics and Security 13, No. 11, pp. 2884-2896, 2018. DOI: 10.1109/TIFS.2018.2833032.

Development of a Multi-spectral Camera for Computer vision applications

Vahid Mohammadi
9 Avenue Alain Savary
ImViA laboratory
UFR Sciences et Techniques, Université de Bourgogne
Franche-Comté, 21000, Dijon, France
Vahid.Mohammadi@u-bourgogne.fr

Sovi Guillaume Sodjinou
BP 526, Cotonou, Littoral
Institute of Mathematics and Physical Sciences
University of Aboney Calavi
Dangbo, Benin
guillaume.sodjinou@imsp-uac.org

Kossi Kuma Katakpe
BP 526, Cotonou, Littoral
Institute of Mathematics and Physical Sciences
University of Aboney Calavi
Dangbo, Benin
kossi.katakpe@imsp-uac.org

Matthieu Rossé
9 Avenue Alain Savary
ImViA laboratory
UFR Sciences et Techniques, Université de Bourgogne
Franche-Comté, 21000, Dijon, France
matthieu.rosse@u-bourgogne.fr

Pierre Gouton
9 Avenue Alain Savary
ImViA laboratory
UFR Sciences et Techniques, Université de Bourgogne
Franche-Comté, 21000, Dijon, France
pgouton@u-bourgogne.fr

ABSTRACT

The need to multispectral cameras is growing in different fields. The goal is to provide inexpensive, flexible, or high resolution acquisition set-ups for different applications. Production of cheap and easy-to-use multispectral cameras requires the design and development of specific multispectral camera sensors equipped with multispectral filter arrays. Filter arrays, band width, and spectral response of filters should be analyzed and determined before mounting the filters on the image sensor. In this study, a multispectral camera sensor was designed to be used for computer vision applications (e.g. crop/weed detection, fruit ripeness estimation etc.) covering visible range. The developed camera sensor is consisted of the image sensor, spectral filter array, a sensor board, and a driving board. A hybrid system is proposed that works using eight bands in visible range (i.e. 400-700 nm). A program was developed based on Genetic Algorithm to find the best combination of filters. The program selects the Gaussian filters using a genetic algorithm powered by wiener filter estimation method. For the selection of bands, minimum RMS of 0.0016 was obtained for the selected bands in visible. The developed camera provides eight high resolution spectral images.

Keywords

Multispectral camera; Spectral filter array; Image sensor; Hybrid sensor design; Genetic algorithm.

1. INTRODUCTION

The use of Multispectral (MS) cameras for various applications in different fields is growing as these cameras provide efficient information in a single-shot image [Shi17]. Also, MS imaging reduces the cost of hyperspectral imaging and analysis. Spectral bands in MS cameras are chosen based on the application and the cameras are designed specifically for special applications [Fre15]. Hence, the specific filter array would be mounted on the image sensor for the desired application. Design and fabrication of MS cameras is of high importance as reduction of cost and increasing the quality and resolution play significant roles. The cost of fabrication is quite high which limits the development of this technology [Lap17]. The most popular technologies currently used in some MS imaging systems are Dichroic filter, Fabry-Perot, Multi-Aperture Filtered Camera, and PixelTEQ Camera.

In previous research, several Multi-Spectral Filter Arrays (MSFA) and MS cameras have been designed and developed [Sun18, Bol18, Cao19, Gut19, Gen20].

https://www.doi.org/10.24132/CSRN.3201.4

[Fre15] proposed a design of a multispectral filter array with an extended spectral range spanning the visible and near-infrared range, using a single set of materials and realizable on a single substrate. They experimentally illustrated also the ability of multispectral nanostructured Fabry−Perot (FP) filters to provide precisely constant peak wavelength across the whole surface of an image sensor in the focal plane of an optical system. [Par17] designed a multispectral imaging system with an onboard flight controller for acquiring multispectral aerial images of crops. The MS system consisted of 6 bands and was used for several crops. [Shi18] proposed an MSFA for snapshot multispectral polarization imaging. The MSFA was a photonic crystal which was used as thin-film wavy multilayer structure. [Wil19] reported a manufacturing process that enables cost-effective wafer-level fabrication of custom MSFAs in a single lithographic step, maintaining high efficiencies (∼75%) and narrow line widths (∼25 nm) across the visible to near-infrared. In a recent report, [Yu20] designed a new multispectral imaging system, named multispectral curved compound eye camera (MCCEC). The proposed MS system consisted of three subsystems including a curved micro-lens array, an optical transformation subsystem, and the data processing unit.

In this work, the development of a fast and cheap MS camera for usage in a wide of computer vision applications is presented. The goal was to provide a sensor based on the MSFA (Multi-spectral filter array) technology that allows to cover from 380 nm to 780 nm in eight different spectral bands.

2. MATERIALS AND METHOD

2.1 Image sensor

2.1.1 Sensor selection

The Hybrid sensor was based on E2V Sapphire (Teledyne E2V, Model: EV76C570, UK) which is a 2k black and white sensor. The EV76C570 is a 2k CMOS image sensor designed with E2V's proprietary Eye-On-Si CMOS imaging technology (Table 1). It is ideal for many different types of applications where superior performance is required. The innovative pixel design offers excellent performance in low-light conditions with an electronic global (true snapshot) shutter, and offers a high-readout speed at 50 fps in full resolution.

2.1.2 Measurement of the spectral response

To estimate the spectral response of the filter, a global estimation of the MS imaging sensor was performed. To perform this calibration, four steps were defined:

- The energy generated by the monochromator from 350 to 1000 nm was measured
- The system generates a monochromatic light each 5 nm from 350 to 1000 nm. So the original CMOS sensor (without a filter mounted) was illuminated. The integrated time of the camera was set as constant.
- The second measure was done with the MS imaging sensor. An algorithm was built to extract from each moxel the pixel related to a specific filter.
- The filter spectral response was obtained from the CMOS sensor response measured in step 2.

Main features of the sensor	▪ 2 million (1600 x 1200) pixels, ▪ 4.5 μm^2 pixel size with micro-lens ▪ Optical format 1/1.8” ▪ 50 fps at full resolution
Performance characteristics	▪ Low power consumption (200mW) ▪ High sensitivity at low light level ▪ Operating temperature [-30° to +65°C] ▪ Peak QE > 48% ▪ B&W
Timing modes	▪ Global shutter in serial and overlap modes ▪ Rolling shutter and Global Reset modes ▪ Output format 8 or 10 bits parallel plus synchronization

Table 1. Technical properties of the image sensor.

2.2 Sensor development

2.2.1 Sensor structure

The global system has 3 blocks that work together for receiving correct images:

- The sensor block was designed to provide the control of the E2V CMOS sensor.
- The FPGA block which was designed via Xilinx's VIVADO platform, whose role was to enable the synchronization of data exchanges from the sensor board, viewing directly on a screen and data acquisition to the computer.
- The Computer block, which contained the software allowed the acquisition, processing and exploitation.

2.2.2 Sensor Board Design

The E2V's EV76C570 sensor offers a dynamic 10-bit output range in digital playback. It includes features such as the ability to have the output image histogram, the multiple ROIs (Region of Interest), the faulty pixel correction, the global shutter, etc. There is also the possibility of setting up a sequence to acquire successive images with different exposure times (up to four times). Because of this feature, then the reconfiguration of the sensor registers between each image is removed. It also has the distinction of having a good sensitivity to low light levels and low consumption (200mW). This low consumption makes it suitable and usable for on-board systems such as smart cameras, and battery-powered applications. The proposed architecture was designed to be able to adapt to a large number of affordable sensors on the market and then the choice of this type of conventional sensors is quite wise.

2.2.3 Hardware description and Specification

For the hardware, FPGA Zboard / Zybo development kit was used having main characteristics as follows:

- The ZYBO (ZYnq Board) is a feature-rich, ready-to-use, entry-level embedded software and digital circuit development platform built around the smallest member of the Xilinx Zynq-7000 family (i.e. Z-7010).
- The Z-7010 is based on the Xilinx all-programmable System-on-Chip (AP SoC) architecture, which tightly integrates a dual-core ARM Cortex-A9 processor with Xilinx 7-series Field Programmable Gate Array (FPGA) logic.
- When coupled with the rich set of multimedia and connectivity peripherals available on the ZYBO, the Zynq Z-7010 can host the whole system design.
- The on-board memories, video and audio I/O, dual-role USB, Ethernet, and SD slot will have your design up-and-ready with no additional hardware needed.

2.2.4 Hardware Block Design and Description

To carry out the Hardware design, the VIVADO/ Xilinx (version 2019.2) development platform was utilized which is a graphical programming environment. Of the benefits of using Vivado is that it allows to quickly detect programming code errors. The design block was performed to control the coming signals from the sensor, so a link between the processor part and the PL part was necessary. To create the design block of the project under Vivado, the IP (Intellectual Property) blocks were made available for this purpose. IPs are available coded modules that can be added to a design. All IPs are binary codded (protected). The block diagram below (Fig 1) gives the most important blocks used in the FPGA for data exchange with the sensor board and computer pour displaying the video.

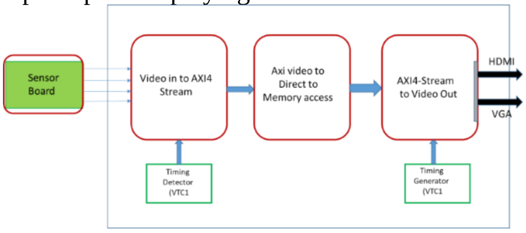


Figure 1. Block Diagram of the FPGA.

The sensor generates the 10 bits of data, in parallel. To be able to process the data, all IP blocks used in the design were as follows:

- *VDMA (Video Direct Memory Access):* To make the link between the microprocessor and memory and then between the Video In and Video Out blocks.
- *Video in to axi4-stream:* Entrance block, receiving sensor data.
- *Axi4-stream to video out*: Exit block that transfers data to HDMI/VGA outputs for visualization.
- *VTC:* Block that detects and generates pixel flows.
- *AXI Interconnect:* It is a block that connects one or more master devices mapped in memory to one or more slave devices mapped in memory.
- *AXI Quad SPI:* Connects the AXI4 interface with the slave SPI blocks that supports the Dual or Quad SPI protocol.

The developed sensor card was connected to the Zedboard (DPGA from Digilent) via a deported and very flexible connection.

2.3 MSFA design

2.3.1 Architecture of the filter

The Fabry-Perot theory was used to create the filter array. The approach is called Color Shade. Due to several constraints, the MSFA was designed in the to solve the problem of energy balancing which is generally the weakness of MS imaging systems [Tho17]. In several Multispectral Systems the problem of energy balancing can be solved by acting on either the bandwidth of the amplitude of each filter element. So to overcome to this weakness, a complex distribution of the MSFA distribution was proposed. Fig. 2 presents a basic MFSA used in this project.

P1	P5	P2	P6
P7	P3	P8	P4
P2	P6	P1	P5
P8	P4	P7	P3

Figure 2. Spatial Distribution for MSFA moxel.

2.3.2 Band selection

The test was carried out using 1269 matt Munsell spectral reflectance data. A white noise of SNR=100 has also been added to the data. The bandwidth is constant and equal to 30 nm which is a good compromise of an overlap filter and a narrow one. All data rearranged between 380-780 nm with 5 nm intervals. Fitness function was ΔE2000 between Wiener filter estimation the actual Munsell spectral data.

3. RESULTS AND DISCUSSION

3.1 Image sensor

Figure 3 shows the spectral response of the E2V sensor. The spectral response was done by projecting light to the sensor by wavelength steps of 5 nm. This figure shows the sensitivity of the sensor for different wavelengths. This spectral response of the sensor should be taken into account for assembly of the MFA and image reconstruction.

The performance of the whole camera and sensor board is observed in Fig. 4. This figure represents the gray-scale images taken by the image sensor which means that the data has been collected well, converted and constructed the image correctly.

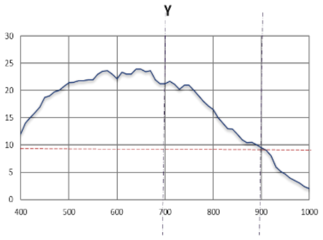


Figure 3. Spectral response of EV76C570 sensor and its impact on the MSI camera.

3.2 MSFA

The statistical results of RMS and goodness of fitness coefficient (GFC) error metrics of the spectral data were calculated. Mean of RMS error for 7 and 8 selected filters exhibit 56 percent improvement in comparison with the 3 selected filters. Moreover, 99% of the reconstructed data using 7 or 8 selected filters have been able to provide a GFC ≥ 0.95.

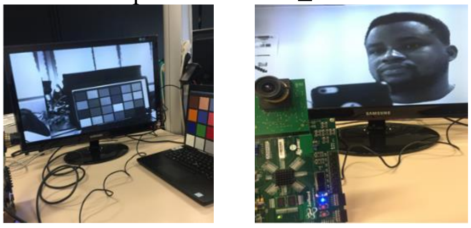


Figure 4. Online imaging using the camera.

When applying the filter on the CMOS, the global response is obtained by spectral response of each filter. We observe that the impact of the sensor leads to amplify the difference between different filters. The central filters are less impacted, while the extreme filters are mitigated (Fig. 5).

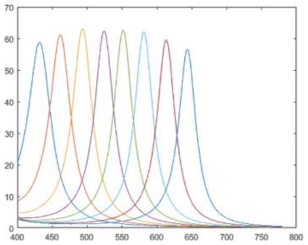


Figure 5. Set of 8 filters without the sensor

4. Conclusions

In this work, a MS camera was developed to be used for different computer vision applications including agricultural systems for weed detection or fruit ripeness estimation. The aim of the study was to prepare the image sensor, the firmware and software and the MSFA for the development of cheaper and faster MS cameras. A monochrome CMOS sensor was used as the camera sensor and its spectral response was evaluated. For the creation of the MSFA, the Fabry-Perot theory was utilized. A distribution of the MSFA was proposed including of 4x4 pixels, with a total of 16 pixels. The best combination of wavelengths was carried of using genetic algorithm. Mean of RMS error for 7 and 8 selected filters exhibit 56 percent improvement in comparison with the 3 selected filters. The camera works properly and an amount of 99% of the reconstructed data using 7 or 8 selected filters have been able to provide a GFC of more than 0.95.

5. ACKNOWLEDGMENTS

The authors would like to appreciate the help and support provided by ImViA (Imagerie et Vision Artificielle) laboratory, University of Burgundy, France.

6. REFERENCES

[Bol18] Bolton, F.J., Bernat, A.S., Bar-Am, K., Levitz, D., Jacques, S. Portable, low-cost multispectral imaging system: design, development, validation, and utilization. Journal of biomedical optics, 23(12): 121612, 2018.

[Cao19] Cao, A., Pang, H., Zhang, M., Shi, L., Deng, Q., Hu, S. Design and fabrication of an artificial compound eye for multi-spectral imaging. Micromachines, 10(3): 208, 2019.

[Fre15] Frey, L., Masarotto, L., Armand, M., Charles, M.L., Lartigue, O. Multispectral interference filter arrays with compensation of angular dependence or extended spectral range. Optics express, 23(9): 11799-11812, 2015.

[Gut19] Gutiérrez, S., Wendel, A., Underwood, J. Spectral filter design based on in-field hyperspectral imaging and machine learning for mango ripeness estimation. Computers and Electronics in Agriculture, 164: 104890, 2019.

[Gen20] Genser, N., Seiler, J., Kaup, A. Camera array for multi-spectral imaging. IEEE Transactions on Image Processing, 29: 9234-9249, 2020.

[Lap17] Lapray, P.J., Thomas, J.B., Gouton, P., Ruichek, Y. Energy balance in Spectral Filter Array camera design. Journal of the European Optical Society-Rapid Publications, 13(1): 1-13, 2017.

[Par17] Paredes, J.A., González, J., Saito, C., Flores, A. Multispectral imaging system with UAV integration capabilities for crop analysis. In 2017 First IEEE International Symposium of Geoscience and Remote Sensing (GRSS-CHILE) (1-4). IEEE, 2017.

[Shi17] Shinoda, K., Yanagi, Y., Hayasaki, Y., Hasegawa, M. Multispectral filter array design without training images. Optical Review, 24(4): 554-571, 2017.

[Shi18] Shinoda, K., Ohtera, Y., Hasegawa, M. Snapshot multispectral polarization imaging using a photonic crystal filter array. Optics express, 26(12): 15948-15961, 2018.

[Sun18] Sun, B., Yuan, N., Cao, C., Hardeberg, J.Y. Design of four-band multispectral imaging system with one single-sensor. Future Generation Computer Systems, 86: 670-679, 2018.

[Tho17] Thomas, J.B., Lapray, P.J., Gouton, P. HDR imaging pipeline for spectral filter array cameras. In Scandinavian Conference on Image Analysis (401-412). Springer, Cham, 2017.

[Wil19] Williams, C., Gordon, G.S., Wilkinson, T.D., Bohndiek, S.E. Grayscale-to-color: scalable fabrication of custom multispectral filter arrays. ACS photonics, 6(12): 3132-3141, 2019.

[Yu20] Yu, X., Liu, C., Zhang, Y., Xu, H., Wang, Y., Yu, W. Multispectral curved compound eye camera. Optics express, 28(7): 9216-9231, 2020.

Visualizing Statistical Complexity in 3D Turbulent Flows using a Robust Entropy Calculation Method

Arne Präger
Leipzig University
Augustusplatz 10
04109, Leipzig, Germany
praeger@informatik.uni-leipzig.de

Baldwin Nsonga
Leipzig University
Augustusplatz 10
04109, Leipzig, Germany
nsonga@informatik.uni-leipzig.de

Gerik Scheuermann
Leipzig University
Augustusplatz 10
04109, Leipzig, Germany
scheuermann@informatik.uni-leipzig.de

ABSTRACT

Highly resolved flow simulation data is becoming more common. These simulations frequently feature high turbulence with complex flow patterns. Finding these regions often requires expert knowledge, and in more complex cases, flow patterns of interest may remain hidden. The concept of statistical complexity was shown to be suitable to indicate regions of interest within flow data with limited prior knowledge. One way to determine the statistical complexity of flow fields is via the local vector field entropy and the correlation. In this work, we improve the method for calculating the Shannon entropy in vector fields. To this end, we introduce a robust entropy computation that takes the scale of the corresponding regions into account. The improved method uses a novel way to determine the distributions required for the entropy calculation and is applicable to unstructured domains. We validate our method with analytic flow fields and apply it to fluid simulation data, visualizing the results via volume rendering. This work shows the applicability of our technique to highlight regions of interest in turbulent flows.

Keywords

flow visualization, data exploration, information theory, statistical complexity, turbulence visualization

1 INTRODUCTION

Three-dimensional flow simulations have become increasingly complex [TLMF21]. Feature detection methods (e.g., vortex identification) and 2D cuts with color mappings of physical properties, e.g., velocity or pressure, are most common for visual analysis [BCP+12, ZH15]. Even though these visualizations are helpful, they require expert knowledge of specific configurations to understand which flow features in which regions are of interest and how to detect them with appropriate parameters.

Our goal is to provide a tool to assist the researcher in identifying regions of interest in individual time steps, which the researcher can further investigate purposefully. To this end, we apply the concept of statistical complexity based on the Shannon entropy [NT12] of vector directions which Xu et al. [XLS10] successfully applied to flow fields. Even though the Shannon entropy could be applied directly, Arbona et al. [ABMP14] argue that complete chaos and complete order should have similar value to the analysis. In contrast, entropy exhibits minimal values for complete order and maximum values for complete chaos.

Our primary focus is improving on the Shannon entropy computation. After providing a background on statistical complexity and Shannon entropy, we discuss the interpretation of statistical complexity in flow fields in general and provide arguments for the usefulness of this type of analysis for turbulent flows. We then discuss the methodology which distinguishes itself from the common approach by: (I) applying a neighborhood scheme suitable for unstructured domains, (II) utilizing the concept of information gain to set neighborhood sizes for the entropy computation dynamically, and (III) introducing a binning scheme to reduce overestimated complexity values. We validate our method with analytic datasets and apply it to a Direct Numerical Simulation (DNS) dataset. The results are visualized via volume rendering.

2 RELATED WORK

In the context of environmental sciences, a recent survey conducted by Bujack and Middel [BM20] gives a broad overview of various flow visualization techniques.

Geometric approaches based on the visualization of integral structures, e.g., streamlines or path-

https://www.doi.org/10.24132/CSRN.3201.5

lines, were discussed by McLoughlin [MLP+09]. Feature-based visualization approaches aim to detect and represent specific flow patterns like vortices [Hal05, JWM88, JH95], shock waves [WXWH13] ,or splat events [NNF+20, NSG+20].

Methods based on vector field topology visualize the topological skeleton consisting of critical points and segment the flow domain into regions of similar behavior [HH89, HH91, LHZP07]. Lagrangian coherent structures (LCS) can be interpreted as an unsteady analogy to flow topology and were extensively studied [Hal15, HS11, HY00].

As flow visualizations often depend on the chosen reference frame, a body of research concerned with this issue was published in recent years [BPKB14, GGT17, RG20, WGS05, WGS07].

Statistical complexity was introduced by Crutchfield and Young [CY89] and later refined by Lopez-Ruiz et al. [LRMC95] who separated the influences of entropy and structure.

Jaenicke et al. [JWSK07] have shown the usefulness of visualizing the statistical complexity computed via cellular automata in multifield data. They have also proven that statistical complexity can automatically detect flow features [JBTS08, JS10]. Arbona et al. [ABMP14] propose a method to apply the concept of statistical complexity to 3D vector data based on the product of Shannon entropy and local correlation.

Xu et al. [XLS10] applied the Shannon entropy to vector fields. They base the entropy at a point on the variance of the orientations of the vectors located in its neighborhood. Xu et al. [XLS10] did not directly visualize this information but used it to seed streamlines in information-rich regions instead. Wang et al. [WYM08] determine the entropy of dataset chunks to customize the level of detail used for their visualization. Furuya and Itoh [FI08], as well as Lee et al. [LMSC11], applied information-theoretic techniques to select a small number of streamlines out of a set optimally representing the original data. Tao et al. [TMWS12] and Ma et al. [MWW+14] used entropy to create automated camera cruises through possibly significant flow regions based on their entropy. An overview of how information-theoretic techniques can be utilized for visualization can be found in the book by Wang and Chen [CFV+16].

3 BACKGROUND

This section provides a background on entropy, statistical complexity, and how these concepts can be applied to flow fields.

3.1 Entropy of Velocity Fields

The Shannon entropy [NT12] describes the information content of a single variable. Its calculation is based on the distribution of the values of said variable. For any given variable X with the possible outcomes $\{x_1,..,x_n\}$ the Shannon entropy is:

$$H(X) = -\sum_{i=1}^{N} p(x_i)\log_2(p(x_i)), \tag{1}$$

where $p(x_i)$ denotes the probability with which the variable X takes the value x_i. $H(X)$ becomes minimal if the variable always assumes the same value and becomes maximal when all potential outcomes share the same possibility.

Xu et al. [XLS10] adapted the concept of entropy for vector fields. The entropy of single points is based on the variance of the orientations of vectors in their neighborhood. They do not specify in detail how a neighborhood is defined. As they use a manually set natural number as a parameter, we assume neighborhoods are grid-based. To generate the required distribution, they use vector orientations. They discretize a sphere and evaluate which vector orientation corresponds to which segment of the sphere. Note that the segments correspond to the bins of a histogram. To avoid bias, they utilize the technique proposed by Leopardi [Leo06] dividing the sphere surface into equally sized regions of similar shape. This technique divides the sphere into almost circular top and bottom areas as well as multiple rings with several quadrangular segments.

3.2 Statistical Complexity of Velocity Field

Arbona et al. [ABMP14] expanded upon the concept of vector field entropy with their definition of statistical complexity by taking local correlation into account. In their study, Arbona et al. [ABMP14] aim to determine the complexity at multiple scales, which they accomplish by utilizing meshes of varying sizes for the computation of the complexity field. They then compute the correlation between the mean velocity within a cell and its neighboring cells. The statistical complexity is then the product of the correlation and the entropy acquired using the method from Xu et al. [XLS10]. Their results show that complex structures become more visible when visualizing statistical complexity instead of the vector field entropy. This work and the velocity vector entropy computation by Xu et al. [XLS10] are the foundations of our research.

3.3 Information in Turbulent Flows

Visualizing entropy and statistical complexity gives insights into a flow with minimal prior knowledge. Xu et al. [XLS10] state that the entropy of a vector field should be suitable to identify critical points in non-converging flows. They show that it can also emphasize structures like regions near separation lines. This property is a result of the inherent nature of entropy. Low entropy values represent that the orientations of vectors in a neighborhood only vary slightly. Thus low entropy regions depict the more laminar parts of a flow. With rising variations in the flow directions, the corresponding entropy values climb as well. Turbulent flow fields, expected to result in high entropy values, exhibit structure depending on the chosen configuration and reference frame. High entropy values, emphasizing regions of high disorder, are not sufficient to visualize potential regions of interest, as highly turbulent regions will always be favored. On the other hand, statistical complexity gives more nuance to the visualization, reducing the impact of the pure disorder. As a result, using statistical complexity, we aim to visualize regions containing critical points and other complex flow patterns like coherent structures.

4 METHOD

We improve the computation of the entropy [XLS10] by making it more robust and applicable to irregular grids as well as utilizing dynamically computed neighborhood sizes. We adapt the method from Arbona et al. [ABMP14] from a cell-based to a point-based approach.

Our method takes a velocity field $\mathbf{v}(\mathbf{x},t)$ as input and computes the statistical complexity:

$$C(\mathbf{x},t) = H(\mathbf{x},t)D(\mathbf{x},t), \tag{2}$$

where $H(\mathbf{x},t)$ is the entropy and $D(\mathbf{x},t)$ is the velocity correlation at a point $\mathbf{x} \in G$ in the flow domain $G \subset \mathbb{R}^3$ at time $t \in \mathbb{R}$.

4.1 Neighborhood Definition

The computation of $H(\mathbf{x},t)$ and $D(\mathbf{x},t)$ requires a neighborhood definition. As stated before, we assume that the neighborhoods applied by Xu et al. [XLS10] depend on the grid of the flow domain. Neighborhoods based on grids can introduce a directional bias when the spacings between points are not equal along the axes. Furthermore a grid based approach may not suitable for non-uniform grids.

Our goal is to find a neighborhood definition free of directional bias which is applicable to any grid type. The neighbors should preferably be distributed equally in space so that the results of the entropy calculation are uninfluenced by the orientation of the field. In the following when we mention dodecahedra we always refer to convex, regular deodecahedra.

We base our neighborhood definition on dodecahedron vertices as they are equally distributed on the surface of a sphere and thus share the same distance to the point in its center. Even though all platonic solids share this property, we use the dodecahedron as it has the most vertices of them, leading to a fine resolution of the neighborhood. We define the neighborhood $n_d(\mathbf{x}) \subset G$ as the union of the vertices of d equidistant layers of dodecahedra (cf. Figure 1). The faces of neighboring dodecahedra layers are parallel. Note that, the number of points contained in the neighborhood is $|n_d(\mathbf{x})| = 20d$, as a single dodecahedron contains 20 vertices. We obtain a neighborhood consisting of points with uniform radial and angular spacing.

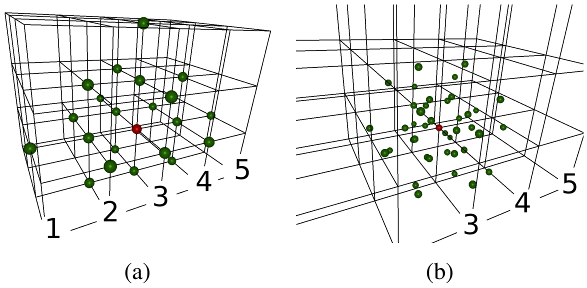


Figure 1: Neighborhoods of size two in green around the red point for (a) the old and (b) the new neighborhood definition

The neighborhood vectors used for the computation are then acquired via interpolation. To prevent aliasing due to undersampling, we respected the Nyquist frequency and chose the distance between two neighboring dodecahedra layers to be 0.4 of the minimal cell edge distance within the domain. Note that the density of neighborhood points decreases as the distance to $\mathbf{x}$ increases. This is justifiable as it creates differences between points in small areas of turbulence surrounded by a laminar field and points in small laminar areas surrounded by a turbulent field. The first case should result in higher entropy values which our method captures correctly.

4.2 Binning Strategy

As mentioned before Xu et al. [XLS10] segment a sphere into equal-sized areas using Leopardis algorithm [Leo06]. This introduces a bias, as flows in the direction of the poles are more robust to small variations in orientation than their orthogonal counterparts. Binning, in general, can lead to vectors with small variances in orientation being assigned to different bins. Especially in laminar flows, small variation can lead to an overestimated statistical complexity. An example of the artifacts resulting from this is shown in Section 5.1.

To avoid bias introduced through the shape of the segment bins, we used the triangular faces of an icosahedron as the base for our segmentation, as they all share the same shape and size. Each of these faces can be subdivided into four equilateral triangles of the same area. The user can set the number of subdivisions $s \in \mathbb{N}$. Using this subdivision strategy, we can provide different consistent binning resolutions as shown in Figure 2.

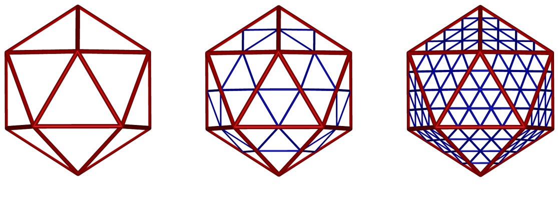

(a) (b) (c)

Figure 2: Icosahedral binning scheme using no (a), one (b) and two (c) subdivisions

To solve the issue of overestimated complexity values for small orientation variances, we apply a randomized rotation to the binning icosahedron. We then repeat the entropy calculation multiple times with different rotation angles. The number of iterations k can be set by the users. As the entropy of the velocity direction should be invariant to rotation, we can assume that high values result from overestimation. Therefore, after k iterations, we store the minimal value. The quality of the results increases in proportion to the chosen number k as artifacts resulting from small variances are not robust against variations of the binning. Note that the random rotations introduce noise. To reduce the noise without further increasing k, we repeat the same number of iterations for points with a higher statistical complexity value than all their edge neighbors. The effects of our binning technique are discussed in Section 5.

4.3 Dynamic Neighborhoods

Choosing appropriate neighborhoods is challenging, as two different regions might require a different number of dodecahedron layers d to highlight regions of interest of varying scales. To solve this, we introduce a local dynamic neighborhood size utilizing relative entropy. The general concept is that we only need to add additional layers to the neighborhood if their vectors contain new information. Beginning with the second layer, we calculate the relative entropy [KL51] between the current orientation distribution of the neighborhood and the distribution containing the vectors of the next layer. We define the distributions as follows.

For each triangular bin, we compute the number of neighbors $|\hat{n}_d(\mathbf{x},t)|$ where the corresponding (scaled) vectors intersect the triangle before and after adding another dodecahedron layer. This is divided by the number of all neighbors $|n_d(\mathbf{x},t)|$, resulting in:

$$p_i(\mathbf{x},t) = \frac{|\hat{n}_d(\mathbf{x},t)|}{|n_d(\mathbf{x},t)|} \quad p_i^+(\mathbf{x},t) = \frac{|\hat{n}_{d+1}(\mathbf{x},t)|}{|n_{d+1}(\mathbf{x},t)|}, \qquad (3)$$

where $p_i(\mathbf{x},t)$ is the estimate of the probability for neighbors of point $\mathbf{x}$ belonging to bin i. $p_i^+(\mathbf{x},t)$ denotes the estimate of the probability after adding the additional dodecahedron layer. As $p_i^+(\mathbf{x},t) = 0$ implies $p_i(\mathbf{x},t) = 0$, absolute continuity is provided, enabling us to apply relative entropy:

$$KL(\mathbf{x},t) = \sum_{i=1}^{N} p_i(\mathbf{x},t)\log_2 \frac{p_i(\mathbf{x},t)}{p_i^+(\mathbf{x},t)}, \qquad (4)$$

where $N = 20 * 4^s$ is the number of bins resulting from s subdivisions of the 20 icosahedron faces (cf. Figure 2). High relative entropy values indicate more significant differences between the two distributions. If $KL(\mathbf{x},t)$ is higher than a user defined threshold r, we repeat the process by setting $p_i(\mathbf{x},t)$ to $p_i^+(\mathbf{x},t)$ and adding the next layer.

4.4 Entropy and Correlation

We can now compute the entropy as follows:

$$H(\mathbf{x},t) = -\sum_{i=1}^{N} p_i(\mathbf{x},t)\log_2 p_i(\mathbf{x},t), \qquad (5)$$

where $p_i(\mathbf{x},t)$ is the distribution after the relative entropy has fallen below the threshold r and $N = 20 * 4^s$ is the number of bins. The correlation is computed by adapting the cell-based strategy from Arbona [ABMP14] to our point based approach:

$$D(\mathbf{x},t) = \frac{1}{N}\sum_{i=1}^{N} \frac{\mathbf{v}(\mathbf{x},t)\mathbf{v}(\mathbf{x}_i,t)}{\mathbf{v}(\mathbf{x},t)^2 + \mathbf{v}(\mathbf{x}_i,t)^2} + \frac{1}{2}, \qquad (6)$$

where $\mathbf{x}_i \in n(\mathbf{x},t)$ is the i^{th} neighbor vertex position in the neighborhood of point $\mathbf{x}$. Note that close to boundaries, neighborhood points can be out of bounds and have to be excluded from the calculation. To prevent the correlation values from being influenced by a varying number of neighborhood points, we set N to $20d$.

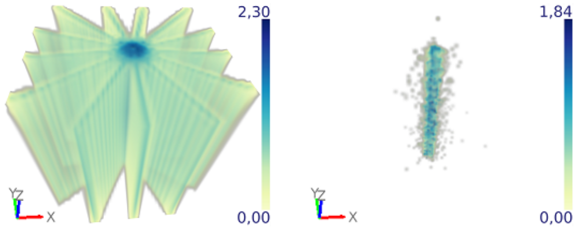
We can now compute the statistical complexity using Equation 2. Note that the results of the presented method are invariant to the magnitude of the velocity vectors.

4.5 Visualization

Developing a novel and suitable visualization for the regions of interest is not within the scope of this work. As the statistical complexity $C(\mathbf{x},t)$ is a scalar field, several established visualization techniques are available. For an in-depth interactive analysis of the results, we apply volume rendering [DCH88]. This is a GPU-based ray casting technique that renders volumes with different transparencies based on their scalar values. We explore the range of the results and emphasize values exhibiting structure. It is to note that the utilized colormap can be adjusted in real-time. An example of the resulting visualization is shown in Figure 3.

5 EVALUATION

In this section, we present the results of our evaluation. First, we show that our method produces the expected



(a) Previous binning approach (b) New binning approach

Figure 3: A comparison of the statistical complexities of $\mathbf{v}_{\text{rot}}(\mathbf{x})$ for $r = 0.01$, $s = 1$ and $k = 5$

results when applying it to analytic test cases. Then we conduct a parameter study to provide suggestions for finding suitable input parameters. Finally, we investigate the performance of our method.

5.1 Application to Analytic Fields

To evaluate our method, we conducted experiments on analytical 3D data sets. We construct the steady fields:

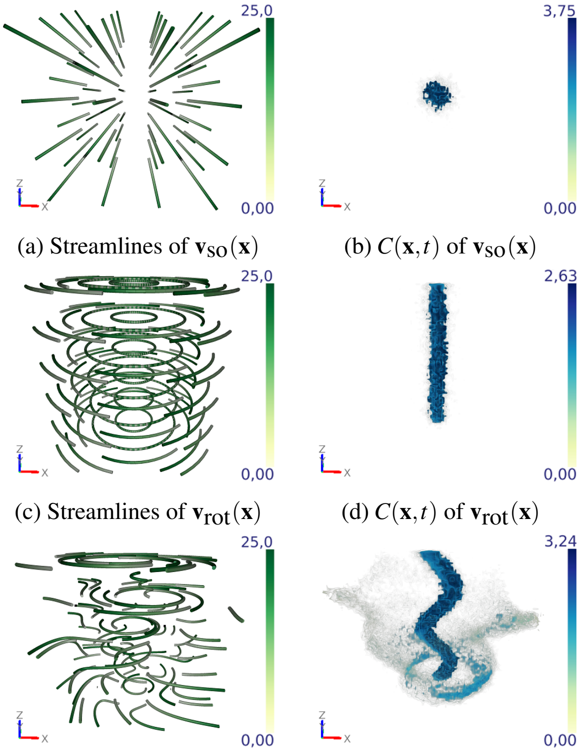
$$\mathbf{v}_{\text{so}}(\mathbf{x}) = [x, y, z]^T \quad \mathbf{v}_{\text{rot}}(\mathbf{x}) = [y, -x, 0]^T, \tag{7}$$

where $\mathbf{v}_{\text{so}}(\mathbf{x})$ is a source and $\mathbf{v}_{\text{rot}}(\mathbf{x})$ is a rotation. Additionally we constructed a steady Crawfis tornado [Cra03] we denote as $\mathbf{v}_{\text{C}}(\mathbf{x})$. The values were then sampled on a $50 \times 50 \times 50$ grid within the domain $[-24.5, 24.5] \times [-24.5, 24.5] \times [-24.5, 24.5]$.

Figure 3 shows a comparison between the classical binning method and our binning scheme applied to $\mathbf{v}_{\text{rot}}(\mathbf{x})$. The left image demonstrates that the classic binning can produce artifacts. This is consistent with the results shown in Figure 3 in [XLS10]. These artifacts, however, are not robust against the binning scheme and are reduced. Note that, as a downside, noise is introduced. This is a clear improvement, as the noise in the right figure is clearly identifiable, whereas the artifacts in the left figure could imply structure in an unknown dataset. We discuss the effect of the number of iterations in the following section.

Figure 4 shows volume rendering visualizations of the results of our statistical complexity approach in comparison to streamlines seeded uniformly in the field. As expected, the center of the source field and the rotation center of $\mathbf{v}_{\text{rot}}(\mathbf{x})$ are the regions with the highest statistical complexity. The rotation center of the Crawfis tornado is also clearly visible. Note that the mantle-like surface around the tornado is inherent to the dataset and also appears when applying vortex identification methods.

Through these experiments, it becomes clear that visualizing the statistical complexity is a useful method to highlight critical points, vortex cores, and other potential regions of interest.



(a) Streamlines of $\mathbf{v}_{\text{so}}(\mathbf{x})$ (b) $C(\mathbf{x},t)$ of $\mathbf{v}_{\text{so}}(\mathbf{x})$

(c) Streamlines of $\mathbf{v}_{\text{rot}}(\mathbf{x})$ (d) $C(\mathbf{x},t)$ of $\mathbf{v}_{\text{rot}}(\mathbf{x})$

(e) Streamlines of $\mathbf{v}_{\text{C}}(\mathbf{x})$ (f) $C(\mathbf{x},t)$ of $\mathbf{v}_{\text{C}}(\mathbf{x})$

Figure 4: A comparison of the visualization of streamlines and the statistical complexities $C(\mathbf{x},t)$ of analytic datasets for parameters $r = 0.01$, $s = 1$ and $k = 5$

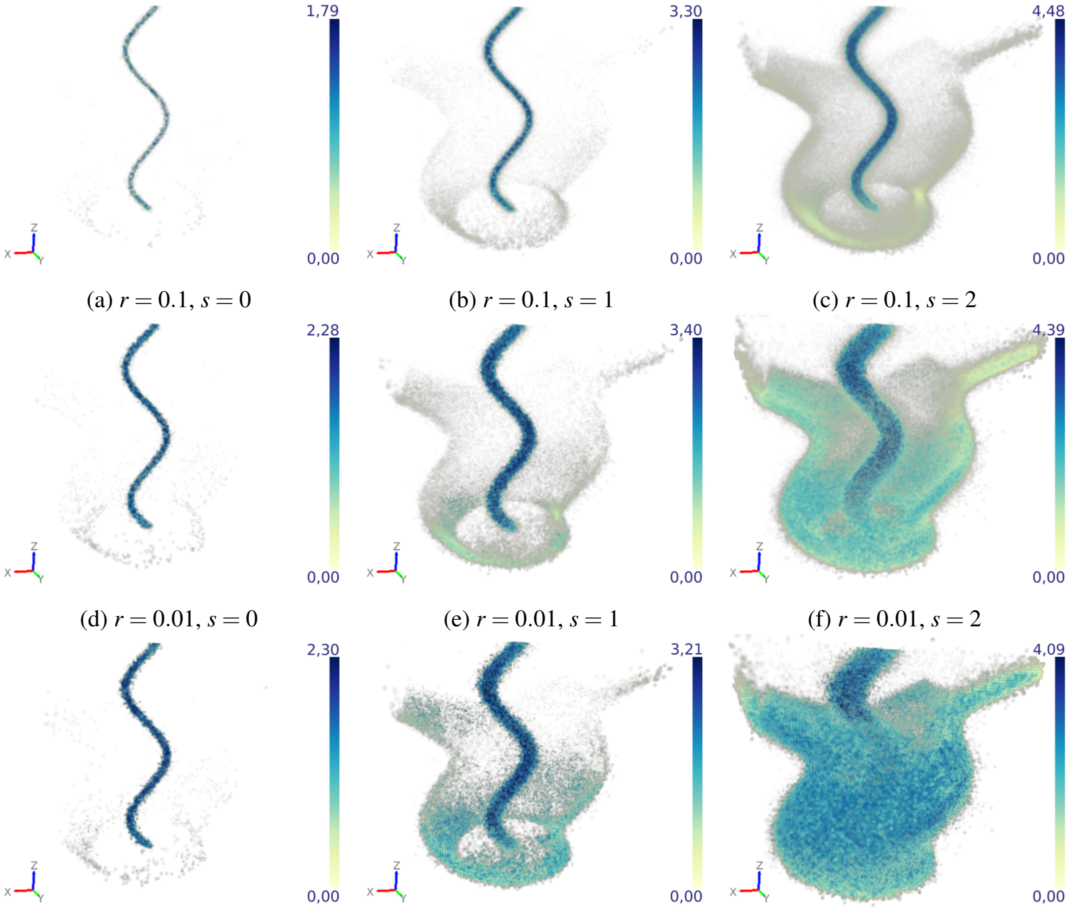
5.2 Parameter Study

5.2.1 *Information Gain and Bin Resolution*

The most impacting parameters in our method are the relative entropy threshold utilized during the dynamic neighborhood size computation and the number of subdivisions of the icosahedron faces used for binning. In this experiment, we compared the results of our algorithm for the Crawfis tornado defined on a $100 \times 100 \times 100$ grid for combinations of these parameters. We apply five iterations ($k = 5$) to provide a robust visualization. For this analysis, we base the evaluation on the quality on the following criteria: (1) the highest statistical complexity values lie exclusively in the vortex core line, and (2) medium values should be located around the vortex core and on the mantle shaped surface (cf. Figure 4f).

The results in Table 1 show that a relative entropy threshold of $r = 0.1$ is too high since the information gained through adding a new layer is rarely enough to surpass it. This leads to small neighborhoods which may not be able to capture the behavior of the flow correctly (cf. Table 1). Figure 5 (first row) shows that no clear distinction between regions with different statistical complexities is possible and our criteria are not met.

A relative entropy threshold of 0.01 delivers the desired results as it allows the neighborhoods to grow to viable



(a) $r = 0.1$, $s = 0$ (b) $r = 0.1$, $s = 1$ (c) $r = 0.1$, $s = 2$

(d) $r = 0.01$, $s = 0$ (e) $r = 0.01$, $s = 1$ (f) $r = 0.01$, $s = 2$

(g) $r = 0.001$, $s = 0$ (h) $r = 0.001$, $s = 1$ (i) $r = 0.001$, $s = 2$

Figure 5: Visualization of the statistical complexity on a $100 \times 100 \times 100$ Crawfis tornado dataset with $k = 5$ and varying r and s

$[d_{min}, d_{max}]$	$s = 0$	$s = 1$	$s = 2$
$r = 0.1$	$[2,4]$	$[2,6]$	$[2,9]$
$r = 0.01$	$[2,13]$	$[3,21]$	$[2,30]$
$r = 0.001$	$[2,34]$	$[2,52]$	$[2,88]$

Table 1: Neighborhood size d intervals resulting from specific subdivision numbers s and information gain thresholds r on the Crawfis tornado dataset (the table corresponds to Figure 5)

sizes. Subdivision of $s = 1,2$ seems sufficient as the difference between areas of different statistical complexity values becomes clearly visible.

For a threshold of 0.001 the neighborhood sizes become larger, accentuating large-scale features. Increasing the subdivision intensifies the phenomenon further (cf. Table 1).

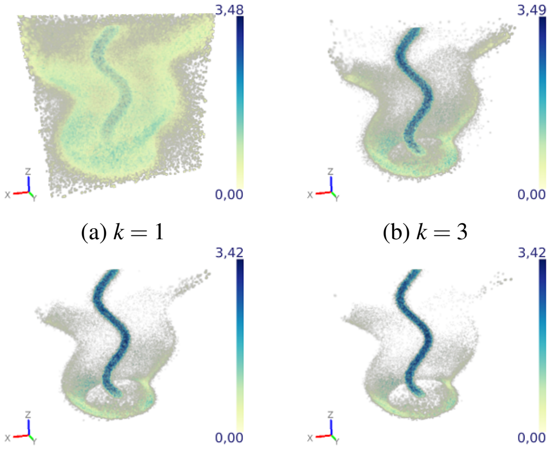
As a strategy for parameter selection, we recommend exploring the parameter space of the threshold r, starting with small values, e.g., $r \approx 0.001$. Choosing a small initial value reduces the risk of false negatives. By increasing the threshold, structures may persist, informing the user of potential regions of interest.

Subdividing the surface of the icosahedron one and two times is both viable. While the first approach shortens the calculation time, the second one should perform better for stronger turbulences as it can capture finer details of the behavior of a flow.

5.2.2 *Number of Iterations*

We conducted another experiment to determine a recommendation for the number of iterations needed to create reliable results. We used the $100 \times 100 \times 100$ Crawfis tornado. The relative entropy threshold was set to 0.01, and we subdivided the icosahedron one time since these parameter settings have produced satisfactory results in the previous test.

Figure 6 displays the statistical complexity field after a different amount of iterations. The visual noise is drastically reduced after three iterations, but some is still present. After five iterations, most of the higher statistical complexity values are near the vortex core and the mantle surface. Increasing the number of iterations



(a) $k = 1$ (b) $k = 3$

(c) $k = 5$ (d) $k = 7$

Figure 6: Results of the parameter study on a 100 × 100 × 100 Crawfis tornado dataset with $r = 0.01$ and $s = 1$ for varying k

further still improves the quality and reliability of the results, albeit decreasingly less.

In order to evaluate the relationship between the statistical complexity and the number of iterations, we computed

$$\delta_k = \frac{1}{N} \sum_{i=1}^{N} |C_k(\mathbf{x_i}, t) - C_{k+1}(\mathbf{x_i}, t)|, \tag{8}$$

where N is the number of grid points and $\mathbf{x_i} \in G$ is a grid point. As depicted in Figure 7 we can observe that the δ_k values decrease and approach zero when k grows. Thus when using our method there always is a tradeoff between the runtime of the algorithm and the robustness of the results. In practical applications, five to seven iterations should be sufficient as the quality of the results only increases marginally afterward.

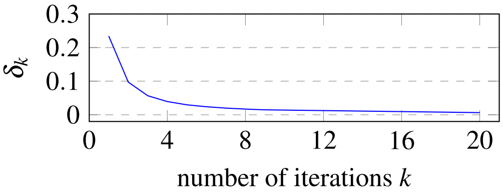


Figure 7: Relationship between δ_k and k

5.3 Performance

In this section, we measure the scalability of our method. The test was conducted on a Linux system with Ubuntu 20.04.3 LTS, 32 GB of RAM and a 32x Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz. Since the calculation of the statistical complexity is performed individually for every grid point, we can utilize OpenMP for parallelization. It is to note that our method is still not fully optimized. We tried to avoid unnecessary calculations but making the algorithm run on GPUs could introduce a drastic speedup as one of the most frequently used operations is a test for triangle intersections.

During our experiment, we measured the runtimes of our method for its application on Crawfis tornados defined on $n \times n \times n$ structured grids with $n = 50, 100, 150, 200$. We set the relative entropy threshold to 0.01, subdivided the icosahedron faces used for the binning one time, and let the algorithm run through three iterations. As shown in Figure 8 a linear increase in the number of grid points leads to a linear increase in runtime.

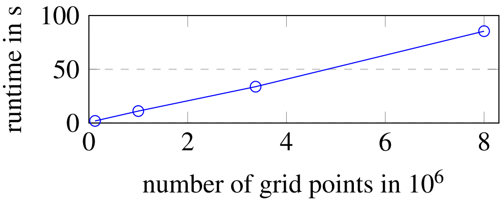


Figure 8: Relationship between runtime and grid points

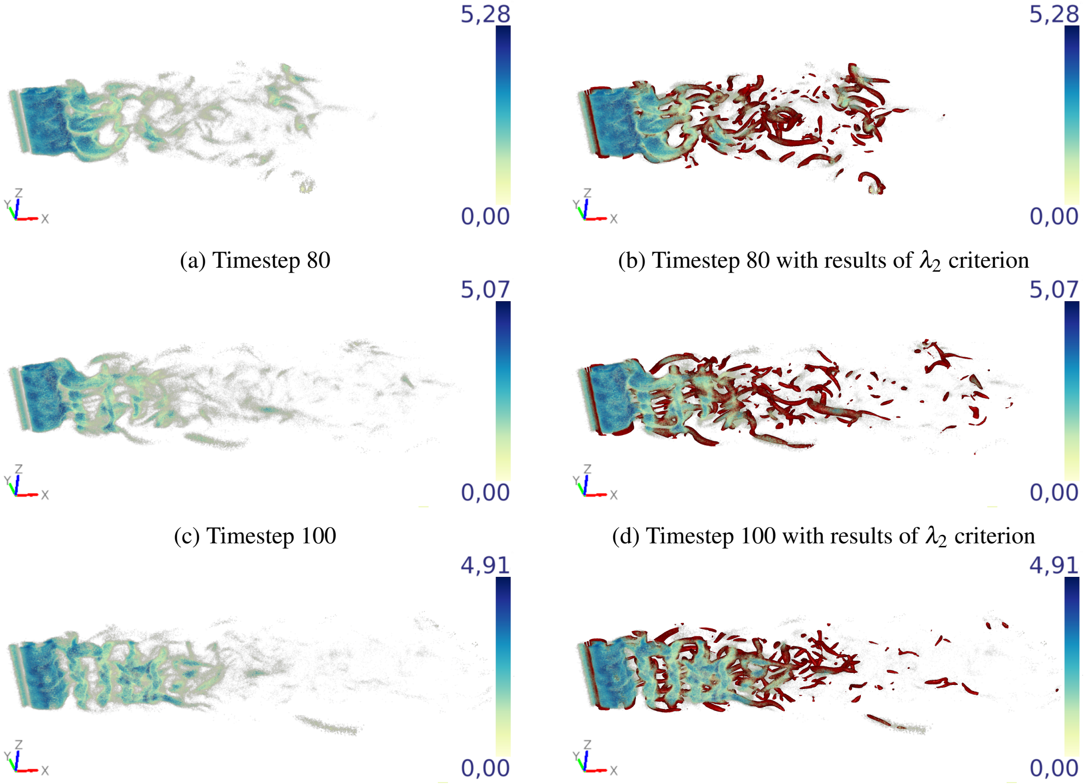
We repeated the the same tests for crawfis tornados defined on $n \times n \times n$ unstructured domains with $n = 50, 100$. The runtimes were increased by a factor of 6.32 and 7.88, respectively, compared to the structured grids.

6 APPLICATION TO VON KÁRMÁN VORTEX STREET

We applied our method to a turbulent DNS dataset of a von Kármán vortex street as it is a well-studied configuration. The dataset was simulated with the Gerris flow solver [Pop04] and made publicly available by the ETH Zürich [BRG19]. The dataset contains a constant flow that hits a half-cylinder, leading to vortex shedding. The flow is defined on a 640 × 240 × 80 grid and has a Reynolds number of 6400. The dataset contains 151 timesteps depicting the build-up of the flow. At around timestep 90, the flow starts to span the whole domain.

We computed the statistical complexities of three timesteps in the later stages of the flows development. The regions with the highest complexities can be found directly around the cylinder. Behind the cylinder, structures of high statistical complexity emerge in regular intervals on alternating sides. With advancing time, more of these structures form as the existing ones move with the general direction of the flow akin to the vortices of a von Kármán vortex street.

We also compared the results of our method with the vortices detected by the λ_2 vortex criterion. A clear relation is visible between the regions with the highest statistical complexity values and the λ_2 vortices in the first third of the field behind the cylinder. Further behind the cylinder, some vortices coincide with regions of middle to lower complexity values, whose magnitude



(a) Timestep 80

(b) Timestep 80 with results of λ_2 criterion

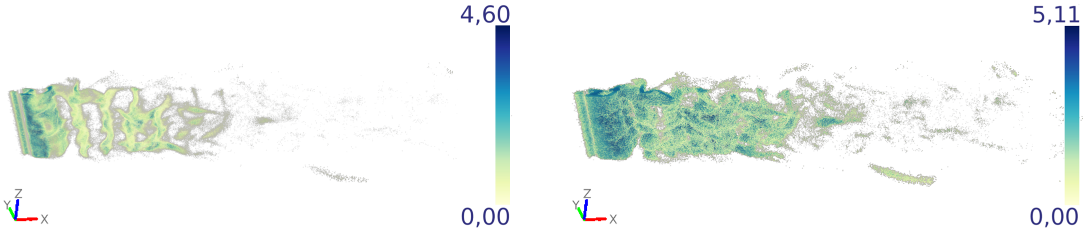
(c) Timestep 100

(d) Timestep 100 with results of λ_2 criterion

(e) Timestep 120

(f) Timestep 120 with results of λ_2 criterion

Figure 9: Visualization of the statistical complexities for $r = 0.01$, $s = 2$ and $k = 5$ of multiple timesteps of the von Kármán vortex street [BRG19, Pop04] compared with results of λ_2 vortex criterion [JH95] displayed as red isosurfaces of the value $\lambda_2 = -15$



(a) $r = 0.1$

(b) $r = 0.001$

Figure 10: Visualization of the statistical complexities for $s = 2$ and $k = 5$ of timestep 120 of the von Kármán vortex street [BRG19, Pop04] for different r

is probably diminished through them lying in areas of high turbulence. The statistical complexity, in general, behaves as expected and captures the behavior of the flow.

We also ran some test on how the statistical complexity behaves for different relative entropy thresholds r. We did not change the degree of subdivision s of the binning icosahedron as we deem 320 bins to be an appropriate number to capture the behavior of a turbulent flow. The number of algorithm iterations was set to $k = 5$ since the results of Section 5.2 indicate that increasing this number further has no significant influence on the quality of the visualization. As seen in Figure 10, changing r has the same influence on the statistical complexity values as we observed in Section 5.2. For high relative entropy thresholds r, the regions with the highest complexities shrink and finer structures are hardly visible anymore. Lower r-values lead to bigger regions of high complexity.

7 CONCLUSIONS

This paper presented an improvement of the statistical complexity calculation for three-dimensional flow fields by making the entropy computation more robust. We achieved this by presenting a new neighborhood definition applicable to any grid type, determining dynamic neighborhood sizes based on the concept of information gain, and introducing a novel binning approach to reduce overestimated complexity values and artifacts. By applying our method to analytical datasets, we evaluated our approach and showed that the statistical complexity is suited to identify critical points and other regions of interest. We have further shown that the direct visualization of the improved statistical complexity via volume rendering gives meaningful insights into the behavior of a flow for DNS data. Thus the proposed method provides an additional tool for the visual analysis of such data.

In the future, we want to apply our method to a more complex DNS dataset and evaluate the applicability of our method to the expert domain. In addition, we will develop a visualization technique suitable for the visualization of statistical complexity, taking into account the neighborhood sizes. This should enable perceiving regions of interest of different scales. Lastly, we will conduct research on the utilization of statistical complexity in different reference frames and especially investigate how to make our method translation invariant.

8 ACKNOWLEDGMENTS

This work was partially funded by the German Federal Ministry of Education and Research within the project Competence Center for Scalable Data Services and Solutions (ScaDS) Dresden/Leipzig (BMBF 01IS14014B) and by the Development Bank of Saxony (SAB) under project number 100335729.

9 REFERENCES

[ABMP14] A Arbona, C Bona, B Miñano, and A Plastino. Statistical complexity measures as telltale of relevant scales in emergent dynamics of spatial systems. *Physica A: Statistical Mechanics and its Applications*, 410:1–8, 2014.

[BCP+12] Andrea Brambilla, Robert Carnecky, Robert Peikert, Ivan Viola, and Helwig Hauser. Illustrative flow visualization: State of the art, trends and challenges. *Visibility-oriented Visualization Design for Flow Illustration*, 2012.

[BM20] Roxana Bujack and Ariane Middel. State of the art in flow visualization in the environmental sciences. *Environmental Earth Sciences*, 79(2):65, 2020.

[BPKB14] H. Bhatia, V. Pascucci, R. M. Kirby, and P.-T. Bremer. Extracting features from time-dependent vector fields using internal reference frames. In *Proceedings of the 16th Eurographics Conference on Visualization*, EuroVis '14, page 21–30, Goslar, DEU, 2014. Eurographics Association.

[BRG19] Irene Baeza Rojo and Tobias Günther. Vector field topology of time-dependent flows in a steady reference frame. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Scientific Visualization)*, 2019.

[CFV+16] Min Chen, Miquel Feixas, Ivan Viola, Anton Bardera, Han-Wei Shen, and Mateu Sbert. *Information theory tools for visualization*. CRC Press, 2016.

[Cra03] Roger Crawfis. Tornado data set generator, 2003.

[CY89] James P Crutchfield and Karl Young. Inferring statistical complexity. *Physical review letters*, 63(2):105, 1989.

[DCH88] Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *ACM Siggraph Computer Graphics*, 22(4):65–74, 1988.

[FI08] Shiho Furuya and Takayuki Itoh. A streamline selection technique for integrated scalar and vector visualization. *Vis Š08: IEEE Visualization Poster Session*, 2(4), 2008.

[GGT17] Tobias Günther, Markus Gross, and Holger Theisel. Generic objective vortices for flow visualization. *ACM Trans. Graph.*, 36(4):141:1–141:11, July 2017.

[Hal05] G. Haller. An objective definition of a vortex. *Journal of Fluid Mechanics*, 525:1–26, 2005.

[Hal15] George Haller. Lagrangian coherent structures. *Annual Review of Fluid Mechanics*, 47:137–162, 2015.

[HH89] J. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. *Computer*, 22(8):27–36, 1989.

[HH91] J.L. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991.

[HS11] George Haller and Themistoklis Sapsis. Lagrangian coherent structures and the smallest finite-time lyapunov exponent. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21(2):023115, 2011.

[HY00] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D: Nonlinear Phenomena*, 147(3–4):352–370, 2000.

[JBTS08] Heike Jänicke, Michael Böttinger, Xavier Tricoche, and Gerik Scheuermann. Automatic detection and visualization of distinctive structures in 3d unsteady multi-fields. In *Computer Graphics Forum*, volume 27, pages 767–774. Wiley Online Library, 2008.

[JH95] Jinhee Jeong and Fazle Hussain. On the identification of a vortex. *Journal of fluid mechanics*, 285:69–94, 1995.

[JS10] Heike Jänicke and Gerik Scheuermann. Towards automatic feature-based visualization. In *Dagstuhl Follow-Ups*, volume 1. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2010.

[JWM88] Hunt' JCR, A Wray, and P Moin. Eddies, stream, and convergence zones in turbulent flows. *Center for turbulence research report CTR-S88*, pages 193–208, 1988.

[JWSK07] Heike Janicke, Alexander Wiebel, Gerik Scheuermann, and Wolfgang Kollmann. Multifield visualization using local statistical complexity. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1384–1391, 2007.

[KL51] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[Leo06] Paul Leopardi. A partition of the unit sphere into regions of equal area and small diameter. *Electronic Transactions on Numerical Analysis*, 25(12):309–327, 2006.

[LHZP07] Robert S. Laramee, Helwig Hauser, Lingxiao Zhao, and Frits H. Post. Topology-based flow visualization, the state of the art. In Helwig Hauser, Hans Hagen, and Holger Theisel, editors, *Topology-based Methods in Visualization*, pages 1–19, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[LMSC11] Teng-Yok Lee, Oleg Mishchenko, Han-Wei Shen, and Roger Crawfis. View point evaluation and streamline filtering for flow visualization. In *2011 IEEE Pacific Visualization Symposium*, pages 83–90. IEEE, 2011.

[LRMC95] Ricardo Lopez-Ruiz, Héctor L Mancini, and Xavier Calbet. A statistical measure of complexity. *Physics letters A*, 209(5-6):321–326, 1995.

[MLP+09] Tony McLoughlin, Robert S. Laramee, Ronald Peikert, Frits H. Post, and Min Chen. Over Two Decades of Integration-Based, Geometric Flow Visualization. In M. Pauly and G. Greiner, editors, *Eurographics 2009 - State of the Art Reports*. The Eurographics Association, 2009.

[MWW+14] Jun Ma, James Walker, Chaoli Wang, Scott Kuhl, and Ching Kuang Shene. Flowtour: An automatic guide for exploring internal flow features. In *2014 IEEE Pacific Visualization Symposium*, pages 25–32. IEEE, 2014.

[NNF+20] Baldwin Nsonga, Martin Niemann, Jochen Fröhlich, Joachim Staib, Stefan Gumhold, and Gerik Scheuermann. Detection and visualization of splat and antisplat events in turbulent flows. *IEEE Transactions on Visualization and Computer Graphics*, 26(11):3147–3162, 2020.

[NSG+20] B. Nsonga, G. Scheuermann, S. Gumhold, J. Ventosa-Molina, D. Koschichow, and J. Fröhlich. Analysis of the near-wall flow in a turbine cascade by splat visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):719–728, 2020.

[NT12] Ikujiro Nonaka and Hirotaka Takeuchi. *Die Organisation des Wissens: Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen*. Campus Verlag, 2012.

[Pop04] S. Popinet. Free computational fluid dynamics. *ClusterWorld*, 2(6), 2004.

[RG20] Irene Baeza Rojo and Tobias Günther. Vector field topology of time-dependent flows in a steady reference frame. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):280–290, 2020.

[TLMF21] Silvio Tschisgale, Bastian Löhrer, Richard Meller, and Jochen Fröhlich. Large eddy simulation of the fluid–structure interaction in an abstracted aquatic canopy consisting of flexible blades. *Journal of Fluid Mechanics*, 916, 2021.

[TMWS12] Jun Tao, Jun Ma, Chaoli Wang, and Ching-Kuang Shene. A unified approach to streamline selection and viewpoint selection for 3d flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):393–406, 2012.

[WGS05] Alexander Wiebel, Christoph Garth, and Gerik Scheuermann. Localized Flow Analysis of 2D and 3D Vector Fields. In Ken Brodlie, David Duke, and Ken Joy, editors, *EUROVIS 2005: Eurographics / IEEE VGTC Symposium on Visualization*. The Eurographics Association, 2005.

[WGS07] Alexander Wiebel, Christoph Garth, and Gerik Scheuermann. Computation of localized flow for steady and unsteady vector fields and its applications. *IEEE transactions on visualization and computer graphics*, 13:641–51, 07 2007.

[WXWH13] Ziniu Wu, Yizhe Xu, Wenbin Wang, and Ruifeng Hu. Review of shock wave detection method in cfd post-processing. *Chinese Journal of Aeronautics*, 26(3):501–513, 2013.

[WYM08] Chaoli Wang, Hongfeng Yu, and Kwan-Liu Ma. Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547–1554, 2008.

[XLS10] Lijie Xu, Teng-Yok Lee, and Han-Wei Shen. An information-theoretic framework for flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1216–1224, 2010.

[ZH15] Liang Zhou and Charles D Hansen. A survey of colormaps in visualization. *IEEE transactions on visualization and computer graphics*, 22(8):2051–2069, 2015.

Efficient Point Cloud Skeletonization with Locally Adaptive L_1-Medial Projection

Stefan Lengauer, Peter Houska and Reinhold Preiner
Institute of Computer Graphics and Knowledge Visualization, Graz University of Technology
Inffeldgasse 16c, 8010 Graz, Austria
s.lengauer@cgv.tugraz.at; p.houska@cgv.tugraz.at; r.preiner@cgv.tugraz.at

ABSTRACT

3D line skeletons are simplistic representations of a shape's topology which are used for a wide variety of geometry-processing tasks, including shape recognition, retrieval, and reconstruction. Numerous methods have been proposed to generate a skeleton from a given 3D shape. While mesh-based methods can exploit existing knowledge about the shape's topology and orientation, point-based techniques often resort to precomputed per-point normals to ensure robustness. In contrast, previously proposed techniques for unprocessed point clouds either exhibit inferior robustness or require expensive operations, which in turn increases computation time. In this paper, we present a new and highly efficient skeletonization approach for raw point cloud data, which produces overall competitive results compared to previous work, while exhibiting much lower computation times. Our algorithm performs robustly in the face of noisy and fragmented inputs, as they are usually obtained from real-world 3D scans. We achieve this by first transferring the input point cloud into a Gaussian mixture model (GMM), obtaining a more compact representation of the surface. Our method then iteratively projects a small subset of the points into local L_1-medians, yielding a rough outline of the shape's skeleton. Finally, we present a new branch detection technique to obtain a coherent line skeleton from those projected points. We demonstrate the capabilities of our proposed method by extracting the line skeletons of a diverse selection of input shapes and evaluating their visual appearance as well as the efficiency compared to alternative state-of-the-art methods.

Keywords
point cloud, curve skeleton, Gaussian mixture, geometric computation

1 INTRODUCTION

Line skeletons – infinitely thin and centered representations of a shape's topology – have been a heavily researched topic since the original concept definition by Blum [Blu67] in 1967. This compact approximation captures the essence of the topological structure of even highly detailed 3D shapes, and acts as an efficient proxy for applications such as shape identification, classification and retrieval.

3D skeletonization approaches typically require a surface mesh or volumetric mesh as input and extract the line skeletons through shrinking, contracting, projecting or other geometry processing techniques. However, in some cases, e.g. scanning of real world objects, shapes are available only as raw point clouds. Obtaining line skeletons from such inputs is possible but entails additional challenges [OI92; CSM07; HLZ*09]. A skeletonization technique able to process such input data was proposed by Huang *et al.* [HLZ*09]. It builds on the *Locally Optimal Projection* (LOP) operator proposed by Lipman *et al.* [LCLT07], which allows for a parametrization-free resampling of the surface given by a raw input point cloud. Let $P = \{p_j\}_{j \in J} \subset \mathbb{R}^3$ be an unordered set of input points and $X^{(0)} = \{x_i^{(0)}\}_{i \in I} \subset \mathbb{R}^3$ an arbitrary set of projected points with $|X^{(0)}| \ll |P|$ and I, J denoting index sets. While $X^{(0)}$ can contain any points from $\mathbb{R}^3$, in practice faster converge can be achieved with the initialization $X^{(0)} \subset P$. The LOP operator tries to determine the set of optimally projected points X by satisfying

$$\underset{X}{\arg\min} \sum_{i \in I} \sum_{j \in J} ||x_i - p_j|| \theta(||x_i - p_j||) + R(X). \qquad (1)$$

Here, the first term describes the locally weighted L_1-median [Web09], driving the points in X towards local centers of P. The repulsion force $R(X)$ prevents projected points from accumulating in clusters. Huang *et al.* [HLZ*09] adopted this operator to account for non-uniform particle distributions with a locally adaptive density weighting, referred to as *Weighted LOP* (WLOP). To this end, they assign weights to each point p_j and x_i, based on the local point cloud density. Preiner *et al.* [PMA*14] show how to reformulate this operator to work on a compact mixture of anisotropic Gaussians $\mathcal{M}$, representing the input point cloud P. Their projection scheme is referred to as *Continuous LOP* (CLOP). They reformulate the attraction force accordingly, where $|\mathcal{M}| \ll |P|$, thereby greatly enhancing the efficiency of the projection.

https://www.doi.org/10.24132/CSRN.3201.6

While this family of projection operators was originally designed to reconstruct surfaces from noisy and incomplete point clouds, Huang *et al.* [HWC*13] showed that with minimal adaptions WLOP can also be used to project the surface points into local centers of the entire shape, thus constituting the basis for a line skeleton. They use a weighted *Principal Component Analysis* (PCA) to detect point connections and branches. The advantage of this approach is that it has few requirements regarding the quality of the input point set as it is very robust *w.r.t.* outliers. Also, neither a surface mesh nor vertex normals are required.

We present a novel method that combines this WLOP-based skeletonization with the much more efficient CLOP operator that operates on a Gaussian mixture representation of the input data. Moreover, we replace the point-based branch detection technique, proposed by Huang *et al.*, with a more robust probability-based approach. Finally, we show that by decoupling projection and branch detection, a further increase of computational efficiency can be achieved.

2 RELATED WORK

The topic of skeletonization is vast and multi-faceted. Several different strategies have emerged over the years, ranging from *shrinking ball methods* and *distance field methods* to *medial-surface-based methods* and *contraction methods*, each with a large number of publications. Among these methods, various approaches can be also be classified based on the type of input data they process, such as (watertight) surface meshes, voxel grids, or point clouds. In this paper, we primarily focus on skeletonization techniques that process point clouds as inputs. For a general discussion on skeletonization techniques, we refer the reader to a number of surveys. Cornea *et al.* [CSM07] give a broad overview of methods for obtaining curve skeletons. Sobiecki *et al.* [SYJT13] focus specifically on the comparison of contraction-based skeletionization methods, and they also look into the comparison of curve- and surface-skeletons operating on voxel shapes [SJT14]. The most recent survey from 2016 by Tagliasacchi *et al.* [TDS*16] constitutes an encompassing report on all types and aspects of skeletonization.

While the vast majority of approaches requires surface meshes as input, Ogniewicz and Ilg [OI92] show how line skeletons can be obtained from point clouds based on a Voronoi diagram of boundary points. A weakness of this approach is that it requires a uniform sampling of an intact input shape, which is not generally provided by many point clouds, especially when they are obtained by scanning real-world physical objects.

Sharf et al. [SLSK07] propose growing a watertight genus-0 mesh inside the input point cloud with multiple competing evolving fronts. While this approach manages to capture various degrees of detail and is able to cope with missing data, it is computationally expensive. Tagliasacchi *et al.* [TZC09] achieve robustness *w.r.t.* holes in the point cloud by inferring a generalized rotational symmetry axis (ROSA) from the points, which, however, requires normals provided along with the point positions as input. Cao *et al.* [CTO*10] show how skeletons can be obtained from point clouds by pairing a Laplacian-based contraction – as proposed by Au *et al.* [ATC*08] for surface meshes – with a local Delaunay triangulation.

3 GMM-BASED SKELETONIZATION

In this section we discuss our method and its main components in detail. Our method performs a continuous optimal projection (CLOP) (Sec 3.1) with dynamically increasing kernel radii (Sec 3.2). The subsequent branch detection (Sec. 3.3) takes the projected points as input and yields a connected 1D structure capturing the essential geometric shape of these points.

3.1 L_1-Medial Projection

Based on the condition given in Eqn. (1), Huang *et al.* [HLZ*09] use the following update rule for the projected points $x_i^{(k)} \in X^{(k)}$ at iteration k:

$$x_i^{(k+1)} = F_1(x_i^{(k)}, P) + \mu F_2(x_i^{(k)}, X_i^{(k)} \setminus \{x_i\}), \quad (2)$$

with the attraction force

$$F_1(x, P) = \sum_{j \in J} p_j \frac{\alpha_j}{\sum_{j' \in J} \alpha_{j'}} \quad (3)$$

and the repulsion force

$$F_2(x, X_i') = \sum_{i' \in I \setminus \{i\}} (x - x_{i'}) \frac{\beta_{i'}}{\sum_{i'' \in I \setminus \{i\}} \beta_{i''}}, \quad (4)$$

where

$$\alpha_j = \frac{\theta(||p_j - x||)}{||p_j - x||} \quad \text{and} \quad (5)$$

$$\beta_{i'} = \frac{\theta(||x - x_{i'}||)}{||x - x_{i'}||} \left| \frac{\partial \eta}{\partial r}(||x - x_{i'}||) \right|. \quad (6)$$

In Eqn. (2), μ governs the balance between the forces and should be within $[0, 0.5)$ for practical applications [HLZ*09; PMA*14]. $\theta(r) = e^{r^2/(h/4)^2}$ is a fast decaying, isotropic kernel function over support radius h. The projection scheme in this form corresponds to the LOP operator by Lipman *et al.*

In order to account for non-uniform densities in the input point cloud, the WLOP operator introduces additional local density weights

$$v_j = 1 + \sum_{j' \in J \setminus \{j\}} \theta(||p_j - p_{j'}||), \quad (7)$$

$$w_i^{(k)} = 1 + \sum_{i' \in I \setminus \{i\}} \theta(||x_i^{(k)} - x_{i'}^{(k)}), \quad (8)$$

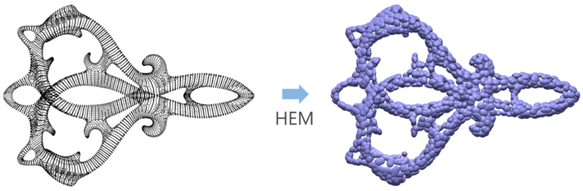


Figure 1: A point cloud is converted to a mixture of anisotropic Gaussians, visualized as blue ellipsoides denoting their 0.5 iso-variance.

acting on Eqn. (5) and Eqn. (6) as $\alpha_j = \alpha_j / v_j$ and $\beta_{i'}^{(k)} = \beta_{i'}^{(k)} w_i^{(k)}$. This adopted scheme results in an improved point regularity and further prevents the forming of point clusters.

With CLOP we approximate the input point set P by a much more compact mixture of anisotropic Gaussians $\mathcal{M} = \{w_j, \mu_j, \Sigma_j\}$, where w_j denotes the Gaussian's convex weights, μ_j their means, and Σ_j their anisotropic covariance matrices. Such a mixture can be computed efficiently by means of a regularized *hierarchical expectation maximization* [VL98; PMA*14]. Accordingly, Eqn. (3) is reformulated to exert the integral attraction force of the continuous density of the mixture $\mathcal{M}$,

$$\mathcal{F}_1(q, \mathcal{M}) = \frac{\sum_j w_j \int_{\mathbb{R}^3} x\, g(x|\mu_j, \Sigma_j)\ \alpha(x)\ dx}{\sum_j w_j \int_{\mathbb{R}^3} g(x|\mu_j, \Sigma_j)\ \alpha(x)\ dx}, \tag{9}$$

which can be efficiently evaluated in closed form [PMA*14].

3.2 Adaptive Kernel-Growth

Huang *et al.* [HWC*13] propose an interleaved WLOP projection and branch detection scheme in order to obtain a L_1 medial projection of points. To this end, a constantly increasing kernel size h is used to cover features of different diametric extents, while avoiding the degeneration of smaller features in the presence of too large h. During the branch detection step, subsets of the projected points that are found to exhibit a branch-like structure are marked as *fixed* and excluded from the further projection iterations in order to prevent their degeneration.

Opposed to an interleaved execution, we found that the efficiency as well as the robustness of the skeletonization process could be improved by performing the projection and branch detection steps consecutively. That is, we perform CLOP projection until *all* projected points X have converged towards their local L_1 median, and only then, branch detection is performed. As a consequence, we also have to adjust the kernel-growth policy accordingly: Since no points are *fixed* until projection is finished, a globally increasing kernel would destroy the valid medial structures which form at a small h in areas of delicate features. To resolve this issue, we employ an increasing but locally adaptive kernel size whose growth rate is governed by the local anisotropy of the shape.

Starting from an initial kernel size of $h^{(0)} = 2bbd / \sqrt[3]{|P|}$ (where *bbd* denotes the bounding box diagonal of P) as proposed by Huang *et al.*, the kernel $h_i^{(k)}$ for point x_i at iteration k is given by

$$h_i^{(k)} = h_i^{(k-1)} + \Delta h\, v_i^{(k)}, \tag{10}$$

with $\Delta h = h^{(0)}/2$. The local growth factor $v_i^{(k)} \in [0,1)$ ensures that areas with highly isotropic neighborhood experience a comparably large kernel growth, while regions of highly anisotropic shape experience almost none. The local growth factor is given by

$$v_i^{(k)} = \begin{cases} 1 & \text{if } n_i^{(k)} \le 2, \\ 1 - \frac{n_i^{(k)}-2}{n_i^{(k)}-1}\sigma_i^{(k)} & \text{otherwise.} \end{cases} \tag{11}$$

Here, the measure for local anisotropy $\sigma_i^{(k)} \in [0,1]$ is given by

$$\sigma_i^{(k)} = \frac{\lambda_{i_2}^{(k)}}{\lambda_{i_0}^{(k)} + \lambda_{i_1}^{(k)} + \lambda_{i_2}^{(k)}}, \tag{12}$$

where $\lambda_{i_0}^{(k)} \le \lambda_{i_1}^{(k)} \le \lambda_{i_2}^{(k)}$ are the real-valued eigenvalues of the weighted covariance matrix

$$C_i^{(k)} = \sum_{i' \in I \setminus \{i\}} \theta(||v_{i,i'}||)\, v_{i,i'} v_{i,i'}^{\top}, \quad v_{i,i'} = x_i^{(k)} - x_{i'}^{(k)}. \tag{13}$$

Note that $\sigma_i^{(k)}$ is scaled relative to the size of x_i's local neighborhood $n_i^{(k)} = |\{i' \in I \setminus \{i\} : ||x_i^{(k)} - x_{i'}^{(k)}|| < h_i^{(k)}\}|$. This counteracts a known downside of the applied weighted PCA that very sparse neighborhoods could exhibit a very high anisotropy, thus preventing any kernel growth and point projection in the first place.

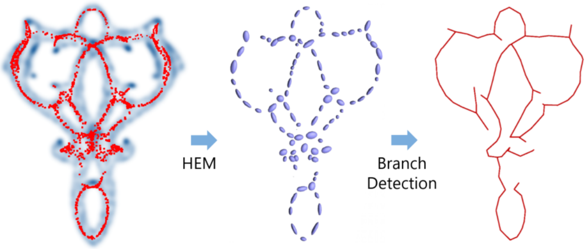


Figure 2: Left: density of $\mathcal{M}$ (blue) and projected skeleton points X (red). Middle: hierarchical EM on X results in a skeleton mixture $\mathcal{M}_{\mathcal{X}}$. Right: Skeleton after probabilistic branch detection.

3.3 Branch Detection

The iterative CLOP projection (Sec. 3.1) results in a regularized point set X in the local L_1-medians of the input shape. Although X resembles the resulting skeleton, the points do not convey any connectivity or branch information. Thus, in a final processing step we extract a set of connected straight line segments constituting the branches of the shape's skeleton.

We circumvent the instabilities we encountered with point-based branch detection approaches [HWC*13] by extracting branches from a probabilistic approximation of the projected points. This way, branch detection is more robust *w.r.t.* outliers and areas with high point density. Similar to the processing of input points P in Section 3.1, we use a regularized hierarchical expectation maximization [PMA*14] to compute a Gaussian mixture $\mathcal{M}_{\mathcal{X}} = \{w_i, (\mu_i, \Sigma_i)\}$ of the final projected points X (Fig. 2, middle). The mean points $\{\mu_i\}$ of the resulting Gaussians are then iteratively connected such that the angle α between adjacent segments (Fig. 3b) does not fall below $\alpha_{min} = \pi/2$, as we assume that the points within a branch are well aligned. Let $M = (d_M(i,j)) \in \mathbb{R}^{|\mathcal{M}|\times|\mathcal{M}|}$, $i,j = 1..|\mathcal{M}|$ denote the matrix of symmetric Mahalanobis distances [Mah36] $d_M(i,j) = \min\{\tilde{d}_M(\mu_i,\mu_j,\Sigma_i), \tilde{d}_M(\mu_j,\mu_i,\Sigma_j)\}$, with $\tilde{d}_M(x_1,x_2,\Sigma) = \sqrt{(x_1-x_2)^\top\Sigma^{-1}(x_1-x_2)}$, and let $E = (||\mu_i - \mu_j||)$ denote the equal-sized matrix of Euclidean distances between components' means. Our greedy branch detection method employs these distances as outlined in Algorithm 1, with the minimum number of components per branch $b_{min} = 5$, and the maximum Euclidean distance between components $d_{max} = 0.1\%$ of the bounding box diagonal.

As this approach can result in multiple disjoint branches, we complete this stage by merging these segments to a single connected set of branches (Fig. 3c). To this end, we perform a connected components analysis to identify unconnected clusters. Branches are merged pairwise by inserting a link between the two points from the respective clusters with the lowest Mahalanobis distance d_M. This merging is conducted iteratively until only one cluster remains, which constitutes the final skeleton (Fig. 2, right).

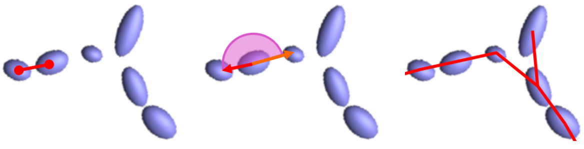

(a) (b) (c)

Figure 3: Starting from a seed pair of mixture components with minimal distance (a), an angle criterion (b) is used to grow the branch in both directions. Finally all branches are merged (c) to obtain a fully connected line skeleton.

4 RESULTS

We compare our method to five alternative skeletonization methods (Sec. 4.1), both mesh-based and point-based approaches, and evaluate the obtained results regarding their visual appearance and capability to capture the essence of a shape (Sec. 4.2), their computational efficiency (Sec. 4.3) and their robustness *w.r.t* noise and outliers (Sec. 4.4).

Algorithm 1 Branch detection based on pair-wise Mahalanobis distances M and Euclidean distances E as well as governing thresholds α_{min}, d_{max} and b_{min}.

```
function BRANCHDETECTION($M$, $E$, $\alpha_{min}$, $d_{max}$, $b_{min}$)
    $B \leftarrow \{\}$                                        ▷ The set of branches
    $A \leftarrow 1..|M_{0*}|$                                 ▷ The index set
    $v \leftarrow \{\}$                                        ▷ The blacklist of used points
    $(\hat{i},\hat{j}) \leftarrow \arg\min_{i\in A, j\in A} M_{ij}$
    while $|A| \geq b_{min}$ and $E_{\hat{i}\hat{j}} < d_{max}$ do
        $b \leftarrow \langle \hat{i},\hat{j} \rangle$          ▷ Init new branch
        $prev \leftarrow \arg\min_{i\in A\setminus b}(M_{ib_0})$
        while $\angle(\overrightarrow{b_0b_1},\overrightarrow{b_0prev}) > \alpha_{min}$ do
            $b \leftarrow \langle prev, b \rangle$             ▷ Grow head
            $prev \leftarrow \arg\min_{i\in A\setminus b}(M_{ib_0})$
        end while
        $next \leftarrow \arg\min_{i\in A\setminus b}(M_{ib_{|b|-1}})$
        while $\angle(\overrightarrow{b_{|b|-1}b_{|b|-2}},\overrightarrow{b_{|b|-1}next}) > \alpha_{min}$ do
            $b \leftarrow \langle b, next \rangle$             ▷ Grow tail
            $next \leftarrow \arg\min_{i\in A\setminus b}(M_{ib_{|b|-1}})$
        end while
        if $|b| \geq b_{min}$ then
            $B \leftarrow B \cup \{b\}$                        ▷ Append to branches
        end if
        $A \leftarrow A \setminus b$                           ▷ Update index set
        $(\hat{i},\hat{j}) \leftarrow \arg\min_{i\in A, j\in A} M_{ij}$   ▷ Update start
    end while
    return $B$
end function
```

4.1 Reference Methods

We compare our method to the original LOP-based approach (L1-WLOP) by Huang *et al.* [HWC*13], the ROSA approach by Tagliasacchi *et al.* [TZC09], and the Laplacian-based contraction of point clouds (LapCon10) by Cao *et al.* [CTO*10]. The former is given as a C++ implementation, while the latter two are available as MATLAB scripts. Besides these point-based methods we additionally compare our method to several mesh-based approaches (Wave Fronts, Geometry Contraction, TEASAR and Tangent Ball). Those are provided within a Python library[1] by Au *et al.* [ATC*08], which we also employ to preprocess the inputs and fix common mesh deficiencies, like duplicate or unreferenced vertices, degenerated faces and the like.

Wave Fronts

This method requires a surface mesh as input. Starting from a randomly selected seed vertex, a wave is prop-

[1] https://navis-org.github.io/skeletor/

agated across the mesh based on a given *n*-ring neighborhood, where *n* defines the propagation step. Vertices that are hit by the wave at the same step are considered rings and are subsequently collapsed to their common center. This approach is particularly well-suited and highly efficient for tubular structures.

Geometry Contraction

Au *et al.* [ATC*08] describe a skeletonization method based on an iterative geometric contraction (henceforth referred to as *LapCon08VC*) of the surface mesh. To this end, the mesh is subjected to an implicit Laplacian smoothing [SCL*04] until the mesh converges to a zero-volume skeleton-like shape. The remaining surface topology is then transformed to a one dimensional curve using a connectivity surgery process. Contraction weights control the efficiency of contraction, while attraction weights dynamically adapt to the local neighborhood. We used the contraction parameter $s_L = 2.0$ recommended by the authors. For inputs where these result in numerical instabilities, a smaller value was used. We also applied a sufficiently large convergence threshold $\varepsilon_{vol} = 0.1$ which lets the contraction stop if the contracted mesh is reduced to less than 10% of its original volume. To obtain a line skeleton from the contracted mesh, a subsequent vertex clustering algorithm is applied which joins vertices within a maximum geodesic distance of $s_{d_{max}}$. As the value of this parameter recommended by the authors resulted in instabilities on several of our inputs, we have increased this parameter for those cases.

Besides vertex clustering, Au *et al.* propose a second, alternative connectivity surgery method (*LapCon08EC*). Here, edges are iteratively collapsed in a greedy fashion based on a cost function that follows established mesh geometry preservation metrics [GH97]. An advantage of this approach is that it can be directly applied without the computational expensive prior contraction of the input (as we do in our experiments), while on the downside, it is more sensitive to the scale of the input.

TEASAR

The TEASAR algorithm by Sato *et al.* [SBB*00] starts from a randomly determined root voxel V_0 of a volumetric input mesh, and determines the voxel that maximizes the shortest-path length to V_0, invalidating all voxels within a given distance along its path. Next, the second longest shortest-path between still valid voxels is picked up repeatedly until all voxels are invalidated. This approach is well suited for shapes with an underlying tree structure and tubular sections like vessels, rib cages and the like. Although the original approach is conceptualized for volumetric meshes, the only available code is an adaption by Dorkenwald *et al.* [DS22], operating on surface meshes. Note however, that this adaption produces skeletons that lie on the surface of the input shape.

Tangent Ball

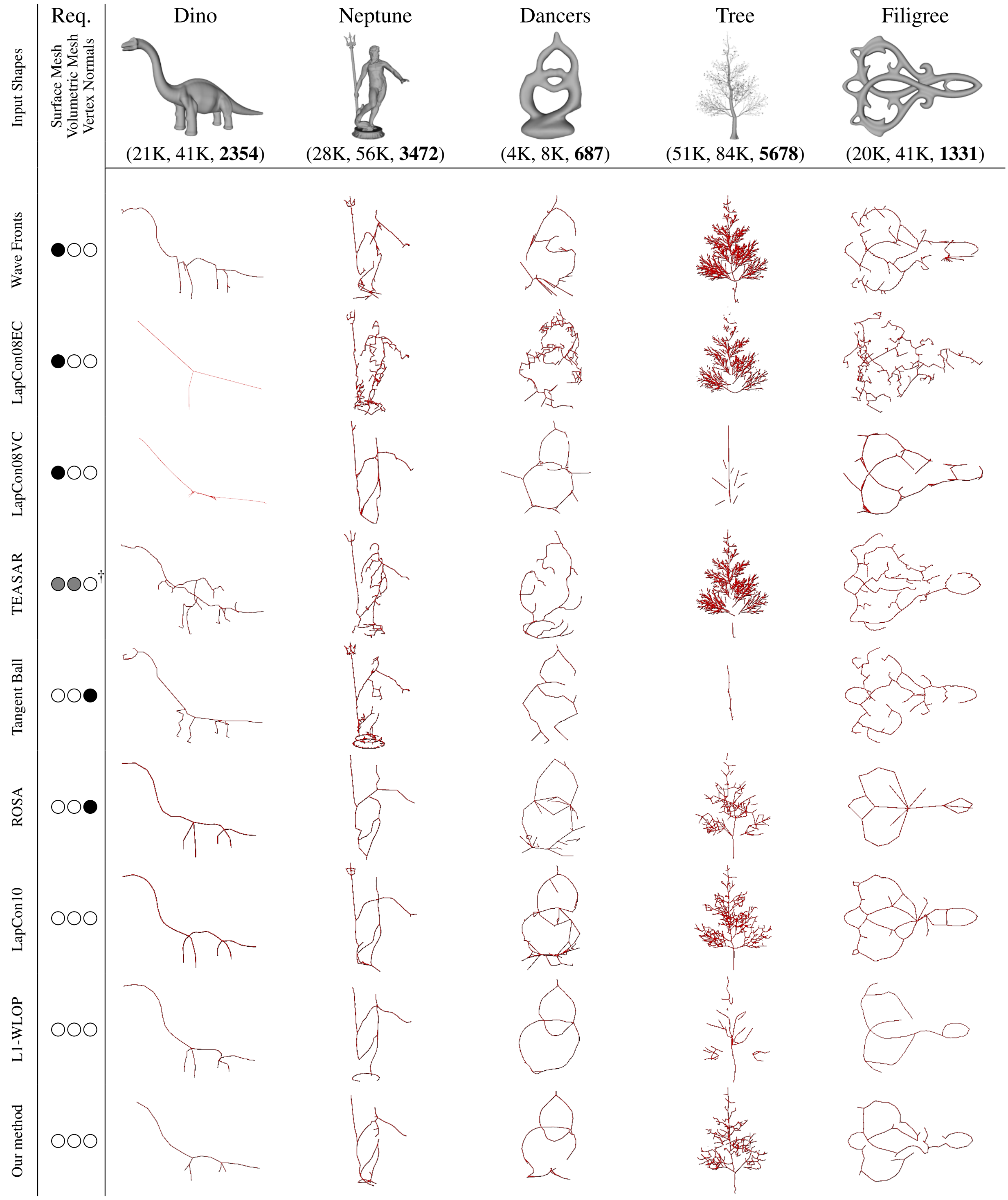
With their medial axis point approximation method (henceforth referred to by *Tangent Ball*) Ma *et al.* [MBC12] leverage vertex normals to determine the line skeleton. A set of randomly determined seed points are used to generate spheres, whose center points are located along their inverse vertex normal axis, such that the respective seed points rest on their sphere surface. The spheres' radii are decreased iteratively via nearest neighbor queries until they contain no other sampling point than their respective seed point. The authors show that for an infinite number of seed points, the sphere centers converge towards the true medial axis. The method is highly sensitive *w.r.t.* the quality of vertex normals and works generally well for smooth surfaces, while errors in the mesh and noisy surfaces pose a limitation.

4.2 Visual Comparison

In order to conduct a qualitative evaluation of our skeletonization approach, we compiled a diverse set of input shapes, posing different challenges to skeletonization: *(i)* a *Dino* shape which is a simple standard model of a creature with a finite number of extremities, *(ii)* a statue of *Neptune*, a humanoid figure with several high-frequency structures like fingers, hair and a trident, *(iii)* an abstract smooth model of *Dancers*, *(iv)* a *Tree* model comprising a large number of thin details, and *(v)* a topologically complex *Filigree* model of genus 10. The inputs, as well as their number of vertices, faces and mixtures, are presented in Fig. 4, top row. All models have been skeletonized using the reference skeletonization methods listed in Sec. 4.1.

For the *Dino* shape (Fig. 4, first column) the Wave Fronts, TEASAR, Tangent Ball, ROSA, LapCon10, L1-WLOP, and our method result in plausible line skeletons with clearly distinguishable extremities like legs, tail and neck. The LapCon08EC and LapCon08VC approaches lead to an oversimplification.

For the *Neptune* model (Fig. 4, second column) all methods yield plausible results. However, in contrast to the Dino model, the LapCon08EC skeleton exhibits various superfluous branches. The same can be observed for the result generated by the TEASAR method in the torso region of the model. Note that LapCon08VC, ROSA, LapCon10, L1-WLOP and our method only partially preserve the thin tip of the trident.



† The original concept requires a volumentric mesh, but the given results were obtained with an adaptation for surface meshes.

Figure 4: Qualitative comparison of our method to eight related approaches using five diverse input shapes. The numbers in the table header denote the models' number of vertices, faces and mixture components after conversion to a GMM. Filled circles on the left indicate the type of input data and attributes required by a specific approach.

The *Dancers* model (Fig. 4, third column) poses a serious challenge to all of the applied algorithms, as none of them are able to capture the genus of the input shape within their resulting skeleton. Nonetheless, all results appear to be plausible, and one could argue that the skeleton calculated by L1-WLOP comes closest to the desired result.

The *Tree* shape (Fig. 4, fourth column) yields very diverse results. While the Wave Fronts, LapCon08EC and TEASAR methods obtain an overly detailed skeleton with a branch for every leaf, the LapCon08VC, Tangent Ball, and L1-WLOP methods return an oversimplified skeleton. ROSA, LapCon10 and our method provide a good trade off and are able to capture the stem as well as the thicker tree branches. Note that the input mesh is comprised of two intersecting but unconnected components (lower stem and tree crown). LapCon10 and our method are the only ones which are able to return a single connected skeleton component for this input.

The *Filigree* model (Fig. 4, fifth column) also exhibits tubular structures of higher genus. Similar to the Dancers model, none of the algorithms is able to capture the shape's essential structure accurately, and the achieved skeleton quality of the individual algorithms is comparable to the skeletons calculated from the Dancers model.

4.3 Efficiency

Table 1 lists the skeleton computation times for all presented methods and models. All computations were performed on commodity hardware. Note that our method was implemented in C++, while the reference methods were provided as either Python scripts, Matlab scripts or C++ implementations.

Table 1: Computation times for different models and approaches in seconds.

Model	Dino	Neptune	Dancers	Tree	Filigree
Wave Fronts	0.16	0.22	0.04	1.25	0.21
LapCon08EC	42.28	80.23	4.32	173.12	52.62
LapCon08VC	100.13	207.93	10.49	226.36	123.01
| Contraction[+]	99.55	206.96	10.32	221.06	122.32
| Clustering	0.59	0.97	0.17	5.30	0.69
TEASAR	0.45	1051.85	0.66	1.18	0.99
Tangent Ball	39.70	7.50	0.42	6.18	4.17
ROSA	5101.80	5479.90	7308.00	2236.60	411.29
LapCon10	195.00	525.00	1168.00	977.00	252.00
L1-WLOP	40.00	50.00	161.00	51.00	12.00
Our Method	8.37	13.54	12.98	17.63	3.89
| Init[†]	1.93	3.06	5.99	7.55	1.26
| HEM[†]	0.54	1.05	5.41	2.11	0.69
| Projection	5.67	9.17	1.49	5.93	1.81
| Branching[‡]	0.23	0.26	0.09	2.05	0.12

[†] Mixture computation of the input point set.
[‡] Mixture computation of the projected points plus branch detection.
[+] Times can vary significantly, depending on parametrization.

From Table 1 we can observe a correlation of efficiency between model sizes and computation times. In particular, the skeletonization of the *Tree* model takes significantly longer with almost all approaches. We can also see that the computation times with different methods vary greatly. LapCon08VC, Tangent Ball and LapCon10 are generally slower, with computation times up to a few minutes. In general, the ROSA technique exhibited the slowest performance, with timings far outside the comparable range for most of the models. Note that the bottleneck of the LapCon08VC method is given by the contraction step, whose efficiency is very sensitive *w.r.t.* the governing parameters. The authors state that these techniques could be orders of magnitude faster with optimal parameterization [ATC*08], which we aimed to find manually to the best of our means. In contrast, the Wave Fronts and TEASAR methods exhibit very short computation times. As an exception, the skeletonization of the Neptune model with TEASAR took more than 17 minutes on our reference platform.

Our method aligns itself in-between the fast and the slow reference methods with computation times settled around several seconds. As it can be seen from the performance break-down, our computation time is mostly composed of the computation of a Gaussian mixture of input points, necessary for CLOP (Sec. 3.1) and the iterative projection (Sec. 3.2). Interestingly, we can observe that the computation time with our approach does not significantly increase with increasing model size and complexity (c.f. Tree model), as the number of mixture components does not increase equally. For a proper interpretation of the presented timings, note that our method receives a raw non-parameterized point cloud as input, while most reference methods receive a fully connected surface mesh or point clouds with preprocessed vertex normals. If the actual input of those methods were given by raw points cloud as well, additional computation times for normal and surface reconstruction would have to be factored in. These timings were omitted in our comparison, since the input shapes were already given as surface meshes and reconstruction timings are themselves known to depend heavily on the applied method and parameters.

4.4 Stability

In most practical applications of line skeletonization it is generally required that similar input shapes result in similar skeletons. However, it has been shown that even very small perturbations on the surface of an input shape can lead to a substantially different skeleton [TDS*16]. This phenomenon has been referred to as *skeletal noise* [RVT08] or *spurious points* [SBTZ02]. Since 3D models captured from real world objects typically suffer from a multitude of different errors [BLN*13], the robustness of a skeletonization algorithm *w.r.t.* surface perturbations is a major quality characteristic.

We analyze the stability of our method by looking at the Filigree model, as it exhibits a multigenus geometry with structures and features of varying sizes. We subject the model to different levels of noise in order to mimic the artefacts present on models obtained by scanning real-world physical objects. In terms of noise, we apply white noise following the Gaussian distribution $\mathcal{N}(0,\sigma^2)$ to all of the vertices. Different magnitudes σ^2 of noise, 0.002, 0.005, and 0.01 times the object's bounding box diagonal, have been investigated, in the following referred to as $\mathcal{N}_{0.002}$, $\mathcal{N}_{0.005}$ and $\mathcal{N}_{0.01}$. The resulting noisy inputs are shown in the top row of Fig. 5.

Rows 2 to 7 of Fig. 5 present the skeletons resulting from our method as well as the reference methods

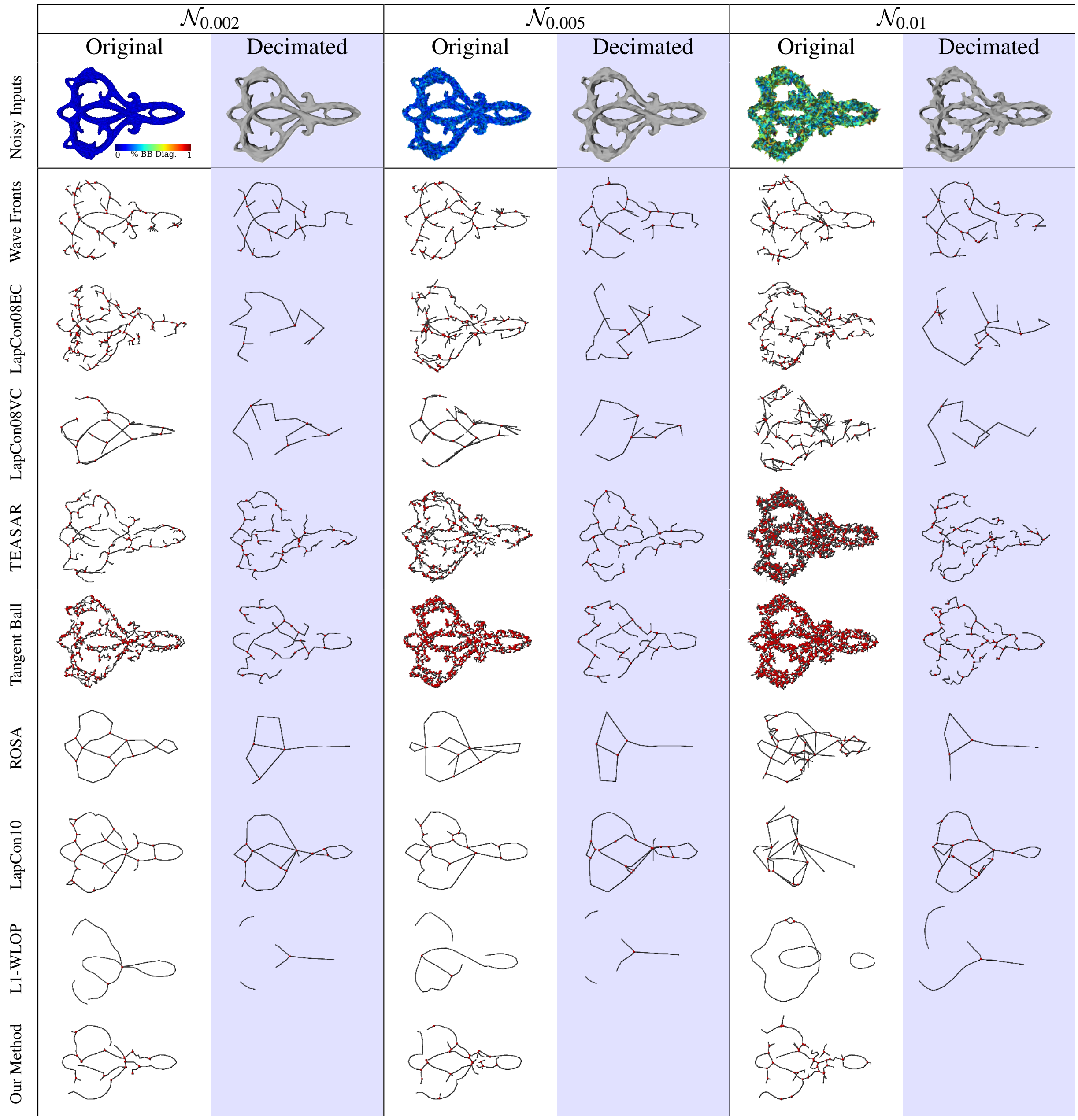


Figure 5: Top: Filigree model subjected to various levels of noise and (shaded in blue) additional decimation step. Rows 2-7: Skeletonization with different methods based on the respective noisy (and decimated) inputs. The red dots in the line skeletons mark junction points.

(Sec. 4.1), based on the different noised inputs. It is immediately visible that the results degenerate with increasing levels of noise, independent from the applied method. Moreover, we can observe clear differences in the quality of the approaches. The LapCon08EC, TEASAR and especially the Tangent Ball methods exhibit a major increase in skeletal noise, even with visually inconspicuous surface perturbations. The skeletons obtained with the Wave Fronts and Iterative Contraction methods do not significantly differ from the results obtained with the noiseless model (Fig. 4). With our method no significant deterioration is apparent at the lower two noise levels $\mathcal{N}_{0.002}$ and $\mathcal{N}_{0.005}$, and only at level $\mathcal{N}_{0.01}$ slight perturbations become visible.

Additionally to this visual comparison, we provide a quantitative evaluation of robustness. To this end, we count the number of junction points (points where branches are joined) of the resulting skeletons as an indicator for skeletal noise. For the *Filigree* model we expect around 20 junction points as visualized in Fig. 6, left. The figure shows the number of junction points resulting from different algorithms and noise levels. The comparably good performance of our method is attributed to the fact that the Gaussian mixtures,

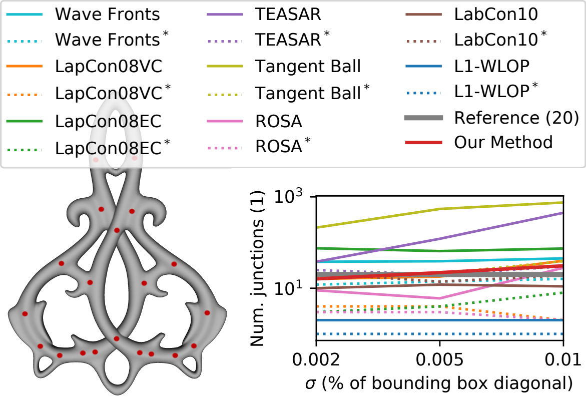


Figure 6: Number of junction points obtained with different approaches and different levels of additive noise. Algorithms marked with an asterisk are based on the decimated inputs.

which the input points are converted to (Sec. 3.1), allow to automatically model the inherent noise level of the data, thereby providing an implicit pre-filtering of the data. To investigate the impact of pre-filtering on the results obtained with the comparison methods, we apply a variant of the *Quadratic Edge Collapse Decimation* by Garland *et al.* [GH98] to the noisy inputs. We choose the parameters such that the resulting number of vertices is equal to the number of mixture components, which do not vary greatly across noise levels. The decimated models can be seen in Fig. 5, top row, shaded in blue. The results for the different methods based on these decimated inputs are given in Fig. 5, rows 2-7. While results became worse with the LapCon08EC and Iterative Contraction skeletonizations, an improvement can be clearly observed for the Wave Fronts, TEASAR and Tangent Ball approaches, where the skeletons obtained with the decimated noisy input is even better than with the unperturbed original input (Fig. 4).

5 LIMITATIONS AND FUTURE WORK

Like many other techniques, the results of the proposed skeletonization approach can be sensitive *w.r.t.* governing parameters. That is, empirically determined parameters are required in three separate steps of our pipeline: *(i)* the computation of the Gaussian mixture for CLOP and balancing of attraction and repulsion forces (Sec. 3.1), *(ii)* the speed of the adaptive kernel growth (Sec. 3.2), and *(iii)* the thresholds governing the branch detection (Sec. 3.3). Some of these parameters could possibly be defined over properties intrinsic to the input point cloud (diameter, density, feature sizes) which can be part of future work.

The PCA-driven kernel growth (Sec. 3.2) and repulsion force (Eqn. (2)) face typical issues of PCA, which are also discussed in Huang *et al.* [HLZ*09]: erroneous directivity estimates near thick point clouds and unwanted propagation between particles at close-by surfaces. The latter can be observed at the trident of the Neptune model (Fig. 4) where the relative proximity of the three parallel prongs leads to their collapse to a single tubular structure. This poses a severe disadvantage compared to mesh-based approaches, which can rely on the surface topology to avoid the merging of close-by surfaces.

Another aspect not investigated is the determination of the appropriate number of projection steps necessary for a successful projection for all shape granularities. 20 to 30 iterations turned out to be a good choice for all inputs. Introducing an automatic halting criterion would increase the flexibility of our method. A naive approach is to stop the projection if the overall points movement $m^{(k)} = \sum_{i \in I} ||x_i^{(k+1)} - x_i^{(k)}||$, falls below a certain threshold. In practice however, m does not gradually decrease with k, but instead can level off and run into a local minimum, or even continue to decrease after two surface branches merge. Finding a simple yet robust halting criterion in the face of these complex processes is thus still open for further investigation.

6 CONCLUSION

We present an efficient projection-based point cloud skeletonization approach. To this end, we marry the concept of the WLOP-based skeletonization by Huang *et al.* [HWC*13] with the CLOP operator [PMA*14]. In contrast to Huang *et al.*, we grow projection kernels for each point individually, based on a local anisotropy measure, and conduct the branch detection after the projection instead of interleaved. We present a probabilistic branch detection approach for the projected point set, which is more robust *w.r.t.* outliers. While point-based approaches generally seem to be inferior to surface or even volume-based approaches – as mesh topologies incorporate much vital information – the strength of our approach is that it does not depend on high quality or manifold input data, and even performs robustly in the face of noisy and incomplete data. Hence, we think that our approach is a reasonable alternative in cases where just the raw point cloud data is available, e.g. in the case of 3D scans if no surface reconstruction is conducted. We have demonstrated the feasibility of our method by different experiments conducted on a wide variety of 3D shapes, and analysed limitations and problem cases to indicate directions for future work.

REFERENCES

[ATC*08] Au, Oscar Kin-Chung, Tai, Chiew-Lan, Chu, Hung-Kuo, et al. “Skeleton Extraction by Mesh Contraction”. *ACM Trans. Graph.* 27.3 (2008) 2, 4, 5, 7.

[BLN*13] Berger, Matthew, Levine, Joshua A., Nonato, Luis Gustavo, et al. "A Benchmark for Surface Reconstruction". *ACM Trans. Graph.* 32.2 (Apr. 2013). doi: 10 . 1145 / 2451236.2451246 7.

[Blu67] Blum, Harry. *Models for the Perception of Speech and Visual Form*. MIT Press, 1967, 362–380 1.

[CSM07] Cornea, Nicu D., Silver, Deborah, and Min, Patrick. "Curve-skeleton properties, applications, and algorithms". *IEEE Transactions on visualization and computer graphics* 13.3 (2007), 530 1, 2.

[CTO*10] Cao, Junjie, Tagliasacchi, Andrea, Olson, Matt, et al. "Point cloud skeletons via laplacian based contraction". *2010 Shape Modeling International Conference*. IEEE. 2010, 187–197 2, 4.

[DS22] Dorkenwald, Sven and Schneider-Mizell, Casey. *MeshParty*. https : / / github . com / sdorkenw / MeshParty. 2022 5.

[GH97] Garland, Michael and Heckbert, Paul S. "Surface simplification using quadric error metrics". *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 1997, 209–216 5.

[GH98] Garland, Michael and Heckbert, Paul S. "Simplifying Surfaces with Color and Texture Using Quadric Error Metrics". *Proceedings of the Conference on Visualization '98*. VIS '98. Research Triangle Park, North Carolina, USA: IEEE Computer Society Press, 1998, 263–269. isbn: 1581131062 9.

[HLZ*09] Huang, Hui, Li, Dan, Zhang, Hao, et al. "Consolidation of Unorganized Point Clouds for Surface Reconstruction". *ACM Trans. Graph.* 28.5 (Dec. 2009), 1–7. doi: 10 . 1145 / 1618452.1618522 1, 2, 9.

[HWC*13] Huang, Hui, Wu, Shihao, Cohen-Or, Daniel, et al. "L1-Medial Skeleton of Point Cloud". *ACM Trans. Graph.* 32.4 (July 2013). doi: 10.1145/2461912.2461913 2–4, 9.

[LCLT07] Lipman, Yaron, Cohen-Or, Daniel, Levin, David, and Tal-Ezer, Hillel. "Parameterization-Free Projection for Geometry Reconstruction". *ACM Trans. Graph.* 26.3 (July 2007), 22–es. doi: 10.1145/1276377.1276405 1.

[Mah36] Mahalanobis, Prasanta Chandra. "On the generalized distance in statistics". National Institute of Science of India. 1936 4.

[MBC12] Ma, Jaehwan, Bae, Sang Won, and Choi, Sunghee. "3D medial axis point approximation using nearest neighbors and the normal field". *The Visual Computer* 28.1 (2012), 7–19 5.

[OI92] Ogniewicz, Robert L and Ilg, Markus. "Voronoi skeletons: theory and applications." *CVPR*. Vol. 92. 1992, 63–69 1, 2.

[PMA*14] Preiner, Reinhold, Mattausch, Oliver, Arikan, Murat, et al. "Continuous Projection for Fast L_1 Reconstruction". *ACM Trans. Graph.* 33.4 (July 2014). doi: 10 . 1145 / 2601097.2601172 1–4, 9.

[RVT08] Reniers, Dennie, Van Wijk, Jarke, and Telea, Alexandru. "Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure". *IEEE Transactions on Visualization and Computer Graphics* 14.2 (2008), 355–368 7.

[SBB*00] Sato, Mie, Bitter, Ingmar, Bender, Michael A, et al. "TEASAR: Tree-structure extraction algorithm for accurate and robust skeletons". *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*. IEEE. 2000, 281–449 5.

[SBTZ02] Siddiqi, Kaleem, Bouix, Sylvain, Tannenbaum, Allen, and Zucker, Steven W. "Hamilton-Jacobi Skeletons". *Int. J. Comput. Vision* 48.3 (July 2002), 215–231. doi: 10 . 1023/A:1016376116653 7.

[SCL*04] Sorkine, Olga, Cohen-Or, Daniel, Lipman, Yaron, et al. "Laplacian surface editing". *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 2004, 175–184 5.

[SJT14] Sobiecki, André, Jalba, Andrei, and Telea, Alexandru. "Comparison of curve and surface skeletonization methods for voxel shapes". *Pattern Recognition Letters* 47 (2014), 147–156 2.

[SLSK07] Sharf, Andrei, Lewiner, Thomas, Shamir, Ariel, and Kobbelt, Leif. "On-the-fly Curve-skeleton Computation for 3D Shapes". *Computer Graphics Forum*. Vol. 26. 3. Wiley Online Library. 2007, 323–328 2.

[SYJT13] Sobiecki, André, Yasan, Haluk C, Jalba, Andrei C, and Telea, Alexandru C. "Qualitative comparison of contraction-based curve skeletonization methods". *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer. 2013, 425–439 2.

[TDS*16] Tagliasacchi, Andrea, Delame, Thomas, Spagnuolo, Michela, et al. "3d skeletons: A state-of-the-art report". *Computer Graphics Forum*. Vol. 35. 2. Wiley Online Library. 2016, 573–597 2, 7.

[TZC09] Tagliasacchi, Andrea, Zhang, Hao, and Cohen-Or, Daniel. "Curve skeleton extraction from incomplete point cloud". *ACM SIGGRAPH 2009 papers*. 2009, 1–9 2, 4.

[VL98] Vasconcelos, Nuno and Lippman, Andrew. "Learning Mixture Hierarchies". *Proceedings of the 11th International Conference on Neural Information Processing Systems*. NIPS'98. Denver, CO: MIT Press, 1998, 606–612 3.

[Web09] Weber, Alfred. *Ueber den Standort der Industrien*. Mohr, 1909. isbn: 9785880850846 1.

DCE MRI Modality Investigation for Cancerous Prostate Region Detection: Case Analysis

Aleksas Vaitulevicius
Vilnius University
Institute of Data Science and Digital Technologies
Vilnius, Lithuania
aleksas.vaitulevicius@mif.stud.vu.lt

Povilas Treigys
Vilnius University
Institute of Data Science and Digital Technologies
Vilnius, Lithuania
povilas.treigys@mif.vu.lt

Jolita Bernataviciene
Vilnius University
Institute of Data Science and Digital Technologies
Vilnius, Lithuania
jolita.bernataviciene@mif.vu.lt

Roman Surkant
Vilnius University
Institute of Applied Mathematics
Vilnius, Lithuania
roman.surkant@gmail.com

Jurgita Markeviciute
Vilnius University
Institute of Applied Mathematics
Vilnius, Lithuania
jurgita.markeviciute@mif.vu.lt

Ieva Naruseviciute
National Cancer Institute
Vilnius, Lithuania
ieva.naruseviciute@nvi.lt

Mantas Trakymas
National Cancer Institute
Vilnius, Lithuania
mantas.trakymas@nvi.lt

ABSTRACT

Typically, prostate evaluation is done by using different imaging sequences of magnetic resonance imaging. Dynamic contrast enhancement, one of such scanning modalities, allow to spot higher vascular permeability and density caused by the malignant tissue. Authors of this paper investigate the ability to identify malignant prostate regions by the functional data analysis and standard machine learning techniques. The dynamic contrast enhanced images of the prostate are divided into the regions and based on those time-signal intensity curves are calculated. Two classification approaches: functional k-Nearest Neighbors and machine learning Support Vector Machine are used to model signal curve behavior on temporal variation matrix and timestamp based prostate region division of image data. Preliminary research shows that both functional data analysis and machine learning classification methods are able to identify highest saturation timestamp that gives best tissue classification results on timestamp based dynamic contrast enhanced region map obtained by Simple Linear Iterative Clustering algorithm. Cancer region classification results are better when the dynamic contrast enhanced images are subdivided into regions at each timestamp than when using a temporal variation matrix.

Keywords

prostate cancer, functional data analysis, machine learning, component, dynamic contrast-enhanced MRI

1 INTRODUCTION

Prostate cancer is one of the leading causes of cancer death worldwide. Among males, prostate cancer has second highest incidence rate after lung cancer according to the research given in paper [Bra20]. Although death rates have been decreasing in some countries, it remains a considerable disease affecting many patients. Due to nature of cancer, early diagnosis and treatment is critical. Preliminary identification of cancer involves Prostate-Specific Antigen (PSA) screening measuring concentration of a protein produced by the prostate, and the concentration is elevated inpatients with prostate cancer. Due to high level of false-negative and false-positive cases in PSA testing which lead to incorrect biopsies, a less invasive and more reliable procedure is needed. With the introduction of PI-RADS in the paper [Alq20], a structured reporting scheme for multi-parametric (mp) prostate Magnetic Resonance Imaging

https://www.doi.org/10.24132/CSRN.3201.7

(MRI) based on literature evidence given in the same paper and consensus expert opinion, the interpretation and performance of prostate cancer evaluation has considerably improved. Cancer evaluation is done by using different types of imaging (T2 Weighted images (T2W), Diffusion Weighted Images (DWI), Apparent Diffusion Coefficient map (ADCmap), Dynamic Contrast-Enhanced (DCE) images, etc.), each having own acquisition methods and purpose. Radiologists typically use at least several imaging sequences for more accurate diagnosis.

This research is focused solely on DCE sequence. Prostate DCE MRI data is gathered by capturing imaging sequences of the entire region of prostate during an intravenous injection of a contrast agent (typically gadolinium). Over a course of several minutes, a set of cross-sectional images is created at different time moments, usually every few seconds. The role of the contrast agent is to evaluate angiogenesis of tumor in DCE imaging. Since blood vessels are essential to cancer growth, tumors typically have higher vascular permeability and density which attracts higher amount of contrast medium as it is described in paper [Low11]. After acquiring such data, each cross-sectional image can be segmented into regions by using algorithm such as Simple Linear Iterative Clustering (SLIC) algorithm. Those regions can then be aggregated to single value by calculating mean, median or another metric of each region. Therefore, collected data can be fitted to functions $f_{xy} : T \to I$, where T is set of time points in which observations were made, I - set of aggregated intensity values and x, y are coordinates of the pixel. Functional Data Analysis (FDA) can be applied on these functions to detect, characterize, and monitor tumors together with machine learning methods.

A lot of work was already done in prostate cancer localization research by using T2W sequences for example in the paper [Juc16]. Moreover, DWI sequences were used to solve prostate cancer segmentation and severity evaluation problems. Examples of such papers are [Hot16], [WuC15] and [Bar15]. However, recent improvement in DCE MRI technology, described in paper [Cha18], create a motive to research DCE MRI sequences. The example of such paper is [Liu19]. However, in this paper only machine learning without FDA approach is tested. Therefore, the aim of this paper is to investigate a dynamic contrast imaging evaluation method for cancerous prostate zones localization in with focus to compare FDA and classical machine learning classification approaches while validating algorithms identifies cancerous zones with the ground truth samples obtained by histological tissue analysis after biopsy. The data for the investigation was provided by the Lithuanian National Cancer Institute (NVI) under the terms of bioethical agreement.

2 WORKFLOW

The structure of analysis workflow consists of data preparation, data preprocessing, segmentation, curve construction, data visualization, functional data analysis modeling and machine learning modeling steps, Fig. 1. The data used for investigation consists of four types: DCE MRI images (an example of single slice and 3 different timestamps is given in Fig. 2), prostate region masks (see Fig. 3), cancer region masks (see Fig. 4), and biopsy results.

The peculiarity of the MRI DCE image construction is that during the scan patient's prostate is scanned numerous times observing contrast agent saturation in tissue. Fast tissue region contrast saturation and fast agent washout may indicate malignant tissue region. As the maximum contrast saturation timestamp is not known prior, thus in this research experiments are conducted with two different data acquisition for magnetic resonance images approaches: segmenting temporal variance matrix calculated between all timestamps and segmenting each timestamp separately.

The block labeled as Data in Fig. 1 corresponds to data used in experiment. Examples of data types of data, used in experiment, are shown in Fig. 2, Fig. 3 and Fig. 4. In the Fig. 5 the overlay of these data types and biopsy masks is provided. More detailed characteristics of data types are explained in this section.

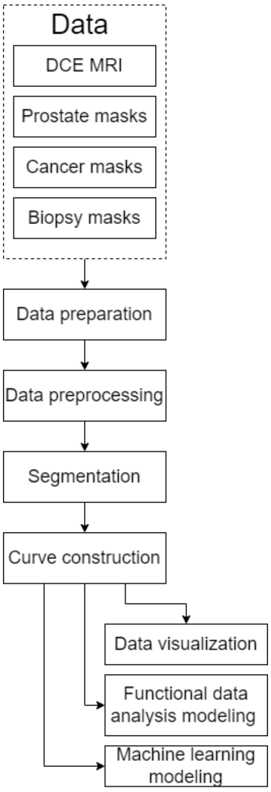


Figure 1: Analysis workflow structure.

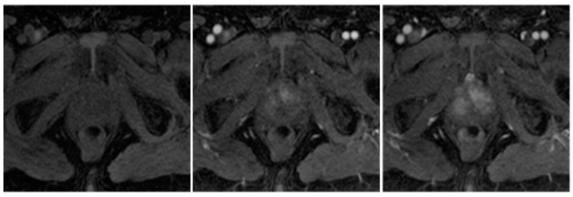


Figure 2: Example of DCE images of a single slice in different timestamps.

2.1 DCE MRI

DCE MRI data are grayscale images that are captured during an MRI procedure. Such images indicate the distribution of contrast agent in a particular cross-section of patient's pelvic region. The differentiating aspect of dynamic contrast imaging compared to other image sequence types is its bi-dimensional nature. DCE MRI sequence has a temporal dimension indicating the time moment when the DCE image was recorded ("timestamp"), and a spatial dimension linking the DCE image to a location of the cross-section in patient's body ("slice"). This allows to interpret the data in a 3D time series-based manner.

As stated earlier DCE MRI modality is the primary target of this research. This dataset contains 135660 anonymized images of 144 patients. Each patient having on average 41 slices (total range: 25 - 102) and 26 timestamps (total range: 5 - 55). For the case analysis and classification one patient data having explicit cancerous regions confirmed by biopsy were selected as not all of the patients have histological registration performed yet. This patient has 26 slices with prostate and 31 timestamps. Authors plan to generalize the investigation on all patients after finished histology registration.

2.2 Prostate region masks

Prostate masks are binary image-type data which indicates the region of prostate for each DCE MRI slice (all timestamps of the same slice have the same prostate region). Such masks were segmented manually by medical experts at NVI. Since MRI covers an area both above and below the prostate, some prostate masks are blank (i.e., are black).

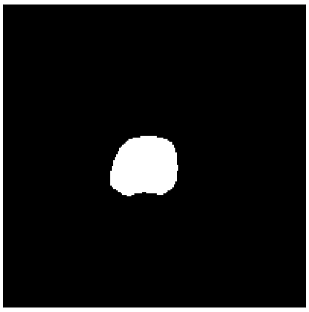


Figure 3: Prostate mask of a single slice.

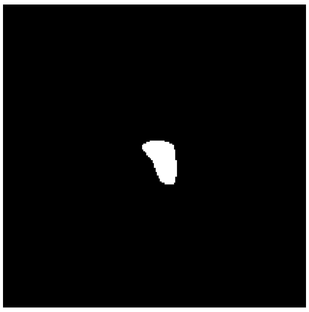


Figure 4: Cancerous region mask.

2.3 Cancer region masks

Similarly, to prostate region masks, cancer region masks are binary images showing the location of cancer tumor which are manually segmented by experts and may be blank. Tumors can be of three types: malignant, clinically insignificant, and benign. Cancer masks indicate only the suspected region of cancer according to medical experts. Factual tumor type is diagnosed through a biopsy.

2.4 Biopsy results

Each patient underwent on average 15 biopsies (total range: 3 - 25), each biopsy showing tumor type and tumor severity based on a Gleason score [Alq20]. The biopsy outcome data is split into two datasets of different format:

- Tabular: contains numerical identifier of patient, slice, biopsy, three Gleason scores (first, second, and combined).
- Mask: multi-label mask showing the location of biopsies; each biopsy has a numerical identifier linking to the tabular dataset.

While biopsies provide ground truth label for tumors, there are two flaws:

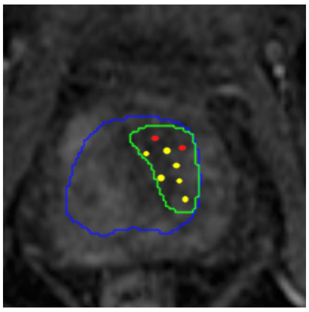


Figure 5: Overlay image combines middle image of Fig. 2, Fig. 3, Fig. 4 and biopsy mask images: prostate region (blue), cancer region (green), malignant biopsy (red), clinically insignificant or benign biopsy (yellow).

- Biopsies provide information only in a point-wise manner. The gaps between biopsies are essentially missing data where ground truth information is unavailable.

- To perform a biopsy, a medical expert inserts a needle into prostate and extracts a tissue sample for examination. The needle punctures several slices without knowing which slice exactly the tissue was taken from. Such process introduces a systematic uncertainty, for instance, a slice having no tumors may show positive diagnosis if cancerous tissue was extracted from a different slice above or below.

2.5 Data preparation

Before workflow data preparation is performed. The first step of data preparation is extracting metadata from each image object (MRI, cancer masks, biopsy masks). Extracted metadata parameters are patient ID, timestamp ID, slice ID, resolution, maximum pixel intensity, minimum pixel intensity, mean pixel intensity. This information allows to detect incompatible data (e.g. differing image resolution) or missing data (e.g. MRI images containing only pixels of identical value). The second step is biopsy aggregation which is the collection and aggregation of biopsy information from multiple CSV-type files into single source for more convenient processing.

2.6 Data preprocessing

The next step of workflow is data preprocessing. Data preprocessing consist of 2 steps. The first step is mask rescaling which is transformation of prostate and cancer mask files into binary format compatible with used algorithms and the other step is image rescaling which is transformation of MRI images into appropriate format with 8-bit pixel precision (256 grayscale values).

2.7 Segmentation

As mentioned before experiments in this research are done with two different approaches. The first approach is using Temporal variation matrix (TVM) calculated between all timestamps. The first step is calculating TVM. This step is a construction of a matrix with identical resolution as source MRI image which represents statistical variance of signal value of each pixel between all timestamps of a selected slice. TVM shows regions with high (bright) and low (dark) change in signal intensity over time which is used for region of interest segmentation. Example of a TVM is shown in Fig. 6 which is calculated from slice whose timestamp examples are displayed in the Fig. 2.

The next step is TVM segmentation by applying SLIC algorithm, introduced in paper [Ach12], to separate the prostate region into segments. Selected SLIC algorithm parameters are 50 and 7 for segment number and compactness respectively. Those parameters were selected using expert judgement based on the following criteria:

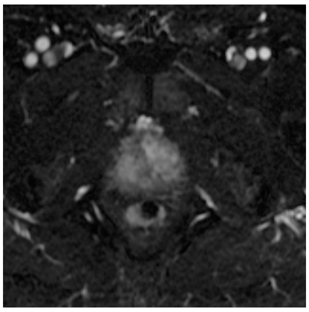


Figure 6: Temporal variation matrix calculated from all timestamps of slice whose timestamp examples are given in Fig. 2.

- Number of segments parameter should be high enough to have separation between cancerous and healthy tissue, but low enough to remain computationally relevant and keep segments visually distinguishable.

- Compactness parameter should capture similar intensities, but still prioritize color proximity in favor of maintaining circular shape because cancerous growth typically does not have irregular formation.

Acquired segment locations are projected back into MRI images that were used for TVM construction (example of TVM segments in Fig. 7). Lastly, segment-level aggregation is performed by calculating the means of pixel intensities inside each segment.

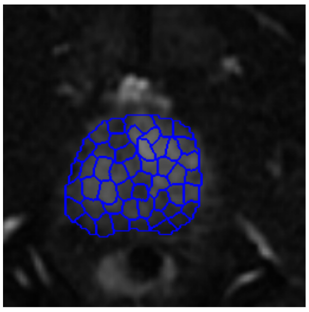


Figure 7: Segmented and zoomed TVM displayed in Fig. 6.

The second approach is segmenting each MRI image separately. The first step is segmentation of each MRI image by applying SLIC algorithm to separate the prostate region into segments in each timestamp separately. Resulting in the number of segmentations which is equal to number of timestamps. The parameters chosen for SLIC algorithm are the same as for the first approach. Then each segmentation is projected to all timestamps e.g.: patient, for which MRI scan was applied 30 times, will have 30 timestamps. Each timestamp is segmented resulting in 30 segmentations. Each segmentation is projected to all 30 timestamps. Further workflow is applied for each segmentation separately. Lastly, segment-level aggregation is performed by calculating the means of pixel intensities inside each segment.

2.8 Curve construction

The resulting time series of each segment contains information of mean intensity of intensities within each segment at the given timestamp. The greater intensity indicates higher concentration of contrast agent. As explained in the introduction, the malignant tissue accumulates contrast agent faster than a healthy one. Therefore, these time series are used to construct functional data which later on is transformed to represent mean intensity growth speed within each region.

Each segment's aggregated intensity values are used to construct a single time series curve. Prior to this step, each segment is labeled into binary classes. Positive class 1 is assigned to segments which has an intersection with cancerous region mask $\geq 50\%$, overlaps with at least one malignant biopsy and does not overlap with any other biopsy. Negative class 0 is assigned to segments which does not overlap with malignant or clinically insignificant biopsies and does not have intersection with cancerous region mask. Meanwhile, the remaining segments are not used in the training or validation. These segments are either:

- have clinically insignificant or malignant biopsy while no overlap with cancerous region mask.
- have $< 50\%$ overlap with cancerous region mask.
- have $\geq 50\%$ overlap with cancerous region mask but no malignant biopsy.

The example of labeled segments is displayed in the Fig. 8. Segments marked with green contour have negative class, while red contour - positive class. Segments with orange contour have overlap $\geq 50\%$ with cancerous region mask and no overlap with malignant biopsy. Furthermore, segments with white contour either have intersection $< 50\%$ with cancerous region mask or no overlap with cancerous region mask but have other than benign biopsy. Segments with green and red contour are used in modeling while the rest are not. The curves of discrete points of this example are displayed in Fig. 9.

Afterwards, the time axis of those curves of discrete points are normalized to interval [0, 1]. The resulting curves of discrete points are then smoothed by using B-spline basis function whose calculation is presented in the paper [DeB72]. The parameters of this calculation are:

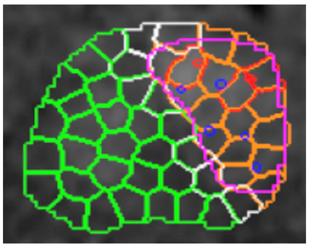


Figure 8: Example of segment classification where segments marked with red contour have class 1, segments with green contour - class 0 and segments with white or orange contours are not used in modeling.

- Order - the order of polynomial function, called B-splines. The used value for this parameter is 4.
- basis function number - number of basis functions to use for calculation. The used value for this parameter is 18.

These parameters were obtained by calculating the best Generalized Cross Validation (GCV) score on a single patient by using grid search-type algorithm. Tested combination of parameters are 2, 3, 4, 5 for order and 5, 6, ..., 29, 30, 35, 40 for basis function number. Functional data, related to Fig. 9, are displayed in Fig. 10.

Furthermore, in curve construction step, derivatives of first degree are calculated from functional data. These calculations are described in the paper [But76] and these derivatives are interpreted as velocity of intensity change over time. The first degree derivatives of functional data, related to Fig. 10, are displayed in Fig. 11.

Lastly, these 1st degree derivatives of functional data are registered by using landmark registration The chosen landmarks are the points in t axis in which functional data has a maximum value. The registrations are performed for each patient and segmentation separately. Registered functional data related to Fig. 11 are displayed in Fig. 12.

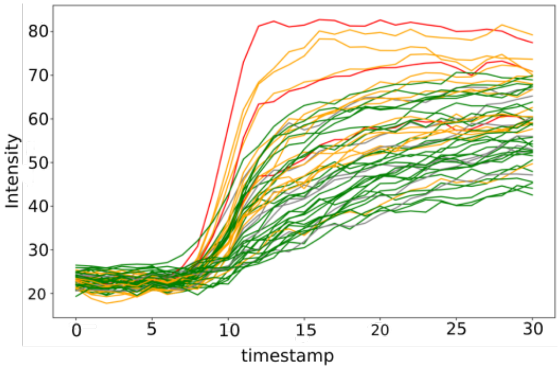


Figure 9: Example of curve created from discrete points where red curves are curves of class 1, green curves - of class 0 while orange is not used in modeling.

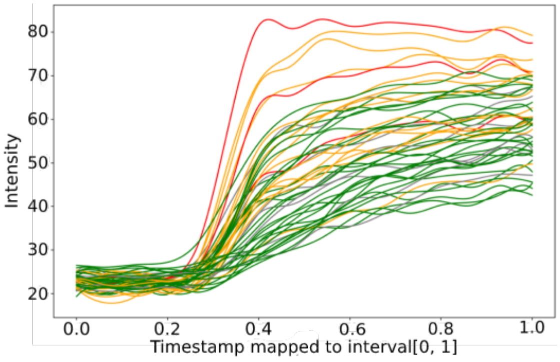


Figure 10: Functional data displayed in Fig. 9 and created by using B-spline.

2.9 Functional data analysis modeling

In this workflow, 2 different approaches of SLIC region classification are tested. The first is Functional Data Analysis approach. In this approach model is created by using k-Nearest Neighbors (kNN) algorithm and using functional data as training set. The neighboring parameter was obtained by the grid search. Seven neighbors were depicted as the optimal choice.

2.10 Machine learning modeling

The second approach of this workflow to classify SLIC regions is modeling by using Machine learning method - Support Vector Machine (SVM) algorithm. Training set used for training is extracted features from functional data derivatives: integrated depth, modified band depth, maximum intensity, time of maximum intensity and 10 uniformly spaced intensities from discretized curve derivative in interval [0.05, 0.95].

Due to the unbalanced class problem SVM model is validated by using cross validation method - stratified 5 k-folds.

3 RESULTS

In the Table 1, metrics of results of SLIC zone classification with different approaches are shown. For precision, recall and F1 metrics the positive class is cancerous region and referred as 1. Classes are highly unbalanced, the number of data points with class 1 for TVM approach is 63 while for single timestamp segmentation - 64. Meanwhile, the number of data points with class 0 for TVM approach is 1184 while for single timestamp segmentation - 1185.

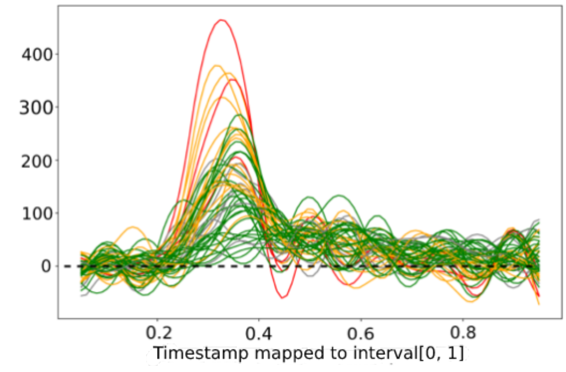


Figure 11: 1st degree derivatives of curves displayed in 10.

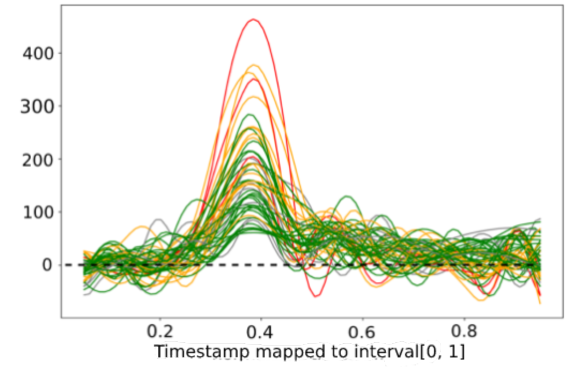


Figure 12: Registered functional data displayed in Fig. 11.

For segmenting in each image separately approach multiple different are obtained as number of different ways to segment the prostate to regions is equals to number of timestamps. In the Table I., the chosen timestamps for each FDA and modeling approach are the one which achieve the best results. For FDA modeling it was 10th timestamp and for ML modeling - 8th.

Number of SVM models are equal to number of folds used in stratified K-folds method which is 5. The metrics of these models are aggregated by calculating mean and standard deviation STD. The best results is produced by the kNN functional data classifier.

4 CONCLUSION

Preliminary research investigating MRI-DCE modality scans by applying functional data analysis and machine learning methods was presented in the paper. The results obtained by the comparison of the machine learning and FDA methods allows authors to conclude:

- Both FDA and ML based classification approaches gives best results at almost the same tissue saturation timestamp while using non-TVM intensity map.
- Grid search of neighboring parameter indicated that the seven neighbors is an optimal choice giving best classification results.
- Timestamp base SLIC application outperforms TVM intensity mapping giving FDA kNN classification precision of 1, recall 0.65, and F1 score - 0.71.

https://www.doi.org/10.24132/CSRN.3201.7

Metric	**FDA modeling**		**ML modeling**							
	TVM	***Best timestamp SLIC segmentation***	***TVM***				***Best timestamp SLIC segmentation***			
Set	-	-	***Training***		***Testing***		***Training***		***Testing***	
	-	-	***Mean***	***STD***	***Mean***	***STD***	***Mean***	***STD***	***Mean***	***STD***
Precision	0.929	1	0.969	0.016	0.8	0.267	0.960	0.08	0.7	0.4
Recall	0.413	0.625	0.504	0.036	0.303	0.125	0.719	0.109	0.5	0.316
F1	0.571	0.714	0.663	0.034	0.428	0.153	0.817	0.085	0.567	0.327
Balanced accuracy	0.706	0.778	0.752	0.018	0.649	0.062	0.859	0.055	0.75	0.158
Specificity	0.998	1	0.999	0.0004	0.996	0.005	0.9998	0.0004	0.999	0.002

Table 1: Modelling results

- SLIC algorithm applied to TVM intensity mapped images in ML modeling produces more stable testing results in terms of standard deviation than those obtained by SLIC applied best timestamp.

Obtained results shows potential points of further action for the modeling results improvement:

- Class discrimination can be improved by incorporating data from the functional boxplot analysis.
- More extensive search of new features, feature combinations, feature transformations may improve model learning.
- Extensive search of more suitable model parameters may improve model learning.
- Proposed workflow needs to be applied on all 144 patient data as currently not all of them has histological tissue analysis performed. Moreover, a lot of patients DCE image data results in unusual functional data. This may indicate faults in data or limitations of the proposed workflow. Therefore, those cases have to be examined.
- underrepresented classes strongly affect model's ability to segregate different classes. Thus resampling techniques such as over-sampling as well as configuring balancing class weights during model training may improve class discrimination and shall be used in further experiments with greater number of patients.
- Further experiments have to be conducted with greater number of patients to determine the timestamp for segmentation which achieves the most accurate results.

5 ACKNOWLEDGEMENT

The authors are thankful for the high performance computing resources provided by the Information Technology Research Center of Vilnius University.

6 REFERENCES

[Ach12] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence 34, no. 11 (2012): 2274-2282.

[Alq20] Alqahtani, S., Wei, C., Zhang, Y., Szewczyk-Bieda, M., Wilson, J., Huang, Z., and Nabi, G. Prediction of prostate cancer Gleason score upgrading from biopsy to radical prostatectomy using pre-biopsy multiparametric MRI PIRADS scoring system. Scientific reports 10.1 (2020): 1-9.

[Bar15] Barrett, T., Priest, A.N., Lawrence, E.M., Goldman, D.A., Warren, A.Y., Gnanapragasam, V.J., Sala, E., and Gallagher, F.A. Ratio of Tumor to Normal Prostate Tissue Apparent Diffusion Coefficient as a Method for Quantifying DWI of the Prostate. American Journal of Roentgenology vol 205, (2015): 585-593. Doi: 10.2214/AJR.15.14338.

[Bra20] Bray, Ferlay, J., Soerjomataram, I., Siegel, R.L., Torre, and L.A. Jemal, A. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: a cancer journal for clinicians 68.6 (2018): 394-424.

[But76] Butterfield, K.R. The computation of all the derivatives of a B-spline basis. IMA Journal of Applied Mathematics 17, no. 1 (1976): 15-2

[Cha18] Chatterjee, A., He, D., Fan, X., Wang, S., Szasz, T., Yousuf, A., Pineda, F., Antic, T., Mathew, M., Karczmar, G.S., and Oto, A. Performance of ultrafast DCE-MRI for diagnosis of prostate cancer. Academic radiology, 25(3) (2018): 349-358.

[DeB72] De Boor, C. On calculating with B-splines. Journal of Approximation theory 6, no. 1 (1972): 50-62.

[Hot16] Hotker, A.M., Mazaheri, Y., Aras, O., Zheng, J., Moskowitz, C.S., Gondo, T., Matsumoto, K., Hricak, H., and Akin, O. Assessment of Prostate Cancer Aggressiveness by Use of the Combination of Quantitative DWI and Dynamic Contrast-Enhanced MRI. AJR. American journal of roentgenology vol. 206,4 (2016): 756-63. Doi:10.2214/AJR.15.14912.

[Juc16] Jucevicius, J., Treigys, P., Bernataviciene, J., Briediene, R., Naruseviciute, I., Dzemyda, G., and Medvedev, V. Automated 2D Segmentation of Prostate in T2-weighted MRI Scans. International journal of computers communication & control, [S.l.], v. 12, n. 1, p. 53-60, dec. 2016. ISSN 1841-9844.

[Liu19] Liu, B., Cheng, J., Guo, D.J., He, X.J., Luo, Y.D., Zeng, Y., and Li, C.M.. Prediction of prostate cancer aggressiveness with a combination of radiomics and machine learning-based analysis of dynamic contrast-enhanced MRI. Clinical radiology, 74(11) (2019): 896-e1.

[Low11] Low, R. N., Fuller, D. B., and Muradyan, N. Dynamic gadolinium-enhanced perfusion MRI of prostate cancer: assessment of response to hypofractionated robotic stereotactic body radiation therapy. American Journal of Roentgenology 197.4 (2011): 907-915.

[WuC15] Wu, C.J., Wang, Q., Li, H., Wang, X.N., Liu, X.S., Shi, H.B., and Zhang, Y.D. DWI-associated entire-tumor histogram analysis for the differentiation of low-grade prostate cancer from intermediate-high-grade prostate cancer. Abdom Imaging 40, 3214-3221 (2015).

The Iterative Development of an Online Multiplayer Escape Room Game for Improving Social Interaction through Edutainment

Ali Najm[1,2]
ali.najm@cut.ac.cy

Maria Christofi[1,2]
mu.christofi@edu.cut.ac.cy

Christos Hadjipanayi[1]
cp.hadjipanayi@edu.cut.ac.cy

Christos Kyrlitsias[1,2]
cm.kyrlitsias@edu.cut.ac.cy

Despina Michael-Grigoriou*[1,2]
despina.grigoriou@cut.ac.cy
(corresponding author)

[1] GET Lab,Department of Multimedia and Graphic Arts,Cyprus University of Technology, Archiepiscopou Kyprianou 30,3036, Limassol, Cyprus

[2] CYENS Centre of Excellence, 23 Dimarchias Square, 1016, Nicosia, Cyprus

ABSTRACT

Digital Games are nowadays used for several purposes beyond entertainment. Such purposes include but are not limited to education, promoting cultural heritage, and improving well-being aspects. A rich body of literature presents experimental studies, investigating whether a serious game achieves its aim. However, most of such papers often omit to provide adequate information on the development process followed, game mechanisms and techniques used, making the reproducibility of the game as such, by other researchers, difficult. This results in a lack of knowledge transfer between researchers, who usually must develop applications under investigation by themselves when at the same time industrial gaming companies rarely publicize the technical insights of their work. This paper aims to contribute towards filling this knowledge gap within the scientific community, using as a case study an online, multiplayer, escape room game, which aims to improve social interaction through edutainment. The full process of its development with details for the various components that the game comprises are presented. We are expanding on the functionality of the game and the optimization of the 3D environment and the assets, among other aspects. Results of white and black-box testings taking place at the end of each development cycle showed that the integration of the various components described within the paper led to a robust game.

Keywords

Serious Games, Multiplayer, Online, Iterative Development, Social Interaction, Edutainment.

1 INTRODUCTION

Digital games are usually used for entertainment and have the ability to engage players, thanks to the games' interactive nature [Bal07a]. Games that do not have entertainment as their primary purpose are called serious games, which, among other purposes, are used for education, the promotion of cultural heritage [Tsi19a], the improvement of well-being aspects [Why15a], and even cognitive training [Kat19a]. Serious games could constitute boundary objects [Sta89a], in the sense that serious games can facilitate cooperation between multiple social systems while maintaining a different identity in each one [Ter21a]. This means that a serious game could be used to elicit various outcomes in different cohorts while providing scientific knowledge on intersecting areas of research. The process of developing a robust serious game can be a complex process, as more aspects need to be taken into account [Bra16a]. Especially for mobile devices, physical and technical characteristics need to be considered throughout the whole development process, from the initial planning to the development of the game itself [Bal15a]. This is because mobile devices, compared to desktop ones, bear important physical and technical differences in their display size, processing power, data input methods, and memory space. These considerations and their practical

https://www.doi.org/10.24132/CSRN.3201.8

solutions and implementations in the development process are often overlooked in the scientific publications, even though they could be useful for reproducibility purposes or for new studies by other researchers.

A great amount of the literature on serious games relies on only a brief description of the applications developed and is usually focusing on the experimental part of the study. Technical details about the development that would benefit future researchers are often missing from research papers. The present paper is in line with the work of Symborski, et al. [Sym17a], which describes two serious games that teach the mitigation of cognitive biases and the experiment cycles and playtesting that they conducted. The research elaborates on various design approaches and provides outcomes from the playtesting phases. Likewise, Zilak, Car and Jezic [Zil18a] describe the development of an elementary mathematical virtual classroom prototype based on Oculus Rift and Leap Motion devices and the user evaluations they conducted. A few examples of studies that attempt to offer more technical details [Del21a, Ari14a] elaborate on the rendering methods or the computation algorithms they used, which avail the reproducibility of similar applications.

The aim of this paper is to present the iterative development method that was used for our multiplayer escape room mobile game to facilitate knowledge transfer between researchers, who usually must develop applications under investigation by themselves when at the same time industrial gaming companies rarely publicize the technical insights of their work. Various technical details and specifications for the components that built up the game, including the game mechanics, functionality, and optimizations made are also provided. The purpose of the developed game itself is to improve social interaction through edutainment and to contribute towards filling knowledge gaps regarding game development for social interaction.

2 MULTIPLAYER ESCAPE ROOM GAME

2.1 Development Method

Iterative development, which breaks the process of developing a software into iterations that contain the whole process (planning, design, development, and testing steps), was chosen as our approach to the development of the game. Specifically, the agile method was mainly followed as it focuses on iterative refinements and incremental improvements to working software, and in our case, a mobile game [Asu11a]. This method was chosen because it provides direct feedback about the improvement of usability and functionality of the game. This development approach is widely used in studies where game tester's feedback is necessary [Ale16a, Bal07a, Ter21a]. The subsection below provides a brief overview of the process that was followed.

2.1.1 Development Method Overview

The first step of the development process was a planning phase, in which the game specifications were decided. Once the type of the game, its platform, target group, aim, setting, and key features were decided, the pre-production phase followed, in which the game scenario and storyline were decided and storyboards were created [Ale16a]. The first prototype of the game was developed and tested, which included the core mechanics of the game (navigation system, interaction with objects, puzzle mechanics, countdown system), with no finished 3D models or textures. When all the core mechanics were internally tested and confirmed to be working according to the required specifications, then the production phase started, where all the assets of the game (3D models, textures, UI elements, sounds, programming scripts) were gathered. The game was developed using the Unity game engine. Then, at the testing phase, a version of the game was tested with either black or white box testing. The testings that were conducted are described in Section 3. After each testing, another iteration of the development process was taking place, for improving and correcting any shortcomings revealed from the testing.

2.2 Game Overview and Design

The genre of the game is a 3D multiplayer escape room mobile game. In escape rooms, players are locked in a room, and by exploring the room, finding clues, and solving puzzles, they can find the way to unlock the room and exit. They usually have time limitations and revolve around various themes and narratives [Bak19a]. Escape room applications are common educational tools for heritage sites. For example, the escape room application for the industrial heritage site of Zissimatos textile factory is designed to educate visitors on the production flow of the factory by gathering objects and solving riddles [Gai18a]. Other similar applications, such as "MillSecret" and "Salamis", make use of the escape room gamification approach and AR technology to enhance the in-situ experience of cultural heritage sites by integrating dispersed digital elements throughout the sites [Tzi20a, Kou18a]. Contrary to the AR related applications mentioned, our game takes a Window on World (WoW) Virtual Reality approach, in order to tackle the limitation of having to be at a specific physical location for the experience to play out. Besides, the functionality of escape room games that are founded on AR technology and physical locations is sensitive to alterations of the games' physical components, which are often prone to change because of temporality [Vas19a]. In our game, players are locked in a series of rooms within the virtual representation of a real castle and are encouraged to communicate and cooperate in order to solve the puzzles and obtain the exit-pass of the castle chambers. Interestingly enough, being virtually transported inside a 3D representation of the castle enables players to safely visit all site locations, even including those restricted in real life for safety reasons.

Another difference with similar escape room games mentioned is that our game was designed to target educational but also social well-being aspects. For the educational aspect, players are able to learn about the architecture of the castle, its significance, and its history through different time periods, among other information. The whole castle and its chambers are used as the playable environment of the game, in order for the players to visit all areas of the castle during gameplay and gain knowledge for all of its parts. For the social aspect, the game has been designed as an online multiplayer game, where the players should communicate through their smartphone's microphone through voice and interact with each other. Some mechanics of the game disallow any team player from taking the lead and solving all the puzzles without help throughout the entire game session. This is a rarely used experience design that makes it necessary for all team players to speak to each other, ask questions and seek clear answers from their teammates, so all of them contribute in solving the puzzles, winning the game, and participating in the learning process. Of course, such an experience design entails that the chances of winning are much higher when the teammates are willing to be respectful and helpful to each other, which could be an additional challenge to the gameplay. Even so, escape room games are found to cultivate a sense of trust and community among strangers while bringing them together through mutual effort [Gai18a], which arguably is one of their most underutilized properties. Therefore, the educational significance of this game is both cultural and prosocial. This combination of multiplayer smartphone system architecture, escape room setting, and digital storytelling is an innovative gamification approach, which, to our knowledge, has yet to be integrated within the context of cultural heritage education. A video demo of the game can be found as supplementary material.

2.3 Storyline

The narrative of the game is inspired by the tragedy play named "The Tragedy of Othello, the Moor of Venice" (in short "Othello"), written by William Shakespeare [Sha93a]. The castle of Famagusta is allegedly associated with the famous legend of Othello, as presented in Shakespeare's play, during the British rule in Cyprus. In the game, the players assume the role of detectives, who seek to investigate and solve a mystery that is taking place in the castle. Once the players solve an initial puzzle inside the castle, players are transferred into another dimension and the castle's main entrance gets locked. Without any obvious escape route left, the players are instructed by Othello's ghost to visit all the rooms of the castle and solve puzzles that will help them discover the means to escape.

2.4 Environment, 3D Assets and Avatars

2.4.1 Environment, Weather Simulation and Lighting

Environment

The environment of the game has been designed with historical and architectural accuracy according to the current state of Othello's Castle which is located in Famagusta in Cyprus. The castle has been 3D modeled based on actual castle plans and reference photos (see Fig.1).

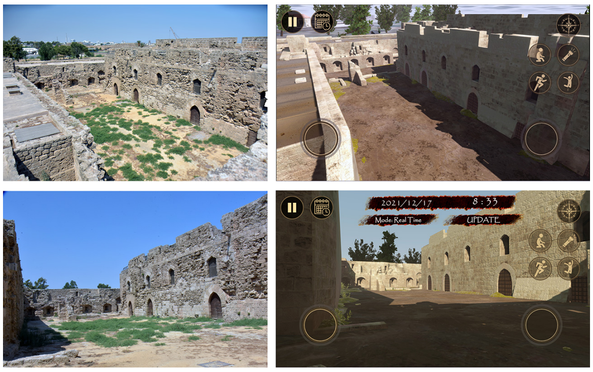


Figure 1: Real photos from the castle (left) and the 3D model created (right).

The model was compared with a 3D laser scan of the castle[1]. 3D scans are not usually used for interactive applications such as games, as they have a massive number of point clouds and segments and they are unnecessarily large and impractical especially for smartphone devices. Even the optimized version of the 3D scan of the castle's structure, includes 3,100,000 polygons and has a size of 346 MB. Instead, the corresponding 3D model, created manually for this game, has 10,000 polygons and a total size of 3.8 MB with more structural details (see Fig.2).

Weather Simulation

In addition, weather effects for simulating natural light and weather conditions were added. For this purpose, the "Enviro Lite - Sky and Weather" (version 2.3) plugin was used from the Unity Asset Store. This plugin provides a real-time weather simulation by simulating the sun and moon through direct light. The lights'

[1] http://ephemera.cyi.ac.cy/?q=OthelloTower

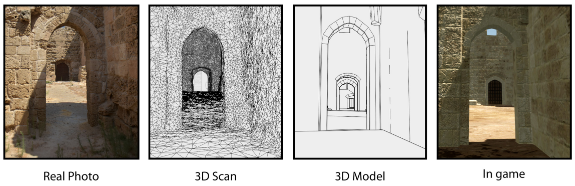


Figure 2: Comparison between the real photo (left), 3D Scan (middle-left), 3D model (middle-right) and in-game result (right).

position, intensity, and color are updated continuously according to the real weather conditions of a specific time and date of the year. Moreover, environmental factors such as fog, dynamic clouds, and stars were added through this plugin.

In the starting area of the game, outside the virtual castle, where players connect, gather, and get familiar with the controls, the weather simulation is happening in real-time. For example, if they start the game on 25 July at 13:00, the sun location and shadows will be according to the real weather conditions on that specific day and time. However, when the players solve the initial introductory puzzle, the game shifts to nighttime for all players, to accommodate the storyline. At the last stage of the game when the players solve the second to last puzzle, a trigger activates rain and lightning effects, which the player can see and listen to through respective sound effects that were added through the same plugin. When the final puzzle is solved, the environment changes to sunrise and players can see more natural light in the environment.

Lighting

Lighting is another aspect of video games that usually requires a lot of real-time processing power and an aspect that had to be taken into consideration. In our mobile game, one real-time directional light was added, with shadows for the sun/moon movement and one baked directional light without shadows for inside and outside lighting. The torch held by the players, an essential asset that helped resolve the visibility issues of players, was created as a real-time light source and the rest of the lighting was baked in the game. Baking is a pre-calculation of highlights and shadows for a (static) scene, the information of which is then stored in a lightmap. The baked lightmap has been rendered on a second UV map of each 3D model generated by Unity. Baking images were selected to be 2048X2048 pixels in ETC_RGB4 (4 bits/pixel compressed RGB) format for an optimized result in a matter of size and quality.

2.4.2 3D Assets and Textures

3D Assets

All the 3D objects inside the castle (e.g. clue items, decorative items, boxes) were created with a low polygon count in order to reduce the rendering process per frame. Additionally, the surrounding environment of the castle consists of low poly models of monuments and greenery for achieving a more realistic experience for the players. One of the main challenges was that it was preferred for the whole game to take place in one scene, which is arguably a non-conventional approach by game development standards. The reasons this approach was chosen, instead of separating each room into a separate scene were: (i) to have one integrated environment where all players can move freely and explore the whole castle, (ii) not loading scenes during run time to avoid delays when moving between rooms and (iii) to preserve the continuity of communications and gameplay.

Textures

For optimizing the performance, which was a major challenge, only a single texture has been used and applied to the whole castle. The single texture used for the castle is designed to cover the diversity of its erosion surfaces, as illustrated in Figure 3 (left). All the blocks (bricks) have been placed on UV maps to resemble the appearance of the real structure. A normal (Figure 3, middle) and a height map (see Fig.3) were also created, to give the illusion of bumps on the wall's surface.

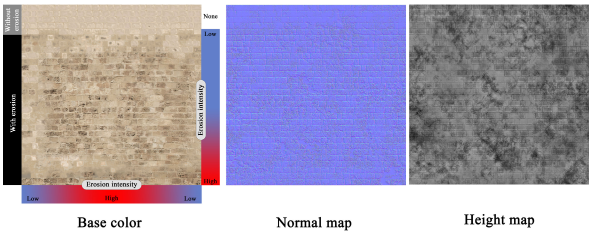


Figure 3: Castle model texture design.

Avatars

An avatar is the digital representation of a player in games [Now18a]. It allows the players to experience and interact with game objects and other users. In our game, players can choose their own avatar from a list of avatars in the main menu of the game and can see each other's avatars during the game (see Fig.4). According to Trepte and Reinecke [Tre10a] identifying with an avatar can increase media enjoyment and has positive outcomes for the play experience. Additionally, Van Ryn, Apperley, and Clemens [Van18a] mentioned that in multiplayer games, avatars are important for visually communicating with the other players. A different color and a circular halo below the avatar are assigned to each player, so the avatars become more distinguishable among teammates.

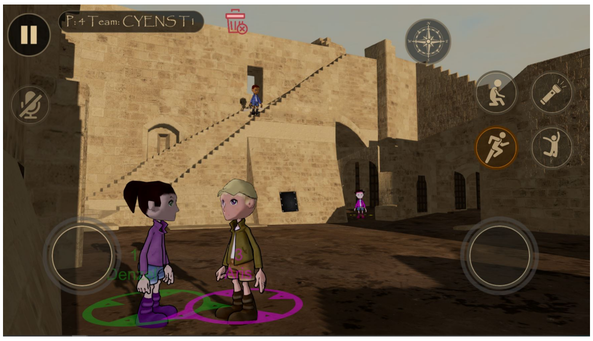


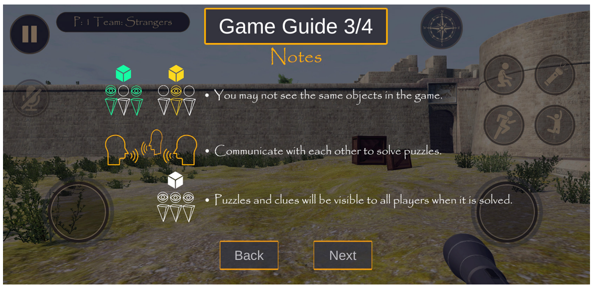
Figure 4: Players can see each other's avatars, visually enhancing communication with other players.

2.5 Functionality

2.5.1 *Navigation System*

Players can control their avatars through virtual joysticks on the screen. Virtual joysticks are inspired by their analog counterparts [Kim15a] and are used widely in commercial 3D mobile games like Terraria[2] and Call of Duty Mobile[3]. Two semi-transparent on-screen joysticks with circular knobs, on the left and right side of the screen were added. The left joystick is used for navigation (avatar movement) and the right one for head and body rotation. The player can drag the knobs relatively to the joystick's center to send directional commands.

Moreover, on the right side of the screen, four small circular buttons represent the players' ability to crouch, turn on/off a torch to light up dark spaces, run and jump (see Fig.4). Players can interact with clues and puzzles in the rooms and move boxes around and jump onto them, to reach higher places. During the game, the ghost of Othello appears as a UI image to guide the players. The players can listen to him and read the text in the text box. Also, at the beginning of the game, there are four UI pages of instructions for new users to explain the controls, mechanics, and features of the game (see Fig.5).



Game Guide 3/4

Notes

- You may not see the same objects in the game.
- Communicate with each other to solve puzzles.
- Puzzles and clues will be visible to all players when it is solved.

Figure 5: One of the instructions pages.

2.5.2 *Map*

On the top right side of the screen, there is a small circular button with a compass icon, which represents the map. Players can tap on that button to show/hide the map. The map is designed to be world-oriented.These kinds of maps show the entire game world with the north direction at the top, regardless of the orientation of the avatar and the player's perspective [Ada14a]. They are usually hidden from the screen, and players must press a specific button to show/hide it and are usually static.

To implement the map, the exact plan that accurately matches the top view of the 3D model of the castle was used to design a 2D map. The designed map was placed as an image under the 3D model of the castle (in the z-axis). An orthographic camera was placed facing on the image and above the 3D model of the castle. This type of camera was used because it is useful for rendering 2D scenes and UI elements and uses an orthographic projection, in which the object's size in the rendered image stays constant, regardless of its distance from the camera. Moreover, between the 3D model of the castle and the 2D designed image, there are small dots connected to the players' location and colors representing each player's live spatial location in the castle. The camera renders only the map-related layers and no other 3D objects in the scene through the Culling Mask feature of the Camera in Unity. Finally, the rendered image through this camera is projected on a target texture where the players can monitor their and other players' location on the map. For navigation aid purposes, an arrow on the map shows players where their next destination is.

2.5.3 *Communication and Collaboration*

It is important to note that there is no competition between the players. Striving toward a common goal could reduce subgroup categorization and transform the "us" versus "them" perception into a more inclusive "we" [Gae00a, Has13a]. One of the main challenges was to find a proper way to maximize and encourage cooperation, interaction, and communication between the players. The challenge was addressed by enabling or disabling the visibility of different key objects to the players, depending on their current role. In each room, one player is randomly chosen as a "solver" who can only view and solve the puzzle but not the related clues, whilst the rest of the players (named as "helpers") can view and have access only to the clues but not to the puzzle.

To ensure that all the players will be selected as a "solver" during the game, the following algorithm was used. Two lists of players are created at the beginning of the game. List A which is the list of all the connected online players at the moment and List B is the list of all the players who didn't play as a "solver" yet.

[2] https://terraria.org/
[3] https://www.callofduty.com/

List B is empty at the beginning. The players in the game, before reaching each room, collide with a trigger for the selection process. As soon as one player activates the trigger, all the players will update List A and List B but only the master player (who hosts the game and is selected automatically through the "Photon Unity Networking (PUN 2)" plugin) will select a random number to indicate the "solver" ID. The result will be shared with all the players through the network. The selected player will be removed from List B. Then, each device will check if the current player ID is equal to the "solver" ID and will take action accordingly. The clues will be deactivated and the puzzle of this specific room will be activated for the "solver" and the opposite for the "helpers". When the puzzle is solved, the clues and puzzle(s) of that room become visible to all players and available for further interaction. Also, if any player leaves or disconnects from the game, the selection process is repeated to avoid losing the "solver" so that the remaining players do not get "stuck" in the game. The flowchart of the "solver selection" algorithm is provided in the supplementary materials.
By implementing this mechanism in the game, cooperation and direct communication between all players become necessary and thus the voice chat feature was added as a convenient communication method. The voice was integrated into the game through the "Photon Voice 2" plugin.

2.6 Rooms, Puzzles, and Clues

2.6.1 Rooms

The castle consists of 6 main rooms which are originally locked. Each room is dedicated to a specific topic relating to the castle. The rooms and their respective topics pertain to Historical periods, Castle visual information, Construction characteristics, Castle's architecture changes, Music knowledge, and the Shakespearean tragedy of Othello. As the game's storyline follows a linear path, the players enter the rooms in a predefined order. For the players to move to the next location, they are encouraged to communicate and collaborate in order to find the clues and solve the puzzle(s). Players have 90 minutes in total to finish the game, with a chance of gaining extra time by collecting time bonus figurines hidden in areas inside the castle.

2.6.2 Puzzles and Clues

For the puzzle mechanics, the object interaction, the narration trigger, and the main UI menu, the "First person narrative adventures + complete puzzle engine" plugin (version 1.1.2) was used from the Unity Asset Store. The plugin was designed for a first-person view offline game without avatars. The network variables and rules in order to have synchronized changes and motions in the game for all network players were added.

Puzzles

As mentioned above, once the players enter a new room, one of them is assigned as a “solver”, who is the only one that can see and interact with the puzzle (see Fig.6). In order to interact with the puzzle, the "solver" must come to close proximity to it. When they do, a small circular puzzle piece icon appears on the puzzle itself, on which the player can tap. After that, the camera zooms in and gets fixed on the puzzle, so that it is in their front view, and then they can interact and solve it, with the help of the “helpers”. The “solver” has the option to reset or exit the puzzle view at any time. “Helpers”, in order to collect clues needed in solving the puzzle, can interact with some objects in the room as well and explore them in a 3D view mode.

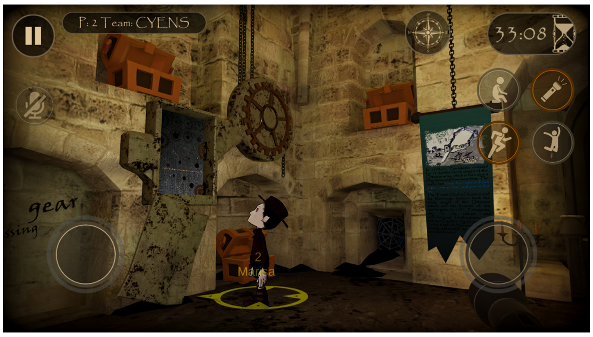


Figure 6: One of the puzzles regarding the construction characteristics of the castle.

The puzzles in the game come in four different forms which are described below (see Fig.7). Logic puzzle: In this type of puzzle, the player must drag and drop some items in their correct place (see Fig.7, A). In the game, several logic puzzles are used in different formats and designs. Some of the items can be photos of areas of the castle, coins, or parts of the castle map which must be placed in the correct order. For example, in the telescope puzzle (see Fig.7, F) the players try to find the correct combination and order of lenses in a telescope and look through it to get the correct keyword.
Lever puzzle: In this type of puzzle, the player must find the correct set of lever positions. A green light indicates their correct position (see Fig.7, B). Also, each puzzle can include several levers that are related to each other, which adds some complexity to it.
Sliding puzzle: In this type of puzzle, there is a board with a picture that is split into several blocks (e.g. 3X3) and a missing block (see Fig.7, C). The player can tap on a block next to the empty place and replace that block's position. The player must find the correct pattern to form the picture by moving the blocks.
Rotary puzzle: The mechanics of this puzzle are similar to the lever puzzle. The player must find the correct position of each part, to form a number code or a word,

but the rotation of the parts is happening vertically (see Fig.7, D) or horizontally (see Fig.7, E). Each part can include 3 to 10 steps.

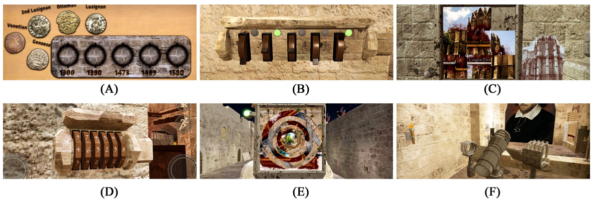


Figure 7: Examples of logic (A), lever (B), sliding (C), horizontal rotary (D), vertical rotary (E), and telescope (F) puzzles in the game.

Clues

When the "helpers" are close to a clue, a circular icon appears on it, which players can tap on and interact with. Some examples of the type of clues that exist in the game are writings on the walls, banners, boards, or books with information.

2.7 Software

The game was developed for Android devices in the Unity software (version 2020.2.0f1) and the programming language C#. The Photon Unity Networking (PUN 2) plugin (version 2.4) was used for the multiplayer features of the game and the Photon Voice 2 (version 2.29) for the voice chat. The open-source software Blender (version 2.9) was used for the modeling, 3D designs, and UV mapping of the castle reconstruction and all the items within it. The UI elements and textures were created in Adobe Photoshop (version 22.5.4), Studio Clip Paint Pro, and Blender.

3 TESTINGS

The whole development process of the application was completed in four more iterations, in addition to the prototype created. The first two iterations used white box testing, with the testers being people who knew the internal structure of the game. The rest of the testing was carried out by black-box testing, with external testers. Throughout testing, the various aspects of the game, like the core game mechanics, the network features, the game structure, and the user experience of the game were evaluated (see Table.1).

3.1 Instruments and Procedure

3.1.1 Instruments

For the black box testing, the questionnaire used included questions regarding the game structure and user experience. The game structure questions were about the puzzle(s) in each room, such as questions about the difficulty of finding the location and clues of the puzzle and included a 5-point Likert scale and an open-ended question, where testers were encouraged to evaluate the puzzles.

	Iteration				
	White-Box testing		**Black-Box testing**		
Game aspect tested (focus)	**Prototype**	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**
Core mechanics	medium	low			
Mechanics integrated in the game		high	high	low	low
Networking	medium	medium	high	low	low
Game structure				medium	medium
User experience				high	high

Table 1: Distribution of the game aspects tested in each iteration and their focus.

The user experience questions included 5-point Likert scale questions relating to the game's visuals, how easy it was to follow the narrative, map usability, and communication with other players. It also included usability questions regarding the avatar selection procedure and navigation. Moreover, open-ended questions were added, where testers could mention if it was clear to them who the solver and helpers were in each room, what aspects could be improved, and what were their favorite and least favorite aspects of the game.

3.1.2 Procedure

Game testers were invited to participate by email invitations that were sent internally and on posts on social media. People could express their interest, their availability and contact the researchers in order to arrange sessions of preferably four people. They were given the link to download and install the game on their mobile device in advance. Sessions could take place remotely or physically in a lab setting. All testers, with the exception of one person in the last iteration, were physically present in the lab during testing.

At the beginning of each testing session, testers were briefed about the premise of the game and that they could communicate with each other through the microphone of their device. They were also informed that, after completing each room's puzzle(s), they would need to complete the corresponding questionnaires. The testers could be either in a separate physical space or the same room, where they could communicate through their devices or face to face but were prohibited from seeing each other's device screen. The problems and difficulties testers were experiencing during testing were observed and written down by the researchers. At the end, when the testers either finished the game or quit, they were asked to complete the user experience part of the questionnaire.

3.2 Prototype

Game prototyping helps developers to check the core mechanics of the game [Ale16a]. During the pre-production phase, a prototype of the game was created, which included the core mechanics of the game with no final models or textures. These core mechanics include the navigation system of the game, the interaction of players with objects and the puzzles, the countdown, the mechanics of the puzzles (logic, lever, sliding, rotary), and the networking feature. White box testing took place with the development team, who knew the internal structure of the game, to test these core mechanics and the networking feature.

3.3 Iteration 1

After the core mechanics and the networking features were tested and working as intended, they were integrated into the game with the rest of the created 3D models. White box testing facilitated the integration of the networking features, the core mechanics, and their applicability in the environment of the game.

Several technical conflicts and bugs were recognized and resolved, mainly regarding network parameters, shared or synchronized objects in the network.

3.4 Iteration 2

For iterations 2 through 4, black box testing was used. After the core mechanics and networking were integrated and working correctly, the researchers proceeded to test mainly the game structure (rooms and puzzles) and user experience of the game.

Two groups, one of four people and the other of three, tested the game from different locations. None of the groups managed to complete the game. During these tests, problems in voice communication and some synchronized objects for the network players were observed and resolved. Also, game instructions were added to the game after this iteration.

3.5 Iteration 3

In the third iteration, three groups of game testers played through the game. The first two groups consisted of two players and the last group of three players. Two of the groups managed to reach the Construction characteristics room (third room) while the other group managed to reach the Music knowledge room (fifth room). Using the feedback from the testers using the questionnaire and the observations made from the researchers, the changes that were made in the application during the first iteration were:

- The Castle visual information and Historical periods rooms were swapped because according to the testing results, the Historical periods room was rated as easier in terms of its puzzle difficulty for the testers. An incremental difficulty escalation was achieved by posing the easiest challenge first.
- More player guidance was added in the Historical periods room to clarify that only one person can see the puzzle and the rest can only see the clues.
- The navigation system on the screen was updated by adding a new joystick on the right side of the screen for the player's head movement (rotation).
- The communication system (voice chat) was improved since technical issues were observed.
- More lighting was added to the environment as users expressed complaints about the visibility in the game.

3.6 Iteration 4

In the fourth iteration, three groups (the first two groups consisted of three players and the last group had four people but one of them left the game before the game finished) played through the game. All teams managed to finish the game. The first team finished it in less than 90 (80min 12 sec) minutes and the second team in less than 120 (116 min 20 sec) minutes and the third team less than 100 minutes (96 min 23 sec), after gaining extra time, by collecting time bonus figurines. The recorded time includes answering the questionnaire as well. Recommendations led to multiple changes:

- Lighting was improved by some adjustments and the player's torch, which was turned on at the beginning of the game, was turned off.
- The number of moving obstacles on the roof after the completion of the Music knowledge room was reduced to balance the difficulty.
- Improvements to the navigation system (jumping power was increased and the sensitivity of the right joystick/head movement was reduced).
- Adding a skip button for the narrations in the game.
- Adding invisible colliders in corners so that players do not get accidentally stuck and more invisible colliders in some walls because some clues were mistakenly accessible from other rooms.

These final changes were implemented in the game, which led to the final version of the game.

4 DISCUSSION AND CONCLUSIONS

The iterative development process of a multiplayer escape room mobile game was presented. The paper described the technical implementations that were integrated into the game, the weather simulation, the baked lightmaps, the single texture used for the 3D model of the castle, the map, the selection algorithm for the "solver" in the rooms and the puzzle mechanics. Game testers, through the use of questionnaires and observations, evaluated the game's functionality and user experience. The iterative design approach revealed persisting issues relevant to the navigation, difficulty scaling, player guidance, and lighting of the game. These issues were identified and resolved with the use of iterative cycles, which helped in incrementally improving the quality of the experience and the game as a whole.

As seen from the results, in the fourth and last iteration of the game, all groups managed to finish the game and their feedback consisted of simple refinements and bug fixes. The initial target of social well-being was achieved, as all the testers found their communication with the other players in this game to be efficient. Additionally, most of the testers emphasized that their favorite parts of the game were the communication and cooperation part, as well as the puzzles. The game itself was found entertaining to play and visually appealing.

The outcomes of this iterative design approach are comparable to the outcomes of similar works on escape room games for socialization and learning, with an example being the escape-room game AScapeD [Ter21a]. A fun and challenging experience was achieved while maintaining equality in the cooperation among players. One of the strongest characteristics that the present game and AScapeD share is the emphasis on turn-taking and the equal contribution of every member of the team in achieving their common goal [Ter21a]. Another example is the work of Thurner-Irmler and Menner [Thu20a], where their twice-tested escape room was accepted by their target group as an interesting way of knowledge transfer.

It is envisioned that the capabilities of the presented game, as a boundary object, will be best utilized towards improving interpersonal and inter-communal relationships. The presented game has the potential of conveying the importance of preserving cultural heritage and the value of cooperation among individuals and communities. These messages lie at the core of the game design and are integrated into both the narrative and the more technical aspects, such as the game mechanics and the 3D models.

5 ACKNOWLEDGMENTS

This work has been funded by the United Nations Development Program (UNDP) through the project 'Digital Heritage Project: Bringing Historic Famagusta to Life for School-Aged Children'.

This work has been supported by the project that has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 739578 and the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

This work has been supported by GET Lab, Department of Multimedia and Graphic Arts, Cyprus University of Technology.

The authors would like to thank the Cyprus Institute for creating and providing a 3D laser scan of the Othello's castle as part of the 'Digital Heritage Project: Bringing Historic Famagusta to Life for School-Aged Children' project.

Authors would also like to thank Domna Banakou for the discussion and input on addressing reviewers' comments and for her suggestions for the camera ready version of the paper.

6 REFERENCES

[Ada14a] Adams, E. Fundamentals of game design. Pearson Education. 2014.

[Ale16a] Aleem, S., Capretz, L.F. and Ahmed, F. Game development software engineering process life cycle: a systematic review. Journal of Software Engineering Research and Development, 4(1), pp.1-30. 2016.

[Ari14a] Aristidou, K., Michael, D. Towards building a diving simulator for organizing dives in real conditions. 22nd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2014. Plzen, Czech Republic. 2014.

[Asu11a] Asuncion, H., Socha, D., Sung, K., Berfield, S. and Gregory, W. Serious game development as an iterative user-centered agile software project. In Proceedings of the 1st International workshop on games and software engineering,pp. 44-47, 2011.

[Bak19a] Bakhsheshi, F.F. December. Serious games and serious gaming in escape rooms. In 2019 International Serious Games Symposium (ISGS) (pp. 42-47). IEEE. 2019.

[Bal07a] Ballagas, R. and Walz, S.P. REXplorer: Using player-centered iterative design techniques for pervasive game development. Pervasive Gaming Applications, 2, p.29. 2007.

[Bal15a] Baldauf, M., Fröhlich, P., Adegeye, F. and Suette, S. Investigating on-screen gamepad designs for smartphone-controlled video games. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 12(1s), pp.1-21. 2015.

[Bra16a] Braad, E., Žavcer, G. and Sandovar, A. Processes and models for serious game design and development. In Entertainment computing and serious games (pp. 92-118). Springer, Cham. 2016.

[Del21a] Del Gallego, N. P., Viaje, C. L., Gerra-Clarin, M. R., Roque, J. M., Non, G. S., Martinez, J. J., and Gana, J. A. A Mobile Augmented Reality Application For Simulating Claude Monet's Impressionistic Art Style. In CSRN. WSCG'2021 -

29. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2021. Západočeská univerzita. 2021.

[Gae00a] Gaertner, S. L., Dovidio, J. F., Banker, B. S., Houlette, M., Johnson, K. M., and McGlynn, E. A. Reducing intergroup conflict: From superordinate goals to decategorization, recategorization, and mutual differentiation. Group Dynamics: Theory, Research, and Practice, 4(1), 98. 2000.

[Gai18a] Gaitanou, M., Charissi, E., Margari, I., Papamakarios, M., Vosinakis, S., Koutsabasis, P., Gavalas, D. and Stavrakis, M. Design of an Interactive Experience Journey in a Renovated Industrial Heritage Site. In Euro-Mediterranean Conference, pp. 150-161. Springer, Cham, 2018.

[Has13a] Hasler, B. S., and Amichai-Hamburger, Y. Online intergroup contact. Oxford University Press. 2013.

[Kat19a] Katsaounidou, A., Vrysis, L., Kotsakis, R., Dimoulas, C. and Veglis, A. MAthE the game: A serious game for education and training in news verification. Education Sciences, 9(2), p.155. 2019.

[Kim15a] Kim, Y.H. and Lee, J.H. Game interface enhancement under smartphone platform focused on touchscreen interaction. Computers and Industrial Engineering, 80, pp.45-61. 2015.

[Kou18a] Koutroumanos, G., and G. Labropoulos. Salamis": A location augmented reality game for local history. In Proceedings of the 11th Pan-Hellenic and International Conference “ICT in Education”, Thessaloniki, Greece, pp. 19-21. 2018.

[Now18a] Nowak, K.L. and Fox, J. Avatars and computer-mediated communication: a review of the definitions, uses, and effects of digital representations. Review of Communication Research, 6, pp.30-53. 2018.

[Sha93a] Shakespeare, W., Mowat, B. A., and Werstine, P. The tragedy of Othello, the Moor of Venice. New York: Washington Square Press. 1993.

[Sta89a] Star, S. L., and Griesemer, J. R. Institutional ecology,translations' and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. Social studies of science, 19(3), 387-420. 1989.

[Sym17a] Symborski, C., Barton, M., Quinn, M.M., Korris, J.H., Kassam, K.S. and Morewedge, C.K. The design and development of serious games using iterative evaluation. Games and Culture, 12(3), pp.252-268. 2017.

[Ter21a] Terlouw, G., Kuipers, D., van't Veer, J., Prins, J.T. and Pierie, J.P.E. The Development of an Escape Room–Based Serious Game to Trigger Social Interaction and Communication Between High-Functioning Children With Autism and Their Peers: Iterative Design Approach. JMIR Serious Games, 9(1), p.e19765. 2021.

[Thu20a] Thurner-Irmler, J. and Menner, M., 2020, November. The development and testing of a self-designed escape room as a concept of knowledge transfer into society. In Joint International Conference on Serious Games (pp. 105-116). Springer, Cham.

[Tre10a] Trepte, S. and Reinecke, L. Avatar creation and video game enjoyment. Journal of Media Psychology. 2010.

[Tsi19a] Tsita, C. and Satratzemi, M. A serious game design and evaluation approach to enhance cultural heritage understanding. In International Conference on Games and Learning Alliance (pp. 438-446). Springer, Cham. 2019.

[Tzi20a] Tzima, S., Styliaras, G. and Bassounas, A. Revealing Hidden Local Cultural Heritage through a Serious Escape Game in Outdoor Settings. Information 12, no. 1 (2020): 10.

[Why15a] Whyte, E.M., Smyth, J.M. and Scherf, K.S. Designing serious game interventions for individuals with autism. Journal of autism and developmental disorders, 45(12), pp.3820-3831. 2015.

[Van18a] Van Ryn, L., Apperley, T. and Clemens, J. Avatar economies: affective investment from game to platform. New Review of Hypermedia and Multimedia, 24(4), pp.291-306. 2018.

[Vas19a] Vaske, K. Megalith Grave Escape: Using escape room game mechanics for cultural heritage sites., University of Skövde, School of Informatics, Dissertation, p. 58, 2019.

[Zil18a] Zilak, M., Car, Z., and Jezic, G. Educational Virtual Environment Based on oculus Rift and Leap Motion Devices. In CSRN. 26. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2017. Západočeská univerzita. 2018.

Human-centered Evaluation of 3D Radial Layouts for Centrality Visualization

Piriziwè Kobina
piriziwe.kobina@imt-atlantique.fr

Thierry Duval
thierry.duval@imt-atlantique.fr

Laurent Brisson
laurent.brisson@imt-atlantique.fr

Anthony David
anthony.david@imt-atlantique.fr

IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France

ABSTRACT

In this paper we propose improvements to the 3D radial layouts that make it possible to visualize centrality measures of the nodes in a graph. Our improvements mainly relate edge drawing and the evaluation of the 3D radial layouts. First, we projected not only the nodes but also the edges onto the visualization surfaces in order to reduce the node overlap that could be observed in previous 3D radial layouts. Secondly, we proposed a human-centered evaluation in order to compare the efficiency score and the time to complete tasks of the 3D radial layouts to those of the 2D radial layouts. The evaluation tasks proposed are related to the central nodes, the peripheral nodes and the dense areas of a graph. The results showed that 3D layouts can perform significantly better than 2D layouts in terms of efficiency when tasks are related to the central and peripheral nodes, while the difference in time is not statistically significant between these various layouts. Additionally, we found that the participants preferred interacting with 3D layouts over 2D layouts.

Keywords

3D Graph Visualization, Centrality Visualization, Graph Layout Evaluation.

1 INTRODUCTION

Centrality measures are topological measures that describe the importance of the nodes in a graph. There has been a lot of works carried out in this topic for network analysis in order to answer the question "Which are the most important nodes in a graph?" [Martino06, Yousefi20]. Other works in graph drawing chose to visually reveal these properties in order to facilitate their exploratory analysis [Brandes11, Raj17]. For example, in graph analytics, some works are interested in understanding and describing the interaction structure by analyzing the topology of the graph [Saqr18, Elmouden20]. Others are interested in identifying and characterizing the nodes that are particularly important in terms of topological position in a graph [Wang17] and how their neighbors are connected to each other [Zhang17].

However, visualizing these measures in 2D could be difficult when the size of the graph is important in terms of the number of nodes and edges. Indeed, there would be a lot of node and edge overlap and edge crossings, which are less of a problem in 3D than 2D [Teyseyre09]. Kobina et al. [Kobina20] therefore proposed new 3D methods based on the 2D radial layouts that highlight the centrality of the nodes [Brandes11] by optimizing the spatial distribution of the nodes. Nevertheless, in 3D some edges could hide others depending on the position of the observer or the 3D layouts and they did not analyze the relevance of their proposed methods.

Our analysis of the state of the art encourages us to pursue this 3D approach by improving the existing one and comparing it to its 2D equivalent. This is why we first propose improvements to the 3D radial layouts by projecting not only the nodes but also the edges onto the visualization surfaces in order to reduce the node overlap. The purpose of our improvements is to provide a better overall view of a complex and large graph than the 3D radial techniques and to reduce the time in exploring and analyzing such a graph. We then propose a human-centered evaluation using a well-known centrality measure in order to compare the efficiency score and the time to complete tasks of the 3D radial layouts to those of the 2D radial layouts. The evaluation tasks are related to the central nodes, to the peripheral nodes and to the dense areas of a graph. The purpose of our evaluation is to show that the 3D radial methods could be better to explore and to analyze graphs whatever the interest, compared to the 2D radial layouts.

This paper is structured as follows: in section 2 we recall some notions about centrality measures in graphs.

https://www.doi.org/10.24132/CSRN.3201.9

We review related work on centrality visualization in section 3. We then present our improvements in section 4 and the human-centered evaluation of these improvements in section 5. In section 6 we present the evaluation results while in section 7 we present our discussion of the various results. In section 8 we present our conclusion and we finally present our future work in section 9.

2 CENTRALITY MEASURES IN GRAPHS

In graph analytics, centrality measures [Saxena20] characterize the topological position of the nodes in a graph. In other words, centrality measures make it possible to identify important nodes in the graph and further provide relevant analytical information about the graph and its nodes.

Some centrality measures, such as degree centrality, can be computed using local information of the node. The degree centrality quantifies the number of neighbors of a node. On the other hand, Betweenness centrality and closeness centrality [Freeman77, Freeman78] use global information of the graph. The betweenness centrality is based on the frequency at which a node is between pairs of other nodes on their shortest paths. In other words, betweenness centrality is a measure of how often a node is a bridge between other nodes. The closeness centrality of a node is the inverse of the sum of distances to all other nodes of the graph.

The importance of a node in a graph can also be characterized by the clustering coefficient [Hansen20] also known as a high density of triangles. The clustering coefficient measures to what extent the neighbors of a node are connected to each other. So, if the neighbors of the node i are all connected to each other, then the node i has a high clustering coefficient.

However, to be able to analyze a graph that could be complex and strongly connected, it is necessary to use visualization tools which, by highlighting these topological properties in the graph, make it possible to visually locate the key nodes of the graph. We therefore discuss, in section 3, related work in graph drawing that make it possible to highlight centrality measures of the nodes in a graph.

3 CENTRALITY VISUALIZATION

Many works in graph drawing made it possible to convey relational information such as centrality measures and clustering coefficient. So, Brandes et al. [Brandes03] and Brandes and Pich [Brandes11] proposed radial layouts that make it possible to highlight the betweenness and the closeness centralities of the nodes in a graph (see Fig.1). In these methods, each node is constrained to lie on a circle according to its centrality value. Therefore, nodes with a high centrality value are close to the center and those of low value are on the periphery.

Dwyer et al. [Dwyer06] also proposed 3D parallel coordinates, orbit-based and hierarchy-based methods to simultaneously compare five centrality measures (degree, eccentricity, eigenvector, closeness, betweenness). The difference between these three methods is how centrality values are mapped to the nodes position. Therefore, for 3D parallel coordinates the nodes are placed on vertical lines; for orbit-based the nodes are placed on concentric circles and for hierarchy-based the nodes are placed on horizontal lines. On the other hand, Raj and Whitaker [Raj17] proposed an anisotropic radial layout that makes it possible to highlight the betweenness centrality of the nodes in a graph. In this method, they proposed to use closed curves instead of concentric circles, arguing that the use of closed curves offers more flexibility to preserve the graph structure, compared to previous radial methods.

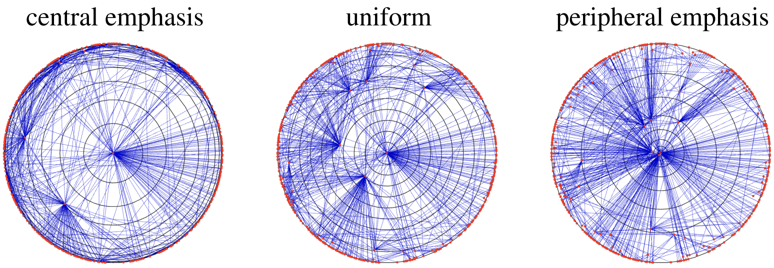
However, it would be difficult to visually identify some nodes that have the same centrality value, compared to the radial layouts. The proposed methods of Dwyer et al. [Dwyer06] make it possible to compare many centrality measures, but it would be difficult to identify the central nodes, compared to that of Brandes and Pich [Brandes11]. On the other hand, 2D methods suffer from lack of display space when one needs to display a large graph in terms of number of nodes and edges.

Kobina et al. [Kobina20] then proposed 3D extensions of the radial layouts of Brandes and Pich in order to better handle the visualization of complex and large graphs (see Fig.2). Their methods consist in projecting 2D graph layouts on 3D surfaces. These methods reduce node and edge overlap and improve the perception of the nodes connectivity, compared to the 2D radial layouts. However, some nodes and edges are less visible depending on the projection surface and edge drawing method. Indeed, the use of straight edges caused most of them to be inside the half-sphere and others to cross the half-sphere. Furthermore, most of the edges are on the surface for the conical projection and outside the surface for the toric projection. Some nodes and edges are then less visible. Therefore, it increases the cognitive effort of an observer. Last, this method has not been evaluated.

However, the solution of Kobina et al. seems the most promising one, so we propose to improve it to overcome its limitations and then to formally evaluate it.

4 IMPROVEMENTS OF THE 3D RADIAL LAYOUTS

In order to reduce node and edge overlap and the cognitive effort in the proposed methods of Kobina et al.

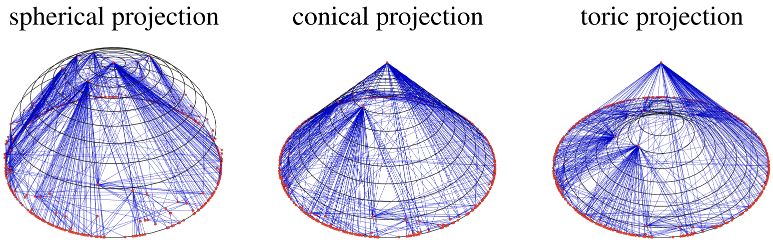


central emphasis

uniform

peripheral emphasis

Figure 1: Betweenness centrality: 2D radial visualization of a graph (419 nodes and 695 edges). Center and periphery are emphasized using transformed radii $r_i' = 1-(1-r_i)^3$ and $r_i' = r_i^3$ ($0 \leq r_i \leq 1$ and $0 \leq r_i' \leq 1$), respectively [Brandes11].



spherical projection

conical projection

toric projection

Figure 2: Betweenness centrality: uniform 3D radial visualization of a graph (419 nodes and 695 edges). The spherical projection spreads out more the peripheral nodes than the central nodes while the toric projection spreads out more the central nodes than the peripheral nodes. The conical projection evenly distributes nodes.

[Kobina20] (Fig.2), we projected the edges onto the visualization surfaces.

Let e be an edge to be projected onto a visualization surface and that connects the nodes j and k, and P_i be any point belonging to e.

$P_i = P_j + (P_k - P_j)t$ where P_j and P_k are respectively the position of the nodes j and k, and $t = i/(n-1)$ where n is the number of control points of the edge e and $i \in \{0, 1, ..., n-1\}$.

4.1 Edge projection onto the cone

In this section, we describe the various steps that are relevant to the proposed method of projecting edges onto the cone:

- we compute the angle θ between the x axis and the z axis of the point to be projected: $\theta = \frac{180}{\pi} atan2(z_{P_i}, x_{P_i})$
- we then rotate by θ about y axis. Let R be the rotation result:
$$R = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$
- we compute the projected point P_p
$P_p = \frac{x_{P_i} x_R + y_{P_i} y_R + z_{P_i} z_R}{||R||} \cdot R$
- we finally compute the altitude $y_{P_p} = 1 - \sqrt{x_{P_p}^2 + z_{P_p}^2}$

4.2 Edge projection onto the half-sphere

Here we describe the projection method of the edges onto the half-sphere:

- we compute the projected point $P_p = \frac{P_i}{||P_i||}$
- we then compute the altitude $y_{P_p} = \sqrt{1-(x_{P_p}^2 + z_{P_p}^2)}$

4.3 Edge projection onto the torus portion

In this section, we describe the projection method of the edges onto the torus portion in four steps:

- we compute the angle θ between the x axis and the z axis of P_i, the point to be projected: $\theta = \frac{180}{\pi} atan2(z_{P_i}, x_{P_i})$
- we then rotate by θ about y axis. Let R be the rotation result:
$$R = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$
- we compute the projected point $P_p = \frac{P_i}{||P_i||} + R$
- we finally compute the altitude of the point: $y_{P_p} = 1 - \sqrt{1-((r-1)(r-1))}$, with $r = \sqrt{x_{P_p}^2 + z_{P_p}^2}$.

Fig. 3 illustrates the result of our projected edges, compared to that of straight edges used in the proposed methods of Kobina et al. [Kobina20] (see Fig.2).

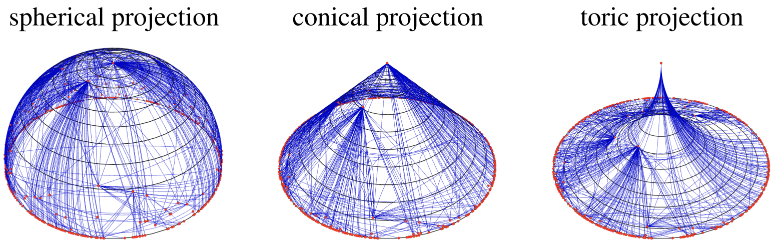


Figure 3: Betweenness centrality: uniform 3D radial visualization of a graph (419 nodes and 695 edges). Edges are projected onto the visualization surfaces, compared to straight edges observed in the proposed methods of Kobina et al. [Kobina20] (see Fig.2).

Therefore, by projecting the edges onto the visualization surfaces, we improved the readability of the graph. Furthermore, there are no edges that cross the visualization surface.

5 EVALUATION

We conducted a human-centered evaluation through a series of tasks performed on generated graphs in order to compare the efficiency score and the time to complete a task of the 3D layouts with projected edges (Fig.3) to those of the 2D radial layouts. We used these 2 metrics to determine whether a kind of visualization performs better or worse than the others with respect to a task. We specifically wanted to answer the following research questions:

Comprehension. What are the effects on comprehension by projecting 2D radial layouts on 3D surfaces?

User Experience. What are the perceived effects of 2D and 3D graph layouts?

5.1 Tasks

Kobina et al. [Kobina20] suggested that the projections of the uniform 2D representation highlight either the center, the periphery, or either moderately the center and the periphery. So we chose these three following tasks that are related to the central nodes, to the peripheral nodes and to the dense areas of a graph:

- **Task 1 (related to the central nodes).** The participants were asked to find one of the nodes that has the greatest degree among the most central node's neighbors.

- **Task 2 (related to the peripheral nodes).** The participants were asked to find one of the least central nodes that has at least two neighbors.

- **Task 3 (related to the dense areas of a graph).** The participants were asked to find one of the nodes of degree at least 3 that has the highest clustering coefficient except 100%.

5.2 Hypothesis

We made hypotheses based on efficiency and speed.

Efficiency: We expected that 3D layouts would perform significantly better in efficiency score than 2D layouts. With respect to task 1, we expected that the participants would score poorly on the 2D that emphasizes the periphery than on the other visualization surfaces. We therefore made the following hypotheses:

H1. The 2D that emphasizes the periphery will perform worse than other layouts when one is interested in the central nodes.

H2. Unlike 2D visualization that emphasizes the periphery, 3D projections that naturally emphasize the periphery (cone and half-sphere) will not be less efficient to perform tasks related to the center.

Then, regarding task 2 we expected that the participants would have bad efficiency score on the 2D that emphasizes the center than on the other surfaces. Moreover, Kobina et al. [Kobina20] suggested that combining the central emphasis with the 3D projections reduces the crushing of the peripheral nodes. So, we made the following hypotheses:

H3. The 2D that emphasizes the center will perform worse than other layouts when tasks are related to the periphery.

H4. Unlike 2D visualization that emphasizes the center, 3D projections that naturally emphasize the center (cone and torus portion) will not be less efficient to perform tasks related to the periphery.

As far as the dense areas (task 3) are concerned, we expected that the participants would perform significantly better on the 3D surfaces than on the 2D layouts. We then made the following hypothesis:

H5. 3D layouts will be better suited for exploring the dense areas of a graph than 2D layouts.

Speed: Whatever the task, we expected that 3D layouts would perform significantly better in speed than 2D layouts. We therefore made the following hypothesis:

H6. The time to complete a task will be longer with 2D layouts than with 3D ones.

5.3 Experimental protocol and measures

We conducted an experimental study using a WebGL version of our graph visualization system because of the Covid-19. Here are links to our experiment for a given configuration starting from 2D `https://anonymnam.github.io/radialvig3dxp1` and for another given configuration starting from 3D `https://anonymnam.github.io/radialvig3dxp2`. The participants could therefore perform the experiment remotely on their own laptop. Kobina et al. [Kobina20] suggested that the combination of the uniform 2D representation and the different projections makes it possible to obtain in addition an emphasis on the center or on the periphery. So in this study, our goal is to show that these 3D methods could be better suited to explore and to analyze graphs whatever the interest (the central or peripheral nodes, the dense areas), compared to the 2D representations. Indeed, since Kobina et al. [Kobina20] optimized the spatial distribution of the nodes and since we projected the edges onto the surfaces, there could be more accurate responses to different tasks. Additionally, the exploration of a graph would be easier, because thanks to these improvements one could better perceive the nodes connectivity. Furthermore, we wanted to analyze the usability of the 3D for exploring and analyzing graphs. On the other hand, we wanted to identify the best layout that could be used to visualize large graphs.

For our experiment we chose to use the betweenness centrality. However, the results of our assessment are not affected by the type of centrality measure used. It will therefore be enough to assess the interest of the proposed methods. We first generated, with the Stochastic Block Model algorithm [Holland83, Stanley19, Lee19], 6 different graphs (250 nodes and 855 edges) that have equivalent topological characteristics ($density = 0.027$ and $diameter = 8$), since it is difficult to find in databases several graphs of the same size with these equivalent topological characteristics. The density of a graph represents the ratio between the number of existing edges and the maximum number of possible edges, while the diameter is the maximum distance between any pair of nodes.

The Stochastic Block Model is a generative model for random graphs which usually produces graphs containing community structure. This means that each node has a fixed community membership, which determines with which probability an edge exists to other nodes [Snijders97]. The model is defined by the number of nodes n, the number of communities C, a probability vector $\alpha = (\alpha_1, ..., \alpha_C)$ specifying the distribution of the nodes on the communities and a symmetric matrix $M \in \mathbb{R}^{CxC}$ with entries in $[0,1]$ specifying the connectivity probabilities [Holland83]. Therefore, the obtained graphs have similar topological characteristics, while being sufficiently different to avoid a learning effect when switching from one to another.

We then built 24 configurations with the various surfaces so that each surface and graph is performed at least once as first, using something similar to the concept of the Latin square [Freeman79, Richardson18]. A Latin square is an n x n array filled with n different symbols in such a way that each symbol occurs exactly once in each row and exactly once in each column. For our configurations, we respected a distribution order between 2D and 3D surfaces so that the running order of a 2D representation corresponds to the one of its equivalent 3D surface. For example, if a configuration starts with the 2D surfaces and the first surface is the one that emphasizes the center, then the first 3D surface will be the torus portion, since it is the one to highlight the most the center. So we make sure that each configuration is tested as many times before as after each of the other configurations. Additionally, half of the participants started the experiment with the 2D followed by the 3D and the second half of the participants with the other way around.

During the experiment and for each task and each surface, we measure an efficiency score and the time spent to complete a task. As the experiment is done remotely, the participants' performance is automatically saved when they validate their responses. Below is how we compute the efficiency score of the participants.

Task 1. Find one of the Nodes that has the Greatest Degree among the most Central Node's Neighbors.

$$score_i = \begin{cases} 100 * (deg_i / deg_{ideal}), & \text{if } d(ctr, i) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where deg_i is the degree of the selected $node_i$. deg_{ideal} is the greatest degree among the central node's neighbors and $d(ctr,i)$ is the shortest distance between the central node and $node_i$. Thus, $node_i$ must be directly connected to the central node, i.e. $d(ctr,i)$ must be equal to 1.

Task 2. Find one of the least Central Nodes that has at least two neighbors.

$$score_i = \begin{cases} 100 * (1 - c_i)/(1 - c_{ideal}), & \text{if } c_{ideal} \neq 1 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where c_i and c_{ideal} are respectively the centrality value of the $node_i$ and that of the ideal node. Furthermore, the score is 0 if the degree of the selected node is less than 2. Indeed, it is easy to check that the degree of the selected node is at least 2. Thus, the score is 0 if the condition is not met. Otherwise, the score varies from 0 at the center to 1 for a node of degree at least 2 and the most on the periphery.

Task 3. Find one of the Nodes of Degree at least 3 that has the Highest Clustering Coefficient except 100%.

$$score_i = \begin{cases} 100 * (ccf_i - ccf_{worst})/k, & \text{if } k > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $k = (ccf_{ideal} - ccf_{worst})$. ccf_i, ccf_{worst} and ccf_{ideal} are respectively the clustering coefficient of the $node_i$, the worst clustering coefficient and the highest clustering coefficient except 100%. The score is therefore 0 if the degree of the selected node is less than 3 or if the clustering coefficient of the selected node is 100%. Otherwise, we compute the score using equation 3.

Since our experiment is done remotely, we organized a video conference for each participant in order to supervise the experiment's process. The experiment consists of a training phase and an evaluation phase. Before starting the training phase, the participants are instructed about the experiment procedure, its environment, navigation and interaction techniques. For example, when the mouse hovers a node, a tooltip shows its clustering coefficient value and its degree. On the other hand, when the participants select a node, its neighbors are highlighted. They are also given the essential notions about graphs in order to ensure that they have the useful knowledge for the experiment. In the training phase, the participants are asked to perform the above tasks on a small graph (the karate club's graph [Zachary77]) and on each surface. Once familiar with the system, they move on to the evaluation phase, but with generated graphs. If the participants are ready to start the training or the evaluation, they click on a start button to see the first task to complete and the next task is automatically displayed after validating the previous task response. At the end of the experiment, the participants complete questionnaires related to the system usability (SUS) [Brooke96] and the user experience.

5.4 Participants

We needed a number of participants that would be a multiple of 24 in order to encounter the same number of these 24 configurations mentioned above. Thus, there were 24 participants (9 female, 15 male) and they were recruited among our colleagues in the laboratory and among students: 50% were between 18 and 25 years old, 37.5% were between 25 and 35, and 12.5% were more than 35 years old. Moreover, most participants had no experience in data analysis and data visualization, but some of them had gaming experience.

6 RESULTS

6.1 User performance

We present here the main results from the analysis of the data collected during our experiment through nonparametric tests using the Kruskal-Wallis and post-hoc tests using the Dunn's method [Dunn64, Sangseok18]. We used nonparametric tests since none of the samples comes from a normal distribution (normality tests were done using the Shapiro-Wilk test). As a reminder, the variables analyzed are the efficiency score and the time for each task and each surface. All data were statistically analyzed using the statistics sub-package of **SciPy** and the **scikit-posthocs** package.

6.1.1 Efficiency score

Task 1: Find one of the Nodes that has the Greatest Degree among the most Central Node's Neighbors.

The nonparametric test showed that there is a statistically significant difference between the visualization surfaces and cannot be due to chance ($statistic = 31.45, p = 10^{-5} < 0.05$). Moreover, the post-hoc test (see table 1) showed that the 2D that emphasizes the periphery had a difference of means. So, we validate hypothesis H1 that the 2D layout that emphasizes the periphery performs worse than other layouts when tasks are related to the central nodes. Furthermore, we validate hypothesis H2 that the 3D projections that naturally emphasize the periphery are not less efficient for performing center-related tasks.

Task 2: Find one of the least Central Nodes that has at least two neighbors.

There is a difference that is statistically significant between the 2D that emphasizes the center and all the other surfaces (see table 2), because the test statistic is 40.31 and the corresponding p-value is $10^{-5} < 0.05$. Thus, we validate hypothesis H3 that the 2D layout that emphasizes the center performs worse than other layouts when a task is related to the peripheral nodes. Moreover, we validate hypothesis H4 that the 3D projections that naturally emphasize the center are not less efficient to perform tasks related to the periphery.

Task 3: Find one of the Nodes of Degree at least 3 that has the Highest Clustering Coefficient except 100%.

The difference between the various surfaces is not statistically significant, since the test statistic is 6.0 and the corresponding p-value is $0.31 > 0.05$. We therefore reject hypothesis H5 that 3D layouts are better suited for exploring the dense areas of a graph than 2D layouts. However, the difference in means (see Fig.4) could lead us to say that the 2D that emphasizes the center performs better than other layouts when tasks are related to the dense areas of the graph, but the statistic analysis failed to demonstrate it.

Based on the efficiency score analysis, the 3D surfaces are well suited for carrying out tasks that are related to the central or the peripheral nodes, since we validated hypotheses H1, H2, H3 and H4. However, we rejected hypothesis H5.

	2D central	2D peripheral	2D uniform	Cone	Half sphere	Torus
2D central	1	10^{-4}***	0.37	0.68	0.42	0.40
2D peripheral	10^{-4}***	1	10^{-3}***	10^{-4}***	10^{-3}***	10^{-3}***
2D uniform	0.37	10^{-3}***	1	0.64	0.93	0.96
Cone	0.68	10^{-4}***	0.64	1	0.70	0.67
Half sphere	0.42	10^{-3}***	0.93	0.70	1	0.97
Torus	0.40	10^{-3}***	0.96	0.67	0.97	1

Table 1: Efficiency score: Task 1: P-values of the post-hoc test using Dunn's method (significant p-values starred (*p < 0.05, **p < 0.01, ***p ≤ 0.001)).

	2D central	2D peripheral	2D uniform	Cone	Half sphere	Torus
2D central	1	10^{-5}***	10^{-2}**	10^{-4}***	10^{-5}***	10^{-3}***
2D peripheral	10^{-5}***	1	0.16	0.69	0.71	0.22
2D uniform	10^{-2}**	0.16	1	0.31	0.08	0.86
Cone	10^{-4}***	0.69	0.31	1	0.44	0.41
Half sphere	10^{-5}***	0.71	0.08	0.44	1	0.11
Torus	10^{-3}***	0.22	0.86	0.41	0.11	1

Table 2: Efficiency score: Task 2: P-values of the post-hoc test using Dunn's method (significant p-values starred (*p < 0.05, **p < 0.01, ***p ≤ 0.001)).

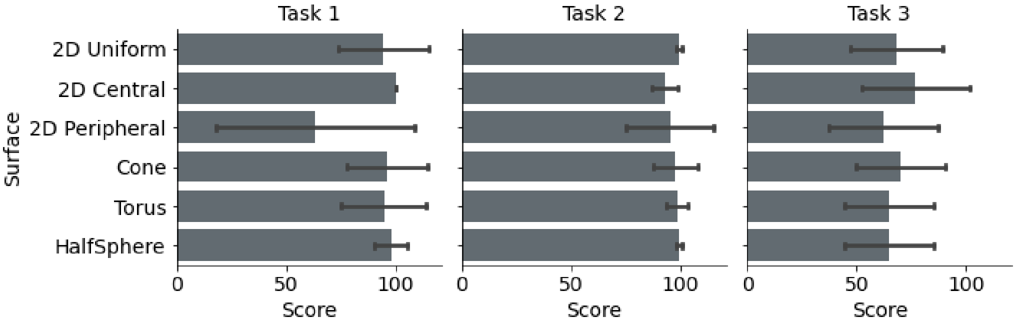


Figure 4: Efficiency score: Means and standard deviations of the efficiency score by task and by visualization surface.

6.1.2 Time

Task 1: Find one of the Nodes that has the Greatest Degree among the most Central Node's Neighbors.

From Fig.5, we could say that the participants spent more time on the 2D that emphasizes the periphery, compared to all other visualization surfaces. However, there is no difference that is statistically significant between the various surfaces ($statistic = 5.990, p = 0.31 > 0.05$). We therefore reject hypothesis H6 that the time to complete a task is longer with 2D layouts than with 3D ones.

Task 2: Find one of the least Central Nodes that has at least two neighbors.

There is no difference that is statistically significant between all the surfaces ($statistic = 1.65, p = 0.90 > 0.05$). We therefore reject hypothesis H6.

Task 3: Find one of the Nodes of Degree at least 3 and that has the Highest Clustering Coefficient except 100%.

We reject hypothesis H6, since the test statistic is 1.04 and the corresponding p-value is $0.96 > 0.05$.

With regard to the time analysis, hypothesis H6 is rejected, since the difference is not statistically significant between the various layouts.

6.2 User experience

As mentioned above (in section 5.3), at the end of the experiment, the participants were asked to complete a questionnaire related to the system usability and to their experience. The usability assessment showed that 3D layouts are more usable than 2D ones, with a score of 81.46 against 75.21 (the usability threshold for a system is 70/100 [Brooke96]). Regarding the participants experience, the participants were asked whether they understood the requested tasks, if they had difficulty interacting with the system, and if they had visual fatigue. The results were that 23 participants over 24 understood the requested tasks, 7 over 24 had difficulty interacting with the system and 7 participants over 24 declared having visual fatigue.

The participants were also asked to specify the surfaces that enabled them to better perform the requested tasks, on the one hand, and to identify the surfaces with which

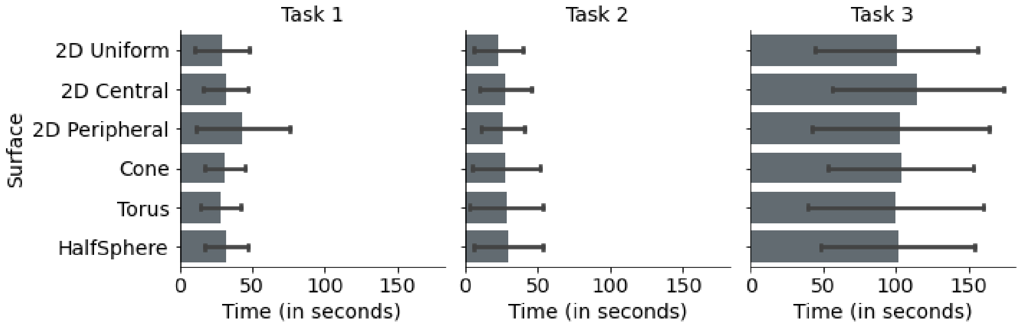


Figure 5: Time: Means and standard deviations of the time by task and by visualization surface.

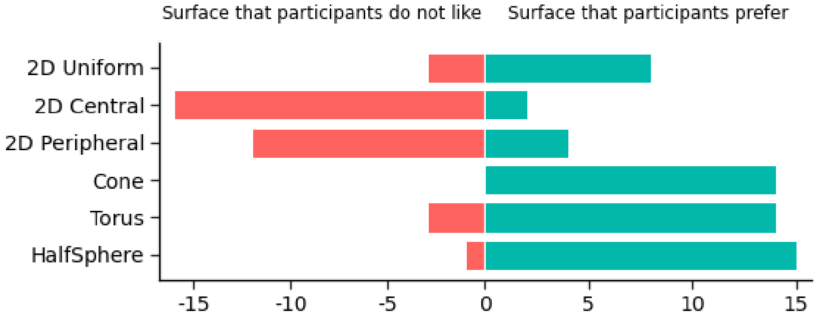


Figure 6: Distribution of user preferences for liking (in green) and disliking (in red) visualization surfaces.

they had difficulty completing the requested tasks, on the other hand. Based on their feedback, 3D surfaces have significantly contributed to the successful completion of the various tasks, compared to the 2D representations (uniform 2D, the 2D that emphasizes the center or the periphery). Fig.6 illustrates the distribution of user preferences for liking and disliking visualization surfaces. It shows that the participants significantly prefer 3D layouts when performing tasks. Moreover, the 2D that emphasizes the center and the one that emphasizes the periphery alone total 80% of the dislike votes while the cone makes 0% dislike.

7 DISCUSSION

Some nodes would be less visible with the use of the straight edges in the proposed methods of Kobina et al. [Kobina20]. Indeed, combining the peripheral emphasis and the projection of the nodes and edges on the half-sphere or on the torus portion, some intermediate nodes would be less visible due to the type of surface, unlike the conical projection. Furthermore, with uniform projections, some nodes and edges would be less visible in the dense areas according to the projection surface. Thus, projecting the edges onto the visualization surface, we reduced the overlap of the nodes and the edges, and we therefore improved the overall readability of the graph.

As far as our evaluation is concerned, we expected that each 3D visualization could be the best for one of the tasks, hence the interest of switching from one type of projection to another depending on the task to be carried out. However, we validated hypotheses H1, H2, H3, H4, since the statistic test results showed that there are differences in efficiency score when tasks are related to the central and peripheral nodes.

Indeed, these results made it possible to validate hypotheses H1 that the 2D that emphasizes the periphery is the worst of the surfaces to visualize the center, and H3 that the 2D that emphasizes the center is the worst of the surfaces to visualize the periphery with respect to the efficiency score of tasks 1 and 2. Moreover, we validated hypotheses: 1) H2 that 3D projections that naturally emphasize the periphery (cone and half-sphere) are not less efficient to perform tasks related to the center; 2) H4 that 3D projections that naturally emphasize the center (cone and torus portion) are not less efficient to perform tasks related to the periphery, always regarding the efficiency score of tasks 1 and 2.

On the other hand, we rejected hypotheses H5, since we were not able to prove that 3D layouts are better suited to explore the dense areas of a graph than 2D layouts. We also rejected hypothesis H6 that the time to complete a task is longer with 2D layouts than with 3D ones, because there is no difference that is statistically significant. We could therefore say that the 2D versus 3D debate still persists [Cliquet17]. However, the participants' feedback showed that the 3D surfaces could be well suited for completing the various requested tasks successfully, compared to the 2D surfaces. Moreover, the system usability assessment showed that 3D is above 2D, since its score is 81.46 and the one of 2D is 75.21.

Table 3 summarizes which hypotheses have been validated (✓) or rejected (✗) for which task and for which measure.

Metrics	Efficiency					Speed
Hypotheses	H1	H2	H3	H4	H5	H6
Task 1	✓	✓				✗
Task 2			✓	✓		✗
Task 3					✗	✗

Table 3: Summary of evaluation hypotheses.

8 CONCLUSION

Our improvements of the edge drawing for 3D radial layouts lead to a better usability of these layouts. These improvements consisted in projecting the edges onto each visualization surface in order to reduce the node and edge overlap. The human-centered evaluation we conducted showed that these 3D layouts can be more efficient than 2D layouts for tasks that are related to the central and peripheral nodes, even if we were not able to say that the time to complete a task is shorter with 3D layouts than with 2D layouts. Additionally, the participants significantly preferred 3D layouts, because they had a better feeling on the 3D when carrying out the requested tasks, compared to 2D layouts. Thus, adding a third dimension to the 2D radial layouts improves the user experience.

9 FUTURE WORK

In the future, we will also study in detail the results obtained with large graphs in order to check whether current trends are confirmed. Specifically, we will check if 3D would perform better than 2D on a 75-inch 4K screen and if immersive 3D would perform better than 3D on a 75-inch 4K screen. We are already conducting a human-centered evaluation with large graphs using a 75-inch 4K screen and in virtual reality. Moreover, we projected the 2D views on other types of 3D surfaces (a parabola, a Gaussian, a hyperboloid and a square root). Thus, we will study in more details the results of these contributions in order to identify the most appropriate approach or combination of approaches that could be used to visualize large and complex graphs.

Furthermore, when a graph contains several thousands of edges, the visualization often suffers from clutter (see the left image of Fig.7). The graph can therefore be almost impossible to analyze. Thus, in order to declutter graphs in the proposed methods of Kobina et al. [Kobina20] and highlight the connectivity between groups of nodes, we will exploit the computer graphics acceleration techniques and the kernel density estimation edge bundling algorithm [Hurter2012]. Fig.7 illustrates the result of a graph which was generated using Stochastic Block Model algorithm presented in section 5.3. It is thus possible to see how groups of nodes are connected to each other with a bundled graph. However, we lose the detailed connectivity of a node (for instance, edges between a node and its neighbors). It could therefore be useful to combine the bundled and the unbundled edges for further analysis if one would need to switch between detailed and bundled views.

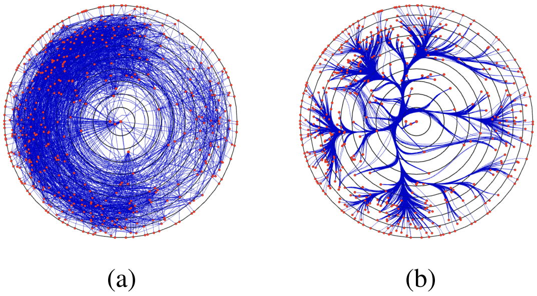


Figure 7: Top view from the cone of a generated graph (500 nodes and 3294 edges): (a) unbundled and (b) bundled using KDEEB algorithm proposed by Hurter et al. [Hurter2012]. Edge bundling makes it possible to declutter the graph.

10 ACKNOWLEDGMENTS

This work has been funded by a grant from Brittany region, France.

11 REFERENCES

[Brandes03] Brandes, U., and Kenis, P., and Wagner, D. Communicating centrality in policy network drawings. IEEE Transactions on Visualization and Computer Graphics 9, pp.241-253, 2003.

[Brandes11] Brandes U., and Pich, C. More flexible radial layout. Journal of Graph Algorithms and Applications 15, pp.151-173, 2011.

[Brooke96] Brooke J. SUS: A quick and dirty usability scale. Usability Evaluation in Industry 189, 1996.

[Cliquet17] Cliquet, G. and Perreira, M. and Picarougne, F. and Prié, Y. and Vigier, T. Towards HMD-based Immersive Analytics. In Immersive Analytics workshop of IEEE VIS 2017, 2017.

[Dunn64] Dunn, O.J. Multiple Comparisons Using Rank Sums. Technometrics 6, No.3, pp.241-252, 1964.

[Dwyer06] Dwyer, T., and Hong, S., and Koschützki, D., and Schreiber, F., and Xu, K. Visual analysis of network centralities. In Asia-Pacific Symposium on Information Visualisation 60, Tokyo, Japan, pp.189-197. Australian Computer Society, 2006.

[Elmouden20] El Mouden, Z.A., and Taj, R.M., and Jakimi, A., and Hajar, M. Towards Using Graph Analytics for Tracking Covid-19. Procedia Computer Science 177, p.204-211, 2020.

[Freeman79] Freeman, G.H. Complete Latin Squares and Related Experimental Designs. Journal of the Royal Statistical Society. Series B (Methodological) 41, No.2, pp.253-262, 1979.

[Freeman77] Freeman, L.C. A Set of Measures of Centrality Based on Betweenness. Sociometry, 40, No.1, American Sociological Association, Sage Publications, Inc., pp.35-41, 1977.

[Freeman78] Freeman, L.C. Centrality in social networks conceptual clarification. Social Networks 1, No.3, pp.215-239, 1978.

[Hansen20] Hansen, D.L. and Shneiderman, B. and Smith, M.A. and Himelboim, I. Chapter 3 - Social network analysis: Measuring, mapping, and modeling collections of connections. In Analyzing Social Media Networks with NodeXL (Second Edition), pp.31-51. Morgan Kaufmann, 2020.

[Holland83] Holland, P.W., and Laskey, K.B. and Leinhardt, S. Stochastic blockmodels: First steps. Social Networks 5, No.2, pp.109-137, 1983.

[Hurter2012] Hurter, C. and Ersoy, O. and Telea, A. Graph Bundling by Kernel Density Estimation. Computer Graphics Forum 31, pp.865-874, 2012.

[Kobina20] Kobina, P., and Duval, T., and Brisson, L. 3D Radial Layout for Centrality Visualization in Graphs. In Augmented Reality, Virtual Reality, and Computer Graphics. AVR 2020, Lecce, Italy, Proceedings, Part I 12242, pp.452-460. Springer-Verlag, Berlin, Heidelberg, 2020.

[Lee19] Lee, C., and Wilkinson, D.J. A review of stochastic block models and extensions for graph clustering. Applied Network Science 4, No.1, 2019.

[Martino06] Martino, F., and Spoto, A. Social Network Analysis: A brief theoretical review and further perspectives in the study of Information Technology. PsychNology Journal 4, pp.53-86, 2006.

[Raj17] Raj, M., and Whitaker, R.T. Anisotropic Radial Layout for Visualizing Centrality and Structure in Graphs. In Graph Drawing and Network Visualization 10692, Boston, MA, USA, pp.351-364. Springer International Publishing, 2017.

[Richardson18] Richardson, J.T.E. The use of Latin-square designs in educational and psychological research. Educational Research Review 24, pp.84-97, 2018.

[Sangseok18] Sangseok, L., and Kyu, L.D. What is the proper way to apply the multiple comparison test? Korean J Anesthesiol 71, No.5, pp.353-360, 2018.

[Saqr18] Saqr, M., and Fors, U. and Nouri, J. Using social network analysis to understand online Problem-Based Learning and predict performance. PLOS ONE 13, No.9, pp.1-20, 2018.

[Saxena20] Saxena, A., and Iyengar, S. Centrality Measures in Complex Networks: A Survey. ArXiv, abs/2011.07190, 2020.

[Snijders97] Snijders, T.A., and Nowicki, K. Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure. Journal of Classification 14, No.1, pp.75-100, 1997.

[Stanley19] Stanley, N., and Bonacci, T., and Kwitt, R., and Niethammer, M., and Mucha, P. Stochastic Block Models with Multiple Continuous Attributes. Applied Network Science, 4, pp.1-22, 2019.

[Teyseyre09] Teyseyre, A.R., and Campo, M.R. An Overview of 3D Software Visualization. IEEE Transactions on Visualization and Computer Graphics 15, No.1, p.87-105, 2009.

[Wang17] Wang, J., and Hou, X., and Li, K., and Ding, Y. A novel weight neighborhood centrality algorithm for identifying influential spreaders in complex networks. Physica A: Statistical Mechanics and its Applications 475, pp.88-105, 2017.

[Yousefi20] Yousefi Nooraie, R., and Sale, J.E.M., and Marin, A., and Ross, L.E. Social Network Analysis: An Example of Fusion Between Quantitative and Qualitative Methods. Journal of Mixed Methods Research 14, No 1, pp.110-124, 2020.

[Zachary77] Zachary, W.W. An Information Flow Model for Conflict and Fission in Small Groups. Journal of anthropological research 33, pp.452-473, 1977.

[Zhang17] Zhang, H., and Zhu, Y., and Qin, L., and Cheng, H., and Yu, J.X. Efficient Local Clustering Coefficient Estimation in Massive Graphs. In Database Systems for Advanced Applications, pp. 371-386. Springer International Publishing, Cham, 2017.

Chatbot Explorer: Towards an understanding of knowledge bases of chatbot systems

Alrik Hausdorf[1]
hausdorf@informatik.uni-leipzig.de

Lydia Müller[2,3]
lydia@informatik.uni-leipzig.de

Gerik Scheuermann[1]
scheuermann@informatik.uni-leipzig.de

Andreas Niekler[4]
aniekler@informatik.uni-leipzig.de

Daniel Wiegreffe[1]
daniel@informatik.uni-leipzig.de

[1] Image and Signal Processing Group, Leipzig University

[2] Institute for Applied Informatics (InfAI), Leipzig

[3] Leipzig University

[4] Computational Humanities, Leipzig University

ABSTRACT

A chatbot can automatically process a user's request, e.g. to provide a requested information. In doing so, the user starts a conversation with the chatbot and can specify the request by further inquiry. Due to the developments in the field of NLP in recent years, algorithmic text comprehension has been significantly improved. As a result, chatbots are increasingly used by companies and other institutions for various tasks such as order processes or service requests. Knowledge bases are often used to answer users queries, but these are usually curated manually in various text files, prone to errors. Visual methods can help the expert to identify common problems in the knowledge base and can provide an overview of the chatbot system. In this paper, we present Chatbot Explorer, a system to visually assist the expert to understand, explore, and manage a knowledge base of different chatbot systems. For this purpose, we provide a tree-based visualization of the knowledge base as an overview. For a detailed analysis, the expert can use appropriate visualizations to drill down the analysis to the level of individual elements of a specific story to identify problems within the knowledge base. We support the expert with automatic detection of possible problems, which can be visually highlighted. Additionally, the expert can also change the order of the queries to optimize the conversation lengths and it is possible to add new content. To develop our solution, we have conducted an iterative design process with domain experts and performed two user evaluations. The evaluations and the feedback from our domain experts have shown that our solution can significantly improve the maintainability of chatbot knowledge bases.

Keywords

application motivated visualization, chatbot, chatbot knowledge base, chatbot maintenance, decision tree.

1 INTRODUCTION

In recent years, advances in NLP (Natural Language Processing) have led to the development of more and more applications for text-based dialog systems, so-called chatbots. A user can interact with these programs using textual queries, and the chatbot's responses are then also delivered in text form, resulting in conversations with the chatbot. Possible use cases for chatbots include ordering processes, service requests, or the control of a smart home. Especially companies try to reach their customers by messenger with the help of a chatbot to reduce their costs. With the high request for systems that are easy to configure and cheap enough to be used by middle or small-sized companies, many tools were created. Chatbots are also used in many households to control the smart home, a well-known example being the voice-based Alexa system developed by Amazon.

Initially, the chatbot matches the user's input to a text item, the so-called *intent*, in its knowledge base. This part of the chatbot system is done by an internal NLP unit. Since these parts of the chatbot system are configured in a complex and language-specific way, most systems do not allow any modification of the configuration here. Next, the bot needs information about how to react to what is expected in response to the user's

https://www.doi.org/10.24132/CSRN.3201.10

request. Those reactions are called *actions* and can have very different functionalities.

The most common reaction of a chatbot is a text-based response. More complex actions could be to activate a light in the living room, select the pizza that the user wants to order, or register for an academic conference. If the bot cannot determine the intention of the user the bot can get the required information by further inquiries. For the pizza example, it is possible that the bot needs information about the cheese topping, which was not selected by the user beforehand. These conversations with the user are so called stories in the context of a chatbot. Stories are connected intents and actions that are configured by the maintainer to reach a specific goal.

Large companies like Google[1] or Microsoft[2] have complete systems to address all of these parts for their own business. However, there are many small companies with different requirements to make chatbots easier to run or configure. For example, the chatbot system s [3] tries to fill the gap of open-source software for this purpose.

However, there are currently still limitations in the use of chatbots. On the one hand, there are open issues in the area of natural language understanding to determine the user requests. On the other hand, it can be tedious to create a knowledge base for the chatbot. Both areas offer great potential for further improvements. In the current state, chatbot systems are mostly trained for one specific topic or task and new knowledge must be added manually. Furthermore, the maintenance tasks of chatbots are currently mostly done by domain experts and are barely assisted by visualizations. Thus, the use of chatbots is only possible productively with greater effort, since malfunctions in commercially used chatbots can lead to lost sales. However, small and medium-sized companies in particular cannot usually afford specialized employees for the support of chatbot systems. Therefore, these companies can only resolve problems or add new content with a great amount of time and effort.

In this paper we propose *Chatbot Explorer*, a system that enables users to get an overview of the chatbot knowledge base, to detect common errors and performance issues, as well as solve those issues with the assistance of visualizations.

Our contributions in this paper are:

- an interactive tree representation of knowledge bases for chatbots
- a semi-automatic detection of errors and problems in these knowledge bases
- interaction mechanisms for modifying and adding content to a chatbot
- a two-stage evaluation of the application using the System Usability Scale (SUS) standard

While creating the system the authors were in recurring contact with the domain experts. After developing the first version, an evaluation with 14 people was done to get valuable feedback on the system. That feedback was incorporated then to develop an improved version of *Chatbot Explorer*. This version was again evaluated with a larger group of participants to show the usability of the developed visualizations and interactions.

2 PROBLEM FORMULATION

For the setup of a chatbot, the maintainer must create a knowledge base using story elements. This mostly tedious task can lead to various errors so that the bot does not react correctly or unpredictably to user requests.

In cooperation with domain experts, we have identified the various sources of problems for this work and have developed mechanisms to detect and fix them. Our cooperation partner is a company that has developed and operated various chatbots for customers in the last years. At the beginning of the cooperation, we analyzed and tested their existing software systems. Additionally, we conducted interviews with the domain experts from their staff. During these interviews, the experts presented their tasks, demonstrated known problems during the setup of chatbots, and discussed possible improvements.

Together with the company a list of maintenance tasks, requirements, and problems that could occur, were identified. In the following, we describe the relevant tasks which are required to set up and maintain a chatbot:

T1 Add new knowledge to the chatbot or modify it to offer more information.

T2 Finding and fixing errors in the stories of a chatbot.

T3 Modification and reordering of the stories to optimize the conversations with the users so that they reach their desired objective faster.

Different chatbot systems mostly include methods to modify the knowledge base (T1). Besides those standard tasks, the experts are interested in a tool that helps with Tasks T2 and T3, too. In order to ensure that the tool can fulfill all three tasks, we have agreed with the domain experts on the following definition for a knowledge base for chatbots: A knowledge base can be reduced to the possible stories between the bot (actions) and a user (intents) [19]. Accordingly, a story is a goal-oriented conversation, for example, a user can order a specific product using the chatbot. A knowledge base of the chatbot then consists of several stories, for example, one

[1] https://cloud.google.com/dialogflow

[2] https://azure.microsoft.com/en-us/services/bot-services/

[3] https://rasa.com/

story for each product of the entire offer of a restaurant. Therefore, the maintenance of the knowledge base is equivalent to the management of the possible stories of the chatbot.

Following the structure of [8] the system requirements were collected together with the experts. In this process, we have identified additional requirements for the system, which are necessary for the successful processing of the tasks. First, the expert must gain an overview of the knowledge base of the chatbot in order to identify relations between the individual elements. On the other hand, knowledge bases can become very large, so the expert should be supported in identifying searched elements. This is especially important for the identification of structural errors, which should be detected (semi-) automatically.

As constraints for the visual system, the following were identified with the domain experts:

C1 The order in which the stories are stored in the knowledge base should not affect the visualization. Only the content of the stories should influence the visualization.

C2 Each story should be visualized as a continuous path.

3 RELATED WORK

Started in the 1960s by Joseph Weizenbaum [26], chatbots aim to entertain the user. One of the ambitious goals of chatbots is passing the Turing-test [24], by communicating like a real person. With the recent advances in the field of NLP, chatbot systems have gained a high amount of interest recently. Modern chatbots are used for instance as a replacement for FAQ systems or to enhance service for customers.

The wide range of possible applications for chatbots leads to a high amount of case studies. However, in this paper, we focus on surveys showing different aspects of chatbots. An overview of chatbot design techniques was created by Abdul-Kader et al. [1] in 2015 analyzing 9 selected studies. The survey by Chaves et al. [5] covers different interactions of bots. Maroengsit et al. [14] provides a survey about different evaluation methods of chatbots and provides a wide overview of chatbot systems. In 2019 Ch'ng et al. [6] give a summary of chatbots that are reported in the literature. Johannsen et al. [9] investigated 14 commercial chatbot providers in 2018 with a focus on supporting customer interaction. An overview of the training of a chatbot and how they perform on different models and methods of understanding the user is given by Csaky [7]. Klopfenstein et al. [12] give an overview of different conversational interfaces, patterns, and paradigms. These works highlight exemplary the state of the art for the deployment and usage of chatbots. However, they are not covering the area of developing and maintaining a knowledge base for a chatbot efficiently.

In the non-academic area, different systems analyze chatbot system data. The three most popular tools in this area are botanalytics [4], Virtual Agent Analytics by chatbase [5] and dashbot.io [6]. All three tools use the logs of chatbots and the collected metadata to provide insights into customer usage. All tools generate a tree structure for the analysis from the knowledge base and visualize the paths that users can follow in the conversations. However, these systems have been developed to show the use of chatbots and less to correct errors.

Several commercial tools are available to help the maintainer with creating and running chatbot systems. One of the largest open-source toolset for that purpose is RASA. With RASA it is also possible to visualize the chatbot knowledge base. For this purpose, the intents and actions are represented as nodes in a DAG (directed acyclic graph). This DAG is then represented using standard algorithms, but this is only suitable for smaller knowledge bases.
Recently, Yaeli and Zeltyn [27] presented a system to identify and fix problems and failures in chatbots. Their system can identify errors in productively deployed systems based on the conversation processes, so that the maintainer can then adjust the knowledge base. However, this requires a larger amount of conversational logs, which is why we chose a different approach for our system.

The usage of visualizations to gain insights into complex knowledge bases can help to understand data, communicate structures, and find the right decisions. Baumeister et al. [2] proposed a tree view-based visualization to show possible paths by creating nodes with different configurations. Renaud et al. [21] show why knowledge base visualization can help various target groups to understand the decisions of a system as well as a framework to increase the power of knowledge visualization. She highlights, the question "For whom?" has a high impact on the system. The work of Jonassen [10] shows that for problem-solving annotated directed graphs are efficient to assist the user with their already known mental representation of a problem. Neumann et al. [16] have used digital mind maps to structure knowledge for different audiences.

The visualization of conversational data is a common problem and was addressed by various works over time. Pascual-Cid et al. [18] developed a hierarchical tool to explore asynchronously online discussions using a hierarchical, radial layout. In the year 2012 Jyothi et al. [11] developed a similar visual tool to analyze asyn-

[4] https://botanalytics.co/
[5] https://chatbase.com/
[6] https://www.dashbot.io/

chronously students' interactions in online communications using radial, directed graphs. The work of Wattenberg et al. [25] uses a collapsible tree view to analyze a large number of conversations. The hierarchical representation of conversational data was proposed by Newman [17] using an icicle plot. All of these representations work with historically collected real-live conversational data. Her main goal is to make large conversations explorable and to give an overview of the conversation. The case of managing those stories or changing the structure of that conversations was not intended due to the used dataset.

None of the described approaches can fully cover all tasks and constraints as they are domain-specific. In the context of our work, we have therefore extended them to apply them to our problem setting.

4 DATA PROCESSING

Knowledge bases of chatbots can exist in different types as shown in Section 3. For *Chatbot Explorer*, we selected the stories as a representation of the data, since they are common for as many chatbot systems as possible.

The system works with the standard configuration files of RASA. This configuration consists of at least three files: domain.yml, story.md, and nlu.md. A specification of those parts of the files that are necessary for the system together with an example can be found in the supplemental material. The used configuration is, in general, the main source for a list of stories, intents, actions, and some meta-information like the messages that are used for the elements.

Chatbot Explorer was tested during the development with a self-created dataset that is supposed to represent a chatbot ordering a menu from a restaurant. We have generated this dataset from the offer of a large restaurant franchise. Therefore, each pizza and pasta was manually disassembled into their features. From the combination of those features, the first intent of each story was the type of the dish (intent "type pasta" or intent "type pizza"). Each story ends with an action, where the bot tells the user which menu s/he has ordered. The dataset contains 19 stories with 33 intents and 27 actions.

5 METHODS

Following the task analysis with the experts, we started with the development of visualizations that provide an overview of a given knowledge base. In the following, we investigated methods to interact with the visualizations and the underlying data. Afterwards, an automatic data analysis was conducted to highlight common problems in chatbot knowledge bases using structural information. These visualizations were created with the mantra of Shneiderman "Overview first, zoom and filter, then details-on-demand" [23] in mind.

Chatbot Explorer uses a tree visualization as an overview as well as a highly detailed view. The expert can apply filters, set highlights, and can zoom into the data. Details like intent examples are shown on demand. This tree representation was then adapted in the following process in order to improve the representation of the knowledge base.

5.1 Tree Layout

The action of a user of a chatbot in a conversation is a central point to decide what information is given to the user at the end. In general, the user requests the chatbot, whereupon the chatbot precipitates this request by asking questions to be able to execute the desired goal, for example ordering a product. These queries of the chatbot should thus be as precise and error-free as possible, so that the conversation is efficient. Therefore, the amount of decisions of the user is key to the performance of the chatbot. For that, we introduce the concept of story levels. If two conversations or parts of those conversations are on the same level, the number of decisions made by users is equal. Decisions where the user can not decide what answer s/he has to give, do not increase the level number. An example is when the chatbot asks for the user's customer number, the user cannot make their own decision for their answer.

Using a tree or graph for the representation of conversation paths is a common approach while using chatbots as already pointed out in Section 3. Existing approaches use an empty graph or tree and let the user manual fill with the desired data. However, our approach needs a suitable mapping from the knowledge base to the tree representation to work with an already defined knowledge base. This representation is created using a basic tree aggregation followed by several optional optimizations.

For the basic aggregation step, each story is integrated into the current tree. Each intent and action is represented as a node and each connection between them is stored as an edge. If a path in the tree already exists that is equal to a part of the added story, all shared nodes get only a reference to the newly added story. One common example is the greeting part of most stories, which merges into one node in the tree.

However, this approach can lead to unnecessarily sequential paths in the tree representation, since a given intent sometimes have only one possible action (e.g., asking for the name and address of the user). Therefore, we added a step optimization. The first approach is to combine intents and the following actions since this follows the idea that a question of the bot is a reaction to the text of the user. A second aggregation step combines in following all nodes that do not allow decisions of the user or the bot. Both aggregations decreased the drawing space of the tree significantly.

To improve the identification of the ending elements of a story, they are highlighted in the visualization. In addition, they can be separated from aggregated intent/action blocks by using a filter. Before the positions of the tree elements is calculated, the children of each node are ordered descending by the number of children. While the order only depends on the number of children after the aggregation, all stories are treated equally with respect to Constraint C1.

We improved the positioning of the nodes iteratively to the final, presented version. In the beginning, we used the default algorithm implemented in d3-tree. This algorithm uses the Reingold-Tilford [20] algorithm with an improvement of Buchheim et al. [4] to run in linear time. Since the nodes can differ in their height we switched to d3-flextree[7] that is size aware. However, the participants of the first evaluation stated, that the space-saving approaches of both algorithms sometimes make it unlikely to find a story end (leave). Figures of this first version can be found in the supplemental material as reference. Therefore, we decided to reimplement the layout on our own to achieve a better positioning of the nodes. This position algorithm starts to position all leaves on a horizontal line. Afterwards, the parent elements are positioned centered above all children. However, this positioning algorithm can create very large distances between children and parents, e.g., if individual children need a lot of height. To overcome this, we align the nodes afterward locally inside the level and/or globally regarding all children.

Each node (e.g., the element with the blue border in Figure 1 (2)) contains several elements that can be intents, actions, or the start element. A legend of the different elements is shown in Figure 1 (c). To distinguish the elements inside, the node are color-coded regarding their type. Additionally, each node is assigned to a level, which is indicated by the colored and labeled background area. The number on the top of each node represents the number of stories that share this path and the node. Below the edge of the node, the number of outgoing paths is shown.

Showing too much information at the same time can lead to visual clutter and reduce the readability of visualizations. Therefore, the visualization allows configuring which information should be shown for each element. Available information is provided as icons, representing the type of element (intent = user-icon, action = roboter-icon, start = play-icon), the name of the element, and a text example that could be used for that element.

5.2 Interaction

The following interactions with the tree view were identified during the requirement process or were suggested in the evaluations.

Pan and Zoom: In order to enable the user to explore the knowledge base it is possible to pan and zoom the view. If the zoom level reaches a certain level, the texts inside the elements are hidden and all boxes are transformed to squares with the same height as width. This "large-elements" drawing stretches the tree in the y-direction and helps to gain an overview even with larger trees on smaller screens.

Details on Demand: The expert may need more information about a specific element in the tree than is shown in the node. Therefore, we provide a tooltip mechanism that appears if the mouse hovers an element. To interact with the knowledge base the expert can select an element by clicking it (done so in Figure 1 (2)). *Chatbot Explorer* then offers the expert the following options: Modify the item, create a new story, and remove a story. Additionally, the highlighted element is shown at the bottom of the Treeview as a single element (see Figure 1 (2)).

Possible Problem Detection: Structural issues (see Task T2) are a serious problem that should be fixed before a knowledge base is used in a productive environment. Structural problems are often missed (see Section 2), causing problems after deployment. Therefore, *Chatbot Explorer* contains functionalities to detect and present these problems. The interface that summarizes possible problems within a given knowledge base is shown in Figure 1 (b). It shows the selected type, the currently selected problem, and how many problems were detected. The problem of "Undecidable paths" represents the error that after an intent of a user more than one possible reaction of the bot is possible. This is a problem since a non-deterministic answer can result in side effects, e.g., the bot has to decide, which way to follow. While selecting one of those "Undecidable paths" the parent-node, the children, and the paths between them are highlighted with a red border. That highlighting mechanism allows the user to use the overview to spot the position of the problem easily. The selected error can be found in Figure 1 (2). In the example, the intent "cheese_mozzarella" is followed by two actions: "utter_ask_topping_meat" and "utter_tell_pasta_cheese". If this problem occurs a chatbot engine randomly selects an action. Nevertheless this can lead to unexpected results. One possible solution to these errors can be the reordering of the queries the bot sends to the user or the addition of new queries to a story. To select a solution, however, a decision of the expert is necessary. Another common problem is the "Unnecessary step". It is defined as an action that asks something, followed by only one possible answer of the user (intent). For example, the bot asks the user for the cheese topping ("utter_ask_cheese") and the only answer that is possible is the selection of "cheese_mozzarella" (see Figure 1, bottom node on the left side in (2)). The bot system stays at that point of the story until the cor-

[7] https://github.com/klortho/d3-flextree

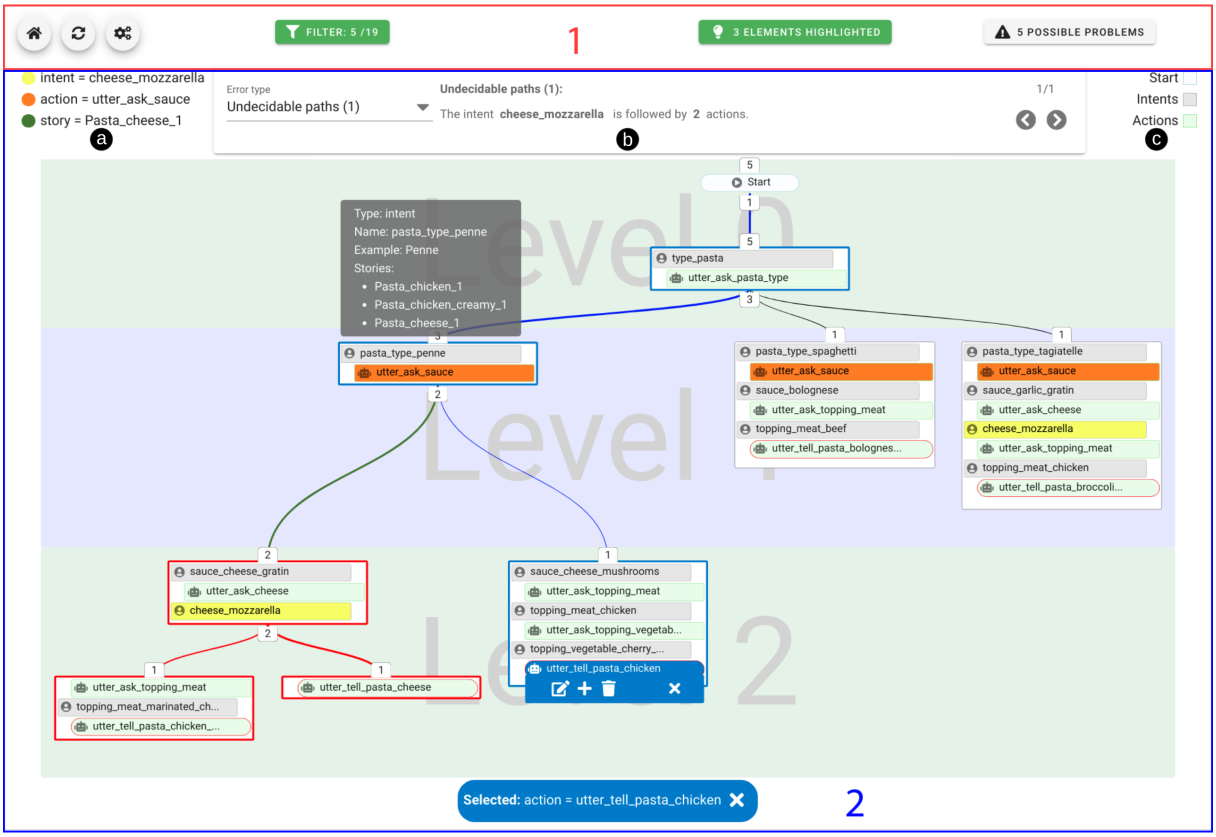


Figure 1: The main interface of the *Chatbot Explorer*. The interface can be separated into two basic areas: The configuration area (1) at the top of the view and the TreeView (2) below. Inside the Tree View (2) there is a legend of the used highlights (a), the possible problems interface (b), and color-scale of the node elements (c). Below that the zoomable and draggable tree of the selected stories are shown.

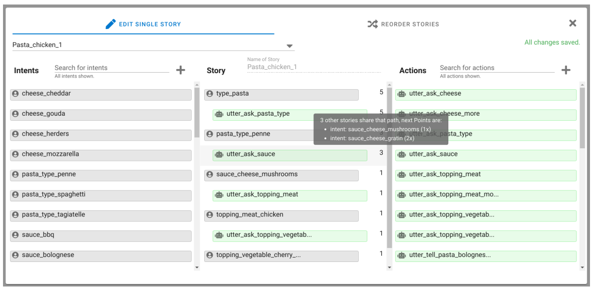
rect intent is provided by the user. The obvious solution is to remove this step from the story.

Filtering: To support the expert in the search for a particular element, *Chatbot Explorer* implements mechanisms to filter the stories by any intent, action, and story. While filtering for a story results in a single story, it is possible to invert every single filter and use that to remove specific stories from the analysis. Additionally, we implemented a user-defined highlight functionality to identify parts of interests faster and find relations between elements. One possible question regarding this is to identify elements where the bot asks questions about the sauce (see orange highlighted action "utter_ask_sauce" in Figure 1). While highlighting intents and actions shows the usage of those elements inside the tree, the highlighting of stories helps to find these stories in the tree more easily.

5.3 Manage Stories

In addition to the tree representation of the knowledge base, other visual representations were developed that experts can use to modify individual elements in the stories (see Task T1).

To create a new story the expert selects a starting point for the new content in the tree. For the new story, all parent nodes of the selected node are inherited to a new story, except the expert selects the root node. Since all stories consist of intents and action, *Chatbot Explorer*s shows all elements as a list, so the expert can add and remove them for the new story. To avoid ambiguities or missing aspects, the systems shows for each item in how many other stories this item is already used. For example, when modeling food orders, the expert can set up all processes as similar as possible for the customer. Since this task requires more than just rearranging existing elements, the expert can also create new intents and actions. Besides the creation of new stories, it is often necessary to modify existing stories. Therefore, the expert can select a node within the TreeView and activate the edit view. In this editing view (see Figure 2) the expert can select a story that uses this element. The selected story is centered in the view as a list. Additionally, a list of all available intents is shown on the left side of the view, and on the other side, a list of all available actions is shown. Equivalent to the creation of a new story, the story can be modified by dragging elements into the list, change the position of elements, or



Pasta_chicken_1

All changes saved.

Pasta_chicken_1

Intents: cheese_cheddar, cheese_gouda, cheese_herders, cheese_mozzarella, pasta_type_penne, pasta_type_spaghetti, pasta_type_tagiatelle, sauce_bbq, sauce_bolognese

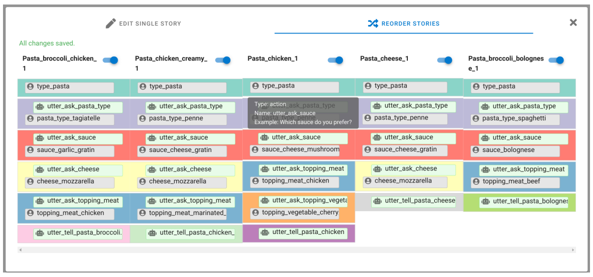
Story:

type_pasta	5
utter_ask_pasta_type	5
pasta_type_penne	3
utter_ask_sauce	3
sauce_cheese_mushrooms	1
utter_ask_topping_meat	1
topping_meat_chicken	1
utter_ask_topping_vegetab...	1
topping_vegetable_cherry_...	1

3 other stories share that path, next Points are:
- intent: sauce_cheese_mushrooms (1x)
- intent: sauce_cheese_gratin (2x)

Actions: utter_ask_cheese, utter_ask_cheese_more, utter_ask_pasta_type, utter_ask_sauce, utter_ask_topping_meat, utter_ask_topping_meat_mo..., utter_ask_topping_vegetab..., utter_ask_topping_vegetab..., utter_tell_pasta_bolognes...

Figure 2: The edit stories interface of the *Chatbot Explorer* that is separated into three parts: The left part shows a list of all intents as well as a full-text search in the list and a button to add a new intent (plus-icon). The same structure is used on the right side for the actions. Between these two lists, the selected story is shown.



All changes saved.

Pasta_broccoli_chicken_1	Pasta_chicken_creamy_1	Pasta_chicken_1	Pasta_cheese_1	Pasta_broccoli_bolognese_1
type_pasta	type_pasta	type_pasta	type_pasta	type_pasta
utter_ask_pasta_type pasta_type_tagiatelle	utter_ask_pasta_type pasta_type_penne	utter_ask_pasta_type pasta_type_penne	utter_ask_pasta_type pasta_type_penne	utter_ask_pasta_type pasta_type_spaghetti
utter_ask_sauce sauce_garlic_gratin	utter_ask_sauce sauce_cheese_gratin	utter_ask_sauce sauce_cheese_mushroom	utter_ask_sauce sauce_cheese_gratin	utter_ask_sauce sauce_bolognese
utter_ask_cheese cheese_mozzarella	utter_ask_cheese cheese_mozzarella	utter_ask_topping_meat topping_meat_chicken	utter_ask_cheese cheese_mozzarella	utter_ask_topping_meat topping_meat_beef
utter_ask_topping_meat topping_meat_chicken	utter_ask_topping_meat topping_meat_marinated_	utter_ask_topping_vegeta topping_vegetable_cherry	utter_tell_pasta_cheese	utter_tell_pasta_bolognes
utter_tell_pasta_broccoli.	utter_tell_pasta_chicken_	utter_tell_pasta_chicken		

Type: action
Name: utter_ask_sauce
Example: Which sauce do you prefer?

Figure 3: The reorder stories interface of the *Chatbot Explorer*. Each story can be disabled so that it does not get reordered along with the other stories. The elements were grouped by the type of question and the corresponding answer.

removing elements from the story. For some cases, the change of the knowledge base could require the deletion of a complete story. One example could be the change of the menu and a pizza can not be ordered anymore. For that purpose *Chatbot Explorer* allows the expert to select an element (see Figure 1 (2)) and use the delete button (trashbin-icon).

A further important aspect identified by the requirements process was the possibility of rearranging existing stories to make them more efficient according to certain criteria (see Task T3). For example, as a retailer, you want to be able to offer customers a short ordering process. For this purpose, *Chatbot Explorer* can identify nodes that are used by many stories. The expert can then activate the Reordering View (see Figure 3), where intents and actions of the stories are grouped. This is especially helpful if stories are very similar, for example in order processes. Therefore, we combine actions and the corresponding intents into groups within a story. Then these groups are compared between all stories using the action name and identical groups in the individual stories are colored with the same background color. The expert can then move blocks in one story and *Chatbot Explorer* automatically adjusts the position of identical blocks in all other stories.

5.4 System Architecture

To ensure that *Chatbot Explorer* is platform independent, we implemented a backend that provides a restful API to serve and import the knowledge base as well as changing it. The backend is written in JavaScript using NodeJs with an express-framework based API. The frontend is written in JavaScript using the VueJs framework for the interface. Some parts, like the color scales, were used from the d3 framework. For performance reasons, the visualizations used are a self-developed, CSS-styled, div-container structure. Due to its modular structure, *Chatbot Explorer* can also be easily integrated into existing systems. For this purpose, it needs access to the chatbot's knowledge base.

6 EVALUATION

While developing *Chatbot Explorer* we conducted two evaluations. Based on the interviews with domain experts, we developed the first prototype in close collaboration with them. This prototype was then tested by a user evaluation. In the first evaluation, several improvements were suggested, so we have extensively revised our visualization. After the implementation of the improvements was completed we conducted a re-evaluation of *Chatbot Explorer*. Both evaluations were using the questions of the SUS (**S**ystem **U**sability **S**cale).

The SUS was introduced by Brooke [3] in the year 1986 and has become since then a standard for evaluating the usability of a software system. It uses a 5-point Likert-type scale for 10 questions to calculate an overall usability score that is presented by a single value between 0 and 100. Following the recommendations of [13], we used the scale defined by Sauro and Lewis. To give the reader a better understanding of the calculated scores we added the grade-based interpretation of Sauro and Lewis [22] that maps the score values to the grades A to F. Additionally, we added a color-based interpretation by McLellan et al. [15] that uses green for excellent results (score between 85 and 100), yellow represents acceptable results (score between 65 and 84) and red represents all not acceptable results (below 65).

Since *Chatbot Explorer* was designed to help maintainers with limited knowledge about the usage of chatbots, there were no special requirements for the participants of the evaluations. Due to the COVID 19 pandemic, the participants had to perform the evaluation on their own hardware, but this also allowed us to test the accessibility as a result of the multiple hardware configurations. The devices used by the participants vary from laptops to workstations, FullHD to 4K displays, different amounts of available screens, and different systems (Linux, Mac, and Windows users). The used browsers were restricted to Chrome and Firefox. For the evaluation, we modified our test dataset to include various errors and problems that the participants were asked to identify and correct.

At the beginning of the evaluation, participants were asked to determine the number of intents that are necessary to order certain dishes. After that, the participants were asked to search for errors such as undecidable paths and unnecessary inquiries in the knowledge base of the chatbot. At the end of the evaluation, participants should rearrange the paths for certain stories in the chatbot to optimize the ordering process. Further descriptions of the use case can be found in the supplemental material. Each evaluation started with a prepared presentation to introduce *Chatbot Explorer* by the tester. For the second evaluation, we added for participants that did not participate in the first evaluation, two more slides for a more detailed view of the used dataset and how it was created. After the presentation, we provided a handout (see supplemental material) with three parts. The first part contains some support images to identify elements and problems. Afterward, multiple-choice questions were asked, which the participants had to answer. The last part contains more complex tasks, like changing the order of some bot queries. Additionally, all participants were asked to share their thoughts on the system while solving the tasks. After finishing the tasks, the tester had an open discussion with each participant about the use of *Chatbot Explorer*.

The form of the first evaluation consisted of the standard SUS questions and the following four additional questions:

1. The gender of the interviewed/participating person (male, female, other)
2. The age of the interviewed/participating person (<18, 18-29, 30-39, 40-49, 50-59, 60-69, >70)
3. The experience of the interviewed/participating person with chatbot-systems in general(5-point Likert-scale: 1 [no experience] ... 5 [expert])
4. If the interviewed/participating person has already worked with the tested system (yes, no, "I have seen pictures or a presentation of it")

For the second evaluation, we adjusted the questions for the experience level based on the findings of the first round (answer options: no experience, some experience, expert) and also ask whether the person participated in the first evaluation.

The first evaluation was attended by 14 persons (12 male and two female). Five persons had an academic natural language processing background, while seven persons worked at the visualization department. The other 2 participants were domain experts of our cooperation partner. In the second evaluation, all 14 participants from the first round attended again as well as six additional participants. Three of the 20 people were female and 17 were male. The background of the participants were: 5 people with a natural language processing background, 11 people with a visualization background, and 4 people from the associated chatbot company. The authors discussed the fact that the second evaluation could be biased by testing the same participants again. After collecting pro and cons we decided, that the reuse of those participants do not have a high impact since the visualizations and interactions comprehensively changed since the first evaluation.

The first evaluation showed some cases where our proposed visualizations did not perform satisfactorily. Also, some improvements were suggested by the participants. For example, many participants had problems with the proposed reordering mechanism due to a missing drag and drop mechanism. Another problem was that the test persons had difficulties in determining whether a branch of the tree represents a decision of the user. The participants also complained that they could only modify selected aspects of the knowledge base. Nevertheless, most participants appreciated the concept of the system and the possibilities of interaction. The first evaluation also shows that the person with more knowledge about chatbots sets a higher standard for the tool than people without experience. The granularity of the 5-point Likert-scale (values from 1 to 5) for the experience question results in one person for each of the values 2 and 4. To provide a better understanding of how the experience influences the results, the authors decided to reduce the experience scale of the first evaluation and aggregate them into three groups: no experience (value 1), some experience (value 2-4), and extensive experience (expert, value 5). The previous scale was mapped, aggregating values of 2 to 4 into the "some experience" group.

The mean of the SUS scores for all participants was a score of 77.9 (grade B+) and a median of 81.3 (grade A). The values according to the experience of the participants are shown in Table 1. Both values are acceptable but can be improved. Based on the results we decided that an additional round of implementing and improving the interaction mechanisms of the system would help to find a better solution.

In the second evaluation, participants who also participated in the first evaluation were more likely to indicate that they already had experience with chatbots in general. The values according to the experience level and if the participants were part of both evaluations can be found in Table 1. The mean of all scores in the second evaluation is 80.88 (grade A) and the median is 85 (grade A+). If one considers only the persons who already participated, the mean value is 79.82 (grade A-) and the median is 85 (grade A+). Those who participated for the second time also noted a significant improvement. All participants of the second evaluation recognized the usability of the *Chatbot Explorer* and expressed the opinion that the use of this system facilitates the maintenance of chatbot knowledge bases. The scores of each participant and

	First Evaluation				Second Evaluation					
Type	All	Experience			All	Participating		Experience		
		N	S	E		Y	N	N	S	E
Number of testers	14	7	6	1	20	14	6	9	7	2
Median	81.3/A	77.5/B+	85.0/A+	67.5/C	85.0/A+	85.0/A+	83.8/A	87.5/A+	90.0/A+	75.0/B
Mean	77.9/B+	75.4/B	82.5/A	67.5/C	80.9/A	79.8/A-	83.3/A	84.7/A+	80.0/A-	73.3/B-

Table 1: Results of the first and second evaluations. The results were obtained by using the following groups as columns: All (All participants that have done the evaluation), separated by the experience level and if they participated in the first evaluation. The columns of the "Experience" groups are: N=no experience, S=some experience, E=extensive experience. The columns of the "Participating" groups are: Y=participated in both evaluations, N=participated in second evaluation only. For each column, the amount of people in this group, as well as the median and the mean of the score-values of the participants in the group, is given.

for each question as well as the computed scores can be found in the supplemental.

7 DESIGN CHALLENGES

Several challenges were faced during the development of the final application that is described in this manuscript. One of the biggest challenges was to develop a visualization that could provide all the necessary information, but not overwhelm the user. This was especially important since Chatbot Explorer is explicitly also aimed at users who have no to little experience in creating and maintaining chatbots. Therefore, the first prototype focused on the visualization and analysis of the structure of the knowledge base. However, users had to identify common errors in knowledge based on the visualization. This turned out to be difficult and time-consuming, therefore we developed automatic error detection. The detected errors are presented in Chatbot Explorer and the user can analyze them individually. This has drastically reduced the required training time for new users.

Another challenge was to create an efficient and easy way of sorting many stories at once. The first version contains a list of stories to manually resort to each story. Based on the feedback from users, we added a function for editing more than one story in parallel is a useful addition as well as the user-friendly drag and drop possibility. The steps to the final described version of the "reorder stories" view contain many tested ideas like the traceability of changes and different types of grouping to assist the user.

8 CONCLUSION AND FUTURE WORK

Our work shows that visualization methods can help to gain a deeper understanding of the knowledge base of a chatbot. Chatbot Explorer can help the maintainer of a chatbot to identify different stories and to recognize parts of stories that s/he is interested in. The proposed tree visualization has the advantage of efficiently representing stories that share elements and the layer visualization helps to get a faster overview of the user's decisions. Additionally, the methods to rearrange the bot's queries and explore the resulting stories have proven to be useful to the experts.

We evaluated the system twice and got overall positive feedback. The participants and our domain experts emphasize the comprehensible visualization as well as the possibility to easily change all elements of a chatbot knowledge base using Chatbot Explorer.

In this work, we observed that changing the order of questions of the bot changes the appearance of several elements and the length of the paths. Therefore, it might be useful to develop a measurement of how certain changes affect corresponding stories and their efficiency to generate suggestions for possible changes and to be able to visualize the consequences of structural changes.

Acknowledgments

This research was funded in parts by the Development Bank of Saxony (SAB) under project numbers 100335729 and 100395769.

9 REFERENCES

[1] S. A. Abdul-Kader and J. Woods. Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7), 2015.

[2] J. Baumeister and M. Freiberg. Knowledge visualization for evaluation tasks. *Knowledge and Information Systems*, 29(2):349–378, Nov 2011.

[3] J. Brooke. System usability scale (SUS): a quick-and-dirty method of system evaluation user information. *Reading, UK: Digital Equipment Co Ltd*, 43, 1986.

[4] C. Buchheim, M. Jünger, and S. Leipert. Improving walker's algorithm to run in linear time. In M. T. Goodrich and S. G. Kobourov, eds., *Graph Drawing*, pp. 344–353. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[5] A. P. Chaves and M. A. Gerosa. How Should My Chatbot Interact? A Survey on Social Characteristics in Human–Chatbot Interaction Design.

International Journal of Human–Computer Interaction, 0(0):1–30, 2020.

[6] S. I. Ch'ng, L. S. Yeong, and X. Ang. Preliminary Findings of using Chat-bots as a Course FAQ Tool. In *2019 IEEE Conference on e-Learning, e-Management e- Services (IC3e)*, pp. 1–5, 2019.

[7] R. Csaky. Deep Learning Based Chatbot Models. National Scientific Students' Associations Conference, 2019. https://tdk.bme.hu/VIK/DownloadPaper/asdad.

[8] M. Glinz. A risk-based, value-oriented approach to quality requirements. *IEEE Software*, 25(2):34–41, March 2008.

[9] F. Johannsen, S. Leist, D. Konadl, and M. Basche. Comparison of Commercial Chatbot solutions for Supporting Customer Interaction. In *ECIS 2018 Proceedings at AIS Electronic Library (AISeL)*, 2018.

[10] D. H. Jonassen. *Tools for Representing Problems and the Knowledge Required to Solve Them*, pp. 82–94. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[11] S. Jyothi, C. McAvinia, and J. Keating. A visualisation tool to aid exploration of students' interactions in asynchronous online communication. *Computers & Education*, 58(1):30–42, 2012.

[12] L. C. Klopfenstein, S. Delpriori, S. Malatini, and A. Bogliolo. The Rise of Bots: A Survey of Conversational Interfaces, Patterns, and Paradigms. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, DIS '17, p. 555–565. Association for Computing Machinery, New York, NY, USA, 2017.

[13] J. R. Lewis and J. Sauro. Item Benchmarks for the System Usability Scale. *J. Usability Studies*, 13(3):158–167, May 2018.

[14] W. Maroengsit, T. Piyakulpinyo, K. Phonyiam, S. Pongnumkul, P. Chaovalit, and T. Theeramunkong. A Survey on Evaluation Methods for Chatbots. In *Proceedings of the 2019 7th International Conference on Information and Education Technology*, ICIET 2019, p. 111–119. Association for Computing Machinery, New York, NY, USA, 2019.

[15] S. McLellan, A. Muddimer, and S. C. Peres. The Effect of Experience on System Usability Scale Ratings. *J. Usability Studies*, 7(2):56–67, Feb. 2012.

[16] A. Neumann, W. Gräber, and S.-O. Tergan. *ParIS – Visualizing Ideas and Information in a Resource-Based Learning Scenario*, pp. 256–281. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[17] P. S. Newman. Exploring Discussion Lists: Steps and Directions. In *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '02, p. 126–134. Association for Computing Machinery, New York, NY, USA, 2002.

[18] V. Pascual-Cid and A. Kaltenbrunner. Exploring Asynchronous Online Discussions through Hierarchical Visualisation. In *2009 13th International Conference Information Visualisation*, pp. 191–196, 2009. doi: 10.1109/IV.2009.14

[19] A. M. Rahman, A. A. Mamun, and A. Islam. Programming challenges of chatbot: Current and future prospective. In *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pp. 75–78, 2017. doi: 10.1109/R10-HTC.2017.8288910

[20] E. M. Reingold and J. S. Tilford. Tidier drawings of trees. *IEEE Transactions on Software Engineering*, SE-7(2):223–228, 1981.

[21] K. Renaud and J. van Biljon. A Framework to Maximise the Communicative Power of Knowledge Visualisations. In *Proceedings of the South African Institute of Computer Scientists and Information Technologists 2019*, SAICSIT '19. Association for Computing Machinery, New York, NY, USA, 2019.

[22] J. Sauro and J. R. Lewis. When Designing Usability Questionnaires, Does It Hurt to Be Positive? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, p. 2215–2224. Association for Computing Machinery, New York, NY, USA, 2011.

[23] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pp. 336–343, 1996.

[24] A. M. Turing. Computing machinery and intelligence. *Mind*, LIX(236):433–460, 10 1950.

[25] M. Wattenberg and D. Millen. Conversation Thumbnails for Large-Scale Discussions. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, p. 742–743. Association for Computing Machinery, New York, NY, USA, 2003.

[26] J. Weizenbaum. ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine. *Commun. ACM*, 9(1):36–45, Jan. 1966.

[27] A. Yaeli and S. Zeltyn. Where and why is my bot failing? a visual analytics approach for investigating failures in chatbot conversation flows. In *2021 IEEE Visualization Conference (VIS)*, pp. 141–145, 2021. doi: 10.1109/VIS49827.2021.9623295

Semi-automatic Acquisition of Datasets for Retail Recognition

Marco Filax, Tim Gonschorek, Frank Ortmeier
Otto von Guericke University Magdeburg
Universitätsplatz 2
Germany, 39106, Magdeburg, Sachsen-Anhalt
firstname.lastname@ovgu.de

ABSTRACT

The acquisition of datasets is typically a laborious task. It is challenging, especially if the required annotations in every image in the dataset are vast. It is even more challenging if the inter-class variance, the visual difference between two distinct classes, is low. Retail product recognition constitutes an example of both issues. Products are densely packed on shelves, resulting in many objects within an image. Products share visual similarities, which makes them hard to distinguish.

In this work, we propose Annotron, a tool tackling the acquisition problem in this domain. Exploiting dataset structures, such as being organized in consecutive frames, we detect real-world objects through pre-trained detectors and reproject detections to generate candidate traces over time. Further, we aid labelers by computing potential matches of real-world objects and reference images based on their visual similarity: We cluster consecutive detections based on a large set of reference images using embeddings acquired from pre-trained networks.

Using the proposed tool reduces manual efforts drastically by diminishing the time spent on repetitive, error-prone tasks. We evaluate Annotron in the retail recognition domain. The domain is commonly considered fine-grained, which means that instance-level annotations are costly due to the described problems. We refine the given dataset, surpass the number of previously found stock-keeping units, and label over 446.500 individual bounding boxes.

Keywords
Dataset Acquisition, Pattern Recognition, Smart Assistance, Retail Product Recognition

1 INTRODUCTION

Collecting datasets for supervised learning tasks is a laborious but mandatory task. It typically requires human data labelers to judge vast amounts of data points such that a model can learn to make correct decisions. In the context of computer vision, we typically let labelers annotate images with bounding boxes and classes, i.e., visual concepts represented by regions of pixels. Further, additional techniques might be required that help to increase the overall accuracy, such as labeler consensus or label auditing. These techniques increase the overall time investment to design a particular solution for new supervised learning tasks

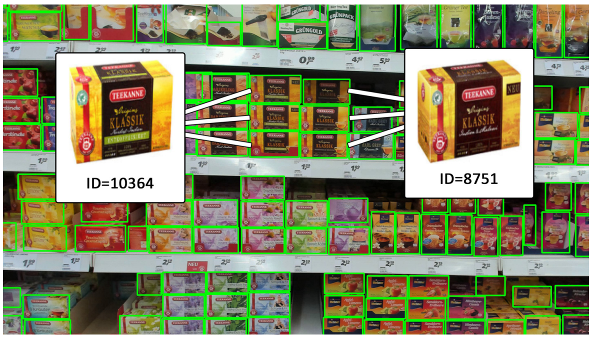


Figure 1: Retail product recognition is a supervised fine-grained visual learning task. Crowded scenes and low visual inter-class variance require significant manual annotation efforts. We propose *Annotron*, a tool that lowers these hurdles in fine-grained domains.

The acquisition of datasets is extremely costly in the fine-grained domain. It requires solid precision to determine the correct visual concept of a particular image region. The problem of retail product recognition gives a perfect example of a fine-grained domain. Figure 1 depicts an exemplary image of this domain. It visualizes two issues that arise in this fine-grained domain: First, images of retail products on shelves comprise crowded scenes. Similar-looking products are densely stacked across the complete frame. Annotating a single frame is a tedious task due to the sheer number of ob-

https://www.doi.org/10.24132/CSRN.3201.11

jects. Second, some products can only be distinguished through small visual cues. Both reference images illustrate this problem: although they are entirely different objects, they share significant visual similarities. If meta-data of the reference set is available: labelers have to annotate the visual concepts based on an exhausting full-text search over the product's name. If meta-data is not available: the situation is even worse. Labelers would have to select the correct match based on visual cues. Semi-automatic approaches could tackle both problems.

[5] surveyed related works in the domain of (semi-)automated image annotation. We focus on more recent works since the authors already gave a broad review of works proposed until 2017. More recently, works have been proposed that use the labeler in a loop in conjunction with trained networks [1], [2], [14], [16], [26]. Here, the complete dataset is typically split into two sets. The first is manually annotated and used to train networks which then predict labels for the latter. Other works extend this approach by fuzzing the dataset split, i.e., by predicting the best video frames to annotate manually [15]. [17] proposed to use properties of existing detectors, which tend to predict multiple different-sized bounding boxes for a single object. The user has then to choose the correct one iteratively. [21] proposed to use GrabCut [19], which is designed to segment an object based on user input. Using additional meta-knowledge, i.e., the depth stream of a video stream, [21] proposed to track the segmented object in 3D space and reproject bounding boxes onto the image planes. [7] used meta-knowledge of the environment to propose a semi-automatic annotation tool tailored for retail environments. The authors used a SLAM approach to annotate objects in 3D space and reproject their annotation onto consecutive video frames. Related works focus on general (semi-)automated image annotation to our knowledge. Our approach does not need any additional knowledge about the environment and is tailored to crowded scenes, in which mainly the manual annotation of individual objects is time-consuming and error-prone.

In this work, we propose a semi-automatic image annotation system called *Annotron*, with the scope of allowing a fast and continuous annotation workflow. We evaluate the proposed approach in the fine-grained domain of retail recognition, in which acquiring instance-level annotations is considered costly. We propose exploiting underlying structures of datasets by reprojecting found bounding boxes to consecutive frames. This approach yields a candidate stream of a particular object over time and extracts multiple views of the same object. We extract embeddings, a lower-dimensional representation of the visual content, of every image patch in the candidate stream. With these, we form groups of similar-looking visual concepts, e.g., retail products that share similar looks or real-world images that look similar to a particular reference image. We gather the set of nearest neighbors for every candidate stream. A labeler finally identifies the correct reference image to acquire the ground-truth annotation.

The proposed approach efficiently lowers the need for manual assistance. It enables labelers to annotate different views of the same stock-keeping unit simultaneously. Further, it reduces the search space dramatically from which the labeler has to choose the correct reference concept. We achieve this by using a mutual embedding space of candidates and reference images. We demonstrate the ease of use throughout the paper. Our experiment shows that the proposed approach combined with lower annotation efforts **extends** a given database. We found more visual concepts using fewer annotation efforts as proposed in the original work.

The paper is structured as follows: Section 2 explains the proposed approach in detail. We describe the required preprocessing step in-depth and elaborate on the resulting tool support that differentiates manual annotation tasks. Afterward, we evaluate the proposed approach with an already given dataset in Section 3. We report on the specific choices that arise in the preprocessing step and demonstrate that the resulting new dataset extends the previously given dataset. Finally, we conclude our work.

2 ANNOTRON

We propose a semi-automatic labeling tool to ease the hurdle of acquiring few-shot datasets focused on fine-grained recognition problems. We follow a two-stage approach: First, we preprocess the original data to identify similar-looking regions across multiple frames as described in Section 2.1. This is done in a fully automated manner. Second, we label these region tracks manually, which we call candidate traces, using different labeling strategies. This allows us to annotate vast amounts of previously tracked candidate traces efficiently. Further, we automatically cluster similar-looking traces to reduce the manual search time invested. We describe the manual tasks in Section 2.2.

2.1 Preprocessing

The first stage of the proposed approach is a fully-automated preprocessing process that aims at generating an intermediate data structure. The core idea is to automatically extract high-level representations of subsequent image regions and determine similar visual concepts. We group and present these in an ordered manner to the labeler (cf. Section 2.2).

The basis for the proposed preprocessing step is the assumption that a dataset is organized in a consecutive manner, i.e., in video streams. Further, we assume

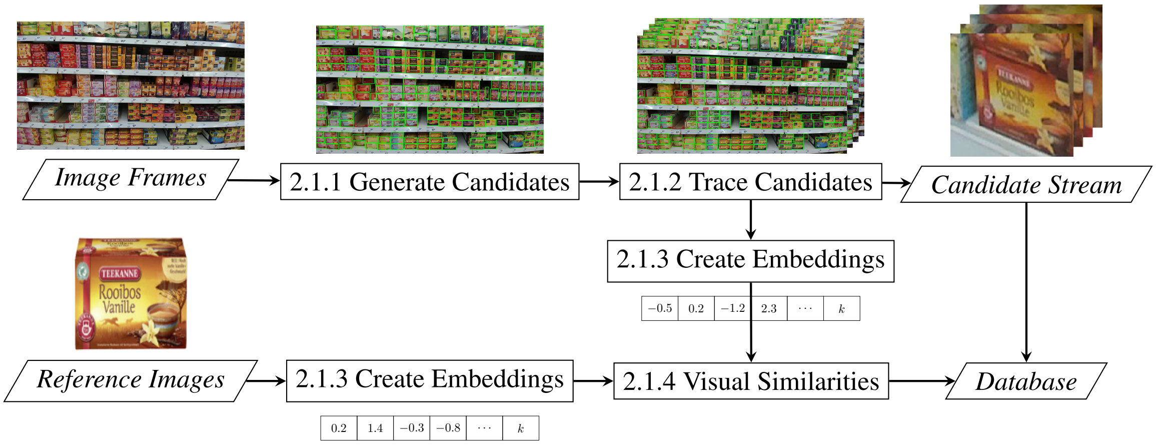


Figure 2: Flowchart of the proposed preprocessing module. We trace candidates across consecutive frames and use embeddings to find similar-looking image patches. We comprise these findings in a database for fast user access.

that at least a single reference image of all relevant visual concepts is available. Figure 2 depicts the proposed preprocessing module. Generally, we propose reprojecting given candidates, e.g., acquired with a generic detection module, onto consecutive frames (cf. Section 2.1.1). Using the resulting candidate streams (cf. Section 2.1.2), we encode these and the reference images using an embedding model (cf. Section 2.1.3). Finally, we extract the nearest neighbors (cf. Section 2.1.4) of every candidate stream and comprise the results in a special data structure.

2.1.1 Generate Candidates

The first step in the proposed preprocessing module is generating possible object candidates on every real-world image. Generally, we assume that an image holds multiple objects of interest. We propose acquiring possible candidates, i.e., arbitrary shapes or volumes in the input space, using a general-purpose detector, if possible. The detector does not need to predict the correct class or concept of a particular object of interest.

In this work, we focus on images as input. We consider a bounding box the most common form of shape that needs to be detected for every frame. Consequently, we use the detector to predict bounding boxes for as many objects as possible. We emphasize the possibility of tailoring the proposed preprocessing module as needed, i.e., replacing the detector with a shape predictor. This might be necessary if the objects of interest are to be described in a different form.

2.1.2 Trace Candidates

We aim at lowering the amount of manual labeling activities. Therefore, the core idea is to trace object candidates - regions within every frame that depict similar visual concepts - across multiple frames. We use the well-known overlap measure intersection over union. Given two shapes $A, A' \in R^2$, the IoU is defined as

$$IoU = \frac{|A \cap A'|}{|A \cup A'|} \tag{1}$$

We iteratively trace two shapes, i.e., bounding boxes, A and A' of two consecutive frames. We select the overlapping bounding boxes with a watershed algorithm while maximizing the IoU. Thereby, we generate candidate streams - traces of real-world objects over time - by iterating over all images of a video to gather different views of the same object. We empirically found that this approach seems reasonable precise for video streams from the retail domain. However, we maintain the possibility to slice a candidate stream manually through the labeler.

2.1.3 Create Embeddings

Next, we aim at clustering found candidate streams to find visual cues of similar objects shown in the image regions. To do so, we create a vectorized representation of the visual content of every image in the candidate stream. We use deep neuronal nets to create embeddings, a lower-dimensional representation of the visual content. This is a common approach for many specialized fields, such as face recognition [6], [20], [27] or person recognition [3], [13], [22], that typically deal with few-shot learning problems.

[4], [24], [25] have shown that generic classification networks can be used for a few shot recognition tasks, with and without any additional training. The authors proposed removing the network's classification head and using the penultimate output as embeddings. For the domain of retail recognition, there also exist specialized neuronal networks [8], [9], [23] designed to produce tailored embeddings. We use a specialized network wherever pre-trained weights tailored to the domain are available and use generic embedders otherwise.

2.1.4 Visual Similarities

Generally, the proposed approach uses the fact that embeddings of similar concepts are mapped to close regions in the embedding space. We exploit the observation by mapping real-world traces and reference images into a mutual embedding space. We then select the nearest reference images for every sample within a candidate stream. Further, we gather the nearest candidate streams of each candidate stream. Here we use the center point of the embeddings, i.e., the mean embedding of the complete candidate stream, because we assume that a candidate stream depicts the same visual concept over time.

We gather the top set of nearest neighbors in both cases, assuming that in the top nearest neighbors most likely is the true visual concept. Given the underlying problem of retail recognition with its fine-grained nature, we found that retrieving the five nearest reference images whereas 50 nearest candidate streams of every candidate stream yield feasible results. We gather all nearest neighbors in a data structure, i.e., using identifies, for fast and easy access during the labeling procedure.

2.2 Labeling Procedure

The second step of the proposed process is the actual labeling, i.e., linking real-world objects with visual concepts. Labeling vast amounts of data is a monotone, time-consuming, and error-prone task. To overcome some of the hurdles induced through the necessity of training with labeled data, we proposed a preprocessing step in the previous Section 2.1. The core idea is to use the previously computed similar-looking candidate streams with their most similar-looking reference images and present this information in an ordered manner to the user. We thereby focus on annotating complete candidate streams with minimal user interaction.

The ability to operate on candidate streams rather than on individual image regions yields different benefits during the labeling process. First, it produces a particular relevance scheme for all candidate streams, which induces an order. Second, it enables us to cluster candidate streams through computing similarity measures. This allows us to identify similar visual concepts from different directions, i.e., previously labeled candidate streams that share joint visual embeddings are more likely to depict the same visual concept.

Ordering the labeling procedure yields soft benefits during processing, such as faster feedback loops that increase morale. We inherit a better ability to monitor the labeling activities because the number of manual interactions is greatly reduced. It is becoming easier to reach achievements.

Using clustered candidate streams further allows us to distinguish two different types of labeling tasks: *Identifying new visual concepts* in the data that have not been linked to visual concepts and *increasing the number of observations* of already seen concepts. In the following, we present both tasks in detail and elaborate on the design choices for each task.

2.2.1 Task 1: Identify Visual Concepts

When we identify new concepts in the data, we link previously unseen reference concepts to candidate streams. Without *Annotron*, we would have to detect new concepts based on the human eye and a full-text search over the reference names if the labeler can identify them. This is error-prone due to the fine-grained nature of the problem. Using *Annotron* provides better tool support to the user while identifying new concepts by presenting similar-looking reference concepts. Presenting visually similar reference and candidate images removes the burden of full-text search activities and, therefore, increases the overall labeling speed of a single labeler.

We propose specialized tool support for this particular task to rank the previously computed candidate streams according to two different metrics, which are defined as follows:

Tracking Stability: We weigh the candidate streams with the longest stable tracking to be the most relevant. Given a candidate stream $c_i \in C$, we define the metric m_t as $m_t(c_i) = -|c_i|$. This metric focuses on fast annotations that quickly maximize the total number of observed examples per concept in the dataset.

Embedding Stability: We sort the candidate streams according to their visual stability. We assume that a candidate stream depicting a particular concept available in the set of reference images will lead to stable nearest neighbors in the embedding space of reference images. Given a candidate stream c_i, with multiple nearest reference concepts N_{c_i} in the embedding space, we define the metric m_e as $m_e(c_i) = \frac{|\{N_{c_i}\}|}{|c_i|}$.

2.2.2 Task 2: Increase Observations of Concepts

The second annotation task increases the total number of annotations of a particular concept within the dataset. The user manually identifies the previously found reference concepts on other video stream regions in the dataset. Without *Annotron*, we would again have to identify a reference concept of a particular candidate stream based on the human eye and full-text search. This task would not differ from the previous and might induce errors in the annotation process.

Using *Annotron* allows us to present similar-looking reference images when viewing a candidate stream. This dramatically reduces the number of full-text queries a labeler has to issue during the annotation procedure. It also inherits a reduced risk of annotation errors. Further, when the labeler selects a reference concept, that is already assigned to a stream, we display similar-looking candidate streams. This decreases the effort of identifying similar-looking ones in other streams. We distinguish two different types of similarity relations:

Similarity to other Candidate Streams: To describe the visual distance of two candidate streams, we use the center of the hyperspheres in the embedding spaces formed with all embeddings in a stream. To do so, we use the mean embeddings μ of a candidates stream $c_i \in C$ which are defined, following [11], as $\mu_i = \frac{1}{|c_i|}\sum_j^{|c_i|} z_j$, whereas z_j is the embedding of an image in the candidate stream. We measure the distance of candidate streams as $distance(c_i, c_j) = ||\mu_i - \mu_j||_2^2$.

Similarity to Reference Concepts: Further, we propose to measure the distance of a candidate stream and reference images. We measure the distance of a candidate stream and a reference embedding z_r similarly, $distance(c_i, z_r) = ||\mu_i - z_r||_2^2$. This allows us to present candidate streams visually similar to reference concepts during the labeling process.

Other works do typically not differentiate these two tasks. We, however, assume that modern visual neuronal nets produce embeddings powerful enough to distinguish the visual content of image patches to some extend. This is justified by observing that these approaches are used in broad scopes, i.e., face, human, or character recognition and anomaly detection. Using visual similarity to assist the user dramatically reduces the mental efforts of an annotator while labeling image patches. We found the ability to distinguish different labeling tasks to be the most dominant benefit. It enables us to differentiate the tool support necessary for every individual step to increase the overall efficiency of the manual input.

We implemented the proposed approach in a tool called *Annotron*. An example is shown in Figure 3. We evaluate the proposed approach using a case study from the retail recognition domain.

3 CASE-STUDY

In this section, we present the implementation of the proposed tool called *Annotron*. We evaluate the proposed tool in the fine-grained domain of retail product recognition. First, we describe an existing dataset that will serve as a basis for our work in Section 3.1. Second, we elaborate on the concrete implementation of the automatic preprocessing module. Third, we give insights on the manual work in Section 3.3. Finally, we evaluate the outcome based on the resulting dataset and compare it to the original dataset.

3.1 A Retail Recognition Dataset

We refine an already existing dataset to evaluate the proposed approach. We chose a fine-grained dataset from the domain of retail product recognition. In our opinion, fine-grained datasets cover a very challenging acquisition problem. Thus, we chose to refine a semi-automatic generated large-scale dataset taken from [7] to illustrate the efficiency of the proposed approach.

The original dataset contains over 23.000 different reference images and offers 41.000 frames from more than 20 video sequences. They were gathered using a webcam mounted to a hololens. The authors used additional meta-knowledge of the environment to label the data semi-automatically. The authors found 871 different stock-keeping units in the database, collected with four labelers.

Unfortunately, the dataset does comprise some inaccuracies. Bounding boxes were tracked over time using the camera's position calculated with the internal measurements of the hololens. This leads to some irregular annotations, possibly due to synchronization issues. Further, the authors deployed standard tool support to identify visual concepts, such as a full-text search using the products' names.

We assume that there is some headroom for improvements regarding the complete localization of available reference concepts, i.e., products. We find it challenging to determine the fine-grained visual differences of retail products because products typically share significant visual aspects and the enormous reference concept space. This makes the Magdeburg Groceries Dataset [7] the perfect basis for our experiments.

3.2 Automatic Preprocessing

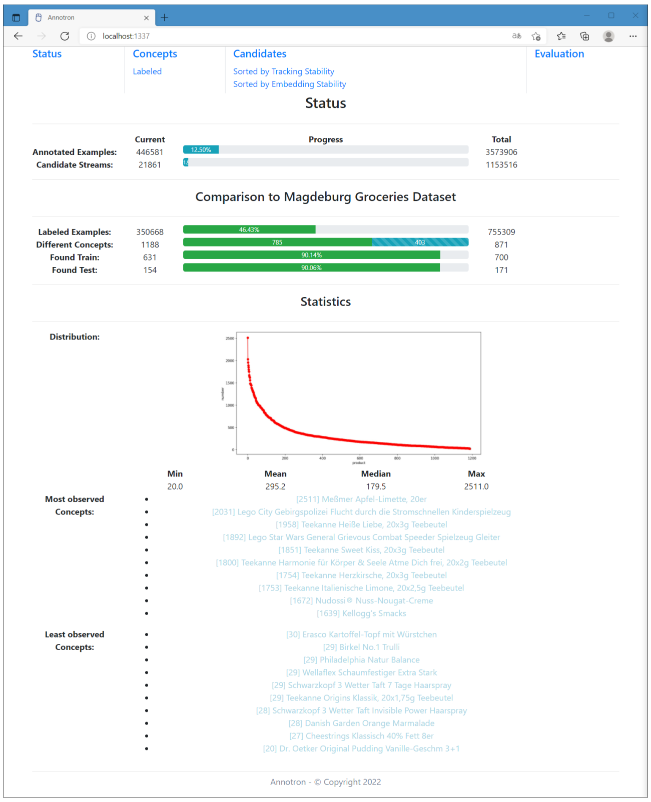
This section describes our implementation of the proposed automatic preprocessing module. Further, we present detailed information on the parametrization.

3.2.1 Generate Candidates

Following the proposed approach, we generated bounding boxes for every frame as proposed in Section 2.1.1. We used a pre-trained network proposed in [18]. It was trained on the SKU-110k dataset [10]. The dataset is taken from a similar domain - retail product *detection*. Thus, it can acquire the location of retail products on shelves. We chose to use the network with pre-trained weights due to the pure availability of a pre-trained detector. However, it cannot recognize the classes of products in any sense. Thus, it serves as a good basis to predict bounding boxes in the retail domain.

3.2.2 Trace Candidates

The previously predicted bounding boxes are mapped from one frame to the next. In this step, the goal is



Status

	Current	Progress	Total
Annotated Examples:	446581	12.50%	3573906
Candidate Streams:	21861	1.9	1153516

Comparison to Magdeburg Groceries Dataset

Labeled Examples:	350668	46.43%	755309
Different Concepts:	1188	785 403	871
Found Train:	631	90.14%	700
Found Test:	154	90.06%	171

Statistics

Distribution:

Min	Mean	Median	Max
20.0	295.2	179.5	2511.0

Most observed Concepts:

- [2511] Meßmer Apfel-Limette, 20er
- [2031] Lego City Gebirgspolizei Flucht durch die Stromschnellen Kinderspielzeug
- [1958] Teekanne Heiße Liebe, 20x3g Teebeutel
- [1892] Lego Star Wars General Grievous Combat Speeder Spielzeug Gleiter
- [1851] Teekanne Sweet Kiss, 20x3g Teebeutel
- [1800] Teekanne Harmonie für Körper & Seele Atme Dich frei, 20x2g Teebeutel
- [1754] Teekanne Herzkirsche, 20x3g Teebeutel
- [1753] Teekanne Italienische Limone, 20x2,5g Teebeutel
- [1672] Nudossi® Nuss-Nougat-Creme
- [1639] Kellogg's Smacks

Least observed Concepts:

- [30] Erasco Kartoffel-Topf mit Würstchen
- [29] Birkel No.1 Trulli
- [29] Philadelphia Natur Balance
- [29] Wellaflex Schaumfestiger Extra Stark
- [29] Schwarzkopf 3 Wetter Taft 7 Tage Haarspray
- [29] Teekanne Origins Klassik, 20x1,75g Teebeutel
- [28] Schwarzkopf 3 Wetter Taft Invisible Power Haarspray
- [28] Danish Garden Orange Marmalade
- [27] Cheestrings Klassisch 40% Fett 8er
- [20] Dr. Oetker Original Pudding Vanille-Geschm 3+1

Annotron - © Copyright 2022

Figure 3: We implemented *Annotron* as a web service for easy access and minimal system requirements. The status page depicts various statistics of the current dataset.

to interconnect the locations across multiple frames. As proposed in Section 2.1.2, we greedily select the bounding boxes by calculating the *IoU* of consecutive frames. Thereby we maximized the overlap based on iteratively selecting bounding boxes that overlap with a watershed algorithm. We accepted consecutive traces up to a threshold of 0.5. The consecutive trace of overlapping bounding boxes, i.e., candidate traces, is fed into the embedding network to describe the visual content.

3.2.3 Create Embeddings

We describe the visual content of candidate traces using embeddings to assist the user during the error-prone manual work. We proposed using an existing network to generate embeddings of visual inputs in Section 2.1.3. We again use an already-trained network [8] from the retail domain. The network architecture is based on the well-known resnet-50 architecture [12]. The classification head was removed, and a 128-dimensional embedding head was attached. We embed the visual content of every reference concept, and every candidate of all traces scaled to 128x128 pixel patches. We use the euclidean distance metric to compare these embeddings.

3.2.4 Visual Similarities

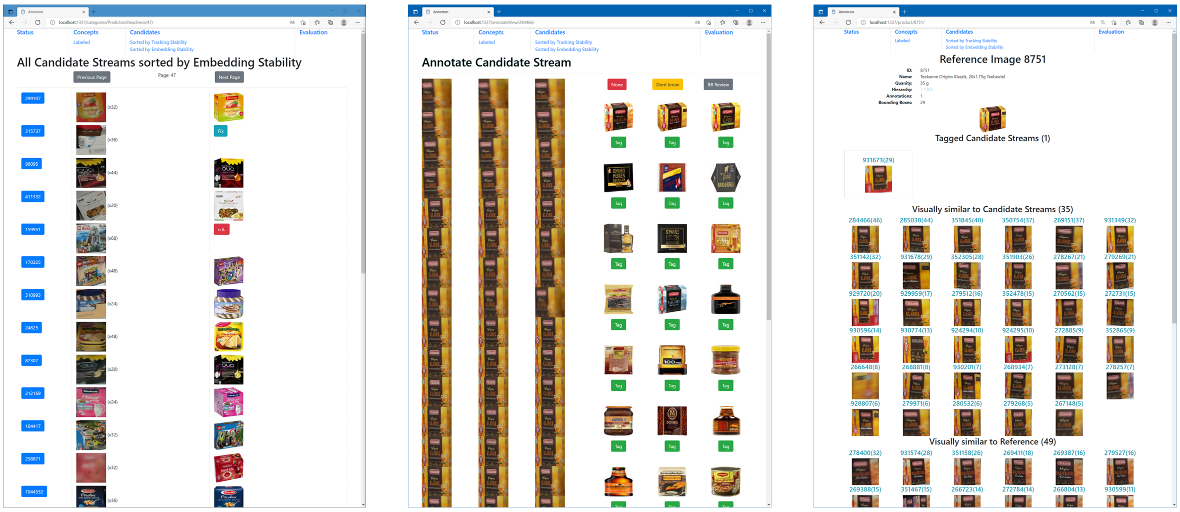
The last preparation step is the identification of visual similarities. As proposed in Section 2.2, these embedded visual representations are used to assist the labeling procedure later. As shown there, we are using the k-nearest neighbors of reference and candidate images as well as the k-nearest neighbors of the mean of candidate images. We use an approximated variant[1] since retrieving k=50 nearest neighbors can be a resource-intensive task. We precompute the nearest neighbors and save their identifiers in a database to allow a smooth user experience.

3.3 Manual Annotation

Finally, it is the task of the labeler to identify visual concepts and annotate them manually. Figure 3 depicts the status page of the proposed tool. It allows the labeler to monitor the labeling progress itself and the distribution of tagged candidate streams.

Further, it displays a comparison to the original Magdeburg Grocery dataset [7]. We identified 1188 visual concepts, i.e., retail products. The original work found 871 visual concepts in the data. We assume that the difference is mainly because the original work used standard tool support to identify products, such as a full-text search over product names. We instead used the visual similarities of image patches. Thus, we conclude that the larger number of found products must be due to the better tool support.

[1] github.com/spotify/annoy


All Candidate Streams sorted by Embedding Stability
Previous Page
Next Page
Annotate Candidate Stream
None
Don't know
Tag
Reference Image 8751
Tagged Candidate Streams (1)
Visually similar to Candidate Streams (35)
Visually similar to Reference (49)
Status
Concepts
Candidates
Evaluation

(a) Identify Visual Concepts. *Annotron* comprises special tool support that structures the annotation workflow. Tracked candidate streams are ordered based on their embedding stability.

(b) Annotation View. *Annotron* depicts all samples of a candidate stream and the nearest neighbors based on visual similarity.

(c) Increase Observations. The user can quickly identify more visually similar candidate streams using already tagged streams or the reference image of the visual concept 8751.

Figure 4: Different stages of the annotation process depicted in *Annotron*. *Annotron* provides specialized tool support for the different stages in the labeling process.

Figure 4 depicts the proposed tool support at a glance. In this work, we describe the three major views of *Annotron*. We cover the particular views of *Annotron* in the following.

3.3.1 Identify Visual Concepts

Figure 4a depicts the user interface of *Annotron* during the identification phase. Here, the candidate streams are ordered by embedding stability. A single click redirects the user to Figure 4b.

On the left side of Figure 4b all individual image patches of the candidate stream are shown. The right side of *Annotron* displays a distinct list of the top-5 nearest neighbors of every image patch in the candidate stream ordered by their number of votes. A single click links the visual reference concept to the candidate stream. *Annotron* redirects the user to the next candidate stream (cf. Figure 4b) without additional interaction.

3.3.2 Increase Observations of Concepts

Figure 4c depicts the user interface of *Annotron* designed explicitly for the phase of increasing the observations of the particular visual concept. The GUI is organized similarly to a classical website. At the top, all information of the visual concept is presented. After that, already annotated candidate streams are depicted. Then, candidate streams that are similar to already tagged streams are depicted. With a single click, the user can link the displayed candidate streams to the visual concept. If the user hovers over a candidate stream, we virtually scroll through it, enabling him to identify invalid patches easily. Finally, the nearest neighbors of the visual concept in the embedding space are depicted at the bottom of the page. In the particular example, we can see the challenges of fine-grained recognition problems: Close neighbors in the candidate stream space depict visually similar classes to already tagged candidate streams, but the nearest neighbors of the reference image depict purely invalid samples that share large visual similarities. It is the duty of the labeler to identify only valid matches and link real-world and reference images accordingly.

3.4 Refined Dataset

This section covers the final result of our labeling activity. We detected 3.573.906 images patches in the complete set of image frames. We further managed to track 1.153.516 candidate streams. We annotated 446.481 image patches of 1.188 different retail products. Thereby, we manually annotated 21.861 candidate streams, meaning that every stream we manually labeled consists of 20 tracked bounding boxes on average. That underlines the greedy strategy we employed through the labeling phase.

As shown in the *Annotron* tool in Figure 3, we managed to identify over 90% of the previously found visual concepts. This is especially remarkable since we annotated the retail products with lower manpower as in the original dataset. We labeled the data with a single labeler, while four labelers were needed in the original work. Further, we identified over 400 new visual concepts that were already presented in that dataset and not labeled in the original variant. We conclude that this increase of found retail products has to originate in the better tool support proposed in this work.

4 CONCLUSION

In this article, we proposed an semi-automatic image annotation tool called *Annotron*, which aims at lowering the hurdle of acquiring new datasets in fine-grained domains. We exploit the structure in datasets, such as in the domain of retail product recognition, in which products are densely packed on shelves, to achieve a fast and continuous annotation workflow by reprojecting bounding boxes of consecutive image frames. This is also helpful in other datasets if they are similarly organized in video frames.

Further, we exploit the capabilities of modern neuronal networks by projecting the visual contents of reference and real-world images into a mutual embedding space. This enables us to extract similar-looking objects and determine possible matches, i.e., depicting the same visual concept, from both input spaces. We implement an intuitive interface that presents possible matches to a labeler to acquire ground truth annotations of objects tracked over time.

We demonstrated the applicability of *Annotron* in the fine-grained domain of retail recognition. We refined an existing database from the literature and extended the total number of found stock-keeping units with lesser manual effort. We showed that the proposed approach efficiently lowered the need for manual assistance during the labeling procedure.

However, we only focused on a single database and heavily relied on the fact that it consists of videos. If the images to be tagged are not consecutively ordered, we cannot track patches across multiple frames, which prevents us from finding candidate samples and ultimately increases the required annotation efforts. The effort gains originated in the proposed idea of using encoded image representations to acquire possible matches remain in effect. We plan to address this validity flaw by extending another database in the future.

REFERENCES

[1] B. Adhikari and H. Huttunen, "Iterative bounding box annotation for object detection," *Proc. - Int. Conf. Pattern Recognit.*, pp. 4040–4046, 2020. arXiv: 2007.00961.

[2] B. Adhikari, J. Peltomäki, J. Puura, and H. Huttunen, "Faster Bounding Box Annotation for Object Detection in Indoor Scenes," *Proc. - Eur. Work. Vis. Inf. Process. EUVIP*, 2019. arXiv: 1807.03142.

[3] X. Bai, M. Yang, T. Huang, Z. Dou, R. Yu, and Y. Xu, "Deep-Person: Learning discriminative deep features for person Re-Identification," *Pattern Recognit.*, vol. 98, 2020. arXiv: 1711.10658.

[4] A. Bendale and T. E. Boult, "Towards Open Set Deep Networks," in *CVPR*, vol. 2016-Decem, IEEE, Jun. 2016, pp. 1563–1572. arXiv: arXiv:1511.06233v1.

[5] Q. Cheng, Q. Zhang, P. Fu, C. Tu, and S. Li, "A survey and analysis on automatic image annotation," *Pattern Recognit.*, vol. 79, no. November, pp. 242–259, 2018.

[6] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," 2018. arXiv: 1801.07698.

[7] M. Filax, T. Gonschorek, and F. Ortmeier, "Data for Image Recognition Tasks: An Efficient Tool for Fine-Grained Annotations," in *ICPRAM*, SciTePress, 2019, pp. 900–907.

[8] M. Filax, T. Gonschorek, and F. Ortmeier, "Grocery Recognition in the Wild: A New Mining Strategy for Metric Learning," in *VISIGRAPP 2021 - Proc. 16th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 4, SciTePress, 2021, pp. 498–505.

[9] M. Filax and F. Ortmeier, "On the influence of viewpoint change for metric learning," in *Proc. MVA 2021 - 17th Int. Conf. Mach. Vis. Appl.*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021.

[10] E. Goldman, R. Herzig, A. Eisenschtat, J. Goldberger, and T. Hassner, "Precise Detection in Densely Packed Scenes," in *2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, IEEE, Jun. 2019, pp. 5222–5231. arXiv: 1904.00853v3.

[11] M. Hassen and P. K. Chan, "Learning a neural-network-based representation for open set recognition," in *Proc. 2020 SIAM Int. Conf. Data Mining, SDM 2020*, Society for Industrial and Applied Mathematics Publications, 2020, pp. 154–162. arXiv: 1802.04365.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, IEEE, 2016, pp. 770–778. arXiv: 1512.03385.

[13] A. Hermans, L. Beyer, and B. Leibe, "In Defense of the Triplet Loss for Person Re-Identification," *arXiv:1703.07737*, 2017. arXiv: 1703.07737.

[14] K. G. Ince, A. Koksal, A. Fazla, and A. A. Alatan, "Semi-Automatic Annotation for Visual Object Tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2021-Octob, 2021, pp. 1233–1239.

[15] A. Kuznetsova, A. Talati, Y. Luo, K. Simmons, and V. Ferrari, “Efficient video annotation with visual interpolation and frame selection guidance,” in *Proc. - 2021 IEEE Winter Conf. Appl. Comput. Vision, WACV 2021*, 2021, pp. 3069–3078. arXiv: `2012.12554`.

[16] T. N. Le, S. Akihiro, S. Ono, and H. Kawasaki, “Toward interactive self-annotation for video object bounding box: Recurrent self-learning and hierarchical annotation based framework,” in *Proc. - 2020 IEEE Winter Conf. Appl. Comput. Vision, WACV 2020*, 2020, pp. 3220–3229.

[17] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari, “We Don’t need no bounding-boxes: Training object class detectors using only human verification,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, no. 1, pp. 854–863, 2016. arXiv: `1602.08405`.

[18] T. Rong, Y. Zhu, H. Cai, and Y. Xiong, “A Solution to Product detection in Densely Packed Scenes,” 2020. arXiv: `2007.11946`.

[19] C. Rother, V. Kolmogorov, and A. Blake, “"GrabCut" - Interactive foreground extraction using iterated graph cuts,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[20] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *CVPR*, IEEE, 2015, pp. 815–823. arXiv: `1503.03832v3`.

[21] D. Stumpf, S. Krauß, G. Reis, O. Wasenmüller, and D. Stricker, “SALT: A semi-automatic labeling tool for RGB-D video sequences,” in *VISIGRAPP 2021 - Proc. 16th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 4, 2021, pp. 595–603. arXiv: `2102.10820`.

[22] X. Sun and L. Zheng, “Dissecting Person Re-Identification From the Viewpoint of Viewpoint,” in *CVPR*, IEEE, 2019, pp. 608–617. arXiv: `1812.02162`.

[23] A. Tonioni and L. Di Stefano, “Domain invariant hierarchical embedding for grocery products recognition,” *Comput. Vis. Image Underst.*, vol. 182, pp. 81–92, 2019. arXiv: `1902.00760`.

[24] R. Vemulapalli and A. Agarwala, “A compact embedding for facial expression similarity,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 5676–5685, 2019. arXiv: `1811.11283`.

[25] N. Vo and J. Hays, “Generalization in metric learning: Should the embedding layer be embedding layer?” *Proc. - 2019 IEEE Winter Conf. Appl. Comput. Vision, WACV 2019*, pp. 589–598, 2019. arXiv: `1803.03310`.

[26] P. Voigtlaender, L. Luo, C. Yuan, Y. Jiang, and B. Leibe, “Reducing the annotation effort for video object segmentation datasets,” in *Proc. - 2021 IEEE Winter Conf. Appl. Comput. Vision, WACV 2021*, 2021, pp. 3059–3068. arXiv: `2011.01142`.

[27] M. Wang and D. Weihong, “Deep Face Recognition: A Survey,” in *Proc. - 31st Conf. Graph. Patterns Images, SIBGRAPI 2018*, 2019, pp. 471–478. arXiv: `1804.06655v8`.

Graphical interface adaption for children to explain astronomy proportions and distances

Kim Martinez
Department of History,
Geography and Communication,
University of Burgos
Pº Comendadores s/n
09001, Burgos, Spain
kmartinez@ubu.es

Jacek Lebiedź
Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology
ul. G. Narutowicza 11/12,
80-233 Gdańsk, Poland
jacekl@eti.pg.edu.pl

Andres Bustillo
Department of Computer Engineering,
University of Burgos
Avda. de Cantabria s/n
09006, Burgos, Spain
abustillo@ubu.es

ABSTRACT

Mobile Science Center is a Polish project that seeks to bring astronomy knowledge to wider social groups through various applications. In its development it is necessary to design a graphical interface that explains a concept that is difficult to assimilate such as spatial proportions and distances. This paper develops a framework to create graphical representations that explain this learning to the target audience of children. Important aspects of this interface are the inclusion of storytelling to guide the educational content, along with feedback and difficulty and accessibility adaptations. Regarding spatial representation, previous works highlight the use of shapes and geometric objects, cartographic tools, reference points, and comparison with known velocities and spaces. The graphical interface proposed is based on a decimal system scale that compares traveling at the speed of light with a person walking. There are 4 proposals that represent the units of this scale with different geometric shapes and interrelated structures, in addition to assigned colors and positions. Future development of this project will apply these ideas to identify the optimal graphical interface so children can learn spatial proportions and distances.

Keywords

graphical interface; interface adaption; astronomy education; STEM education; science centers;

1. INTRODUCTION

Digital applications for STEM (Science, Technology, Engineering and Mathematics) education have been developed in recent years with very positive learning results. Using devices such as smartphones or computers, in addition to gamified interfaces, makes learning more attractive, especially for children. However, one of the branches of science for which these tools have been little applied is astronomy. Understanding the universe is vital to foster care for the Earth, understand human origin and develop critical thinking about pseudo-sciences. Primary education gives a basic knowledge of astronomy, but advanced knowledge is offered by science centers, which are generally located in big urban agglomerations. The location and non-digital format in which this information is offered can make it difficult to access and understand.

Recently, the project "Development of pioneering technologies necessary to launch the Mobile Science Center on the market and preparation of a prototype of the solution" has been created for the implementation of astronomical and astronautical learning applications in Poland [Kam20]. The objective is to guarantee access to wider social groups through the Mobile Science Center, with a specific interest in attracting children to this branch of STEM. For this goal, a total of 15 demonstrative stations have been developed in the first phase. Each application can be handled by non-qualified beginners, deepening their basic science understanding of astronomy, space exploration and space missions.

This project represents a challenge, not only due to the development of 3D modeling, physics and programming, but also for its graphical interface design. Space concepts can be difficult to represent and understand, especially those related to astronomical proportions and distances. Interpreting celestial and planetary maps is difficult for

https://www.doi.org/10.24132/CSRN.3201.12

inexperienced people in this research area [Kar19, Tri18]. Realistically calculating the sizes of space orbs and the distances between them is complicated by the lack of reference points that we are used to on Earth. When we look at maps or terrestrial images, we have human constructions (roads, cities) or geographical features (mountains, lagoons, beaches) to estimate the sizes of what is represented [Har15]. In fact, a land surface without these human or natural landmarks makes interpretation of the scales extremely difficult [Hei07].

Of the 15 applications in the project, 4 require an understanding of astronomical proportions and distances. Their descriptions are as follows:

- *Solar System's virtual tour* (Fig.1). This application shows the path of a spacecraft between the different planets of the Solar system. Its aim is showing the different distances and proportions between these celestial bodies, in addition to the time necessary to make each trip.
- *Hohmann's transfer.* The application's aim is to explain the maneuver required to move a spacecraft from one circular orbit to another using the minimum amount of fuel. For this purpose, it is necessary to understand, apart from concepts such as force and speed, the distances between orbits.
- *Constellation's composition.* This application explains how to recognize different constellations and how they are composed by different stars. It also seeks to show the real distance between the stars, although from Earth they are seen on the same plane.
- *Interactive hologram of the Milky Way.* This application shows what a spiral galaxy looks like and where the Sun is located in the Milky Way. Distance and proportion understanding of Solar System in relation to the galaxy is another goal.

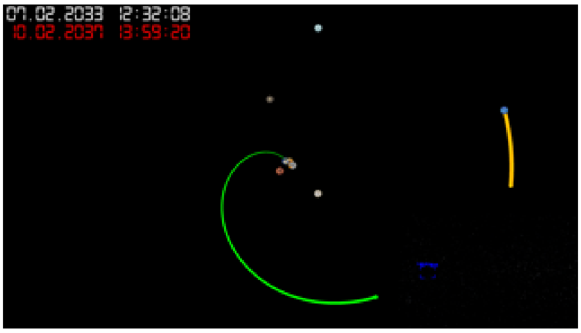


Figure 1. Screenshot of actual prototype of Solar System's virtual tour.

After completing the first phase of the project, prototypes for each application have been developed. Currently, prototypes fulfil basic operations and representations of each educational goal as Figure 1 shows. Nevertheless, for its final implementation, it is necessary to design a graphical interface capable of explaining spatial distances and proportions. Therefore, this paper develops a framework to apply a common interface to spatial representation, with 2D and 3D animations, so children can understand it. The result are different proposals that the work presents and discusses to implement to the final applications.

2. METHODS

To develop the framework that represents spatial proportions and distances, it is necessary to review previous works on Results section. On the one hand, it is studied what specific requirements serve for graphical interfaces with educational purposes that are aimed at children. These aspects take into account the tactile devices that will be used in the Mobile Science Center and its interface elements. The introduction of narrative and its relationship with the learning content are also considered, as well as the graphic representation of distances that are traditionally used. Finally, the importance of feedback and the adaptation of applications to different users are included.

The other part of the review has to study previous astronomy applications with educational goals. Their educational contents are checked and those that have represented spatial proportions and distances are analyzed. The graphic representations of these distances and the relationships between the celestial bodies' proportions are studied, as well as the explanation that the projects' educators have introduced. Any type of interactions that the applications allow the user are also analyzed, along with the changes and choices that occur in the interface as a result. This process gives keys to apply to the framework since they have already demonstrated their educational usefulness.

Once the theoretical bases of the design of this graphic interface project are reviewed, its development proceeds on Discussion section. It begins by defining its virtual elements and the basic interactions with the touch screen that all applications will have. A storytelling is also developed that will be applied to the entire project and will guide the user through all applications, along with the feedback and adaptation of content. Next, the graphical keys that have been obtained from the study of previous astronomy applications are developed to lay a foundation for this framework.

The scale in which the different distances and proportions between celestial bodies will be represented is defined. This scale has the same ratio of proportions in all applications and each of its units is assigned a color that remains the same. From this basis, this paper develops different proposals to graphically represent these relationships using distinct geometric shapes. Its inclusion and interaction with the users to achieve the educational goal is explained. Finally, the adequacy of each proposal for its future implementation in the final phase is discussed.

3. RESULTS

3.1 Design of children graphical interface

Graphical interfaces must take into account their target audience in its design process. This project seeks to represent astronomy proportions and distances so that children understand them. The applications are designed to be installed and managed in a Mobile Science Center, so users will have touch screens. These devices, to which children are accustomed from their first years of life, offer a direct and natural interaction with the digital contents. Tactile devices offer positive learning experiences that bring STEM subjects closer [Str20] and allow better spatial visualization than other types of interfaces [Bay18]. Children are able to directly interact with virtual objects and agents from the applications. There are also widgets such as buttons, controls and labels that offer different information and generate events with user’s interactions [Bot16]. Children can use a finger to tap the desired options, or two fingers to move through the visualized space.

A main aspect of graphical interfaces that has shown great utility for children’s learning is the inclusion of storytelling [Ber18]. Introducing information about spatial visualization and mental rotation skills through a story facilitates understanding and assimilation [Bay18]. Children associate the app's sequence of events with the educational goals they achieve. These should be reinforced with positive feedback when the user is solving a problem, so they also feel that they need this knowledge [Str20]. In addition, narrative is also perfect to guide the application management. Giving them too instructed use, or on the contrary a completely free use, can cause the user boredom or stress. However, story encourages exploration of virtual content and involvement of users with learning [Bay18]. Instead of passive observation, children actively practice with STEM experiences [Str20].

Regarding representation and understanding of spatial distances in a graphical interface, previous works have used shapes and geometric objects as well as patterns. The most common, in a two-dimensional or a three-dimensional way, are lines, points, spheres, squares, triangles, cubes, and cylinders [Bay18]. Children learn mathematics and spatial visualization from school with this type of manipulative objects. When they grow, these primitive forms continue to be a learning base on which other concepts can be fixed. When these are related to abstract thoughts, such as astronomical distance, concrete associations with the world are built [Str20]. To make it easier for them to differentiate, the chosen objects should have different shapes, colors, and scales [Bay18].

Finally, design of applications for children should adapt their content to the user's capabilities [Alm17]. If the users show difficulties in any of the events, they will be offered indications and reinforcements so that they can achieve the educational objective as well. These aids are a type of feedback that can be included in the narration, animations or sound effects [Bay18]. Using sounds with different timbres and rhythms, which fit in with the diegetic world of the application, will reinforce learning. However, for users with possible auditive issues, subtitles have to be added to all the narration of the application, as well as visual effects that match the auditive ones [Pir21].

3.2 Previous astronomy representations

There are some educational projects that have simulated the Solar System so that students could gain a better knowledge of astronomy. Kurniawan et al. [Kur12] designed the Space Exploration 3D game that visualizes and simulates space travel to challenge and engage children. The game is based on the Celestia 3D program, in which the user can travel at different speeds, from 0.001 m/s up to millions of light years/sec. Celestia allows to display space objects in three dimensions, in a scale ranging from a small spacecraft to the entire galaxies, and to interact with them. The learning objectives were the names, shapes, sizes and order of the planets in the Solar System, as well as comparing the rotation period and the revolution period of each planet. This application was pioneer to give the opportunity to realistically visualize the planets while showing their data so users could recognize them. However, this game did not offer a way to appreciate spatial distances and proportions, which is this paper purpose.

The issue of explaining the size relationships between planets was addressed by Gede and Hagitai [Ged17], who were aware that planetary maps are usually produced by astrogeologists for other professionals, but not for the general public. However, in the field of geography, outreach websites, applications and games for cartographic learning are common [Sim11]. Therefore, these authors applied planetary spatial datasets to design a web application for students, which uses cartography as a framework to aid virtual exploration of Solar System planets and moons. The learning goals were the interpretation of sizes and distances in these celestial bodies and the acquisition of extraterrestrial planetary geological knowledge in order to better understand the uniqueness of the Earth.

For this purpose, this application used the concepts of comparison of sizes, as well as of distances by calculating travel time. Since it is easier to estimate the size of an area when we can compare it with reference points, they took as a measure the size of the country of origin with which users are familiar. The application allows you to choose any country in the world and represent it on the surfaces of the planets of the Solar System, thus differentiating the size of the orb. Regarding the representation of distances, the user can form a straight line of desired length on every planet; then the app will calculate the time it takes a person, a Mars Rover, a spacecraft and a car to travel it with their different speeds. The tool does not take into account the topography or the surface of the planet, but allows a realistic estimation of the required movement and time in a space mission.

Another project developed to promote STEM education in the context of the Solar System was PlanetarySystemGO [Cos20]. This augmented reality smartphone application is based on GPS location to teach the relationships of the distances between planets using different scales, coinciding with the purpose of this paper. The objective of this work was the acquisition of general information about celestial bodies and planets orbits, for which multiple choice question sets were used to assess learning.

The application consists of a kind of search for planets in an outdoor space. Players need to walk in the real world to find virtual objects, which are celestial bodies such as stars or planets that appear on the screen of the mobile device. The arena of the game must be defined by choosing a certain scale according to the preferences of the user and the outdoor space available for walking in the real world. The application will use its location in the real world (collected by GPS coordinates) which will be the location of the star of the Solar System in the virtual world and the limit of the arena will correspond to the orbital radius of the last planet. Users earn points as they successfully find the orbits of the planets and the celestial bodies, or also answering the questions correctly. In this way, they will learn data about the planets and perform a realistic comparison of the distances between planets as they move through real space.

4. DISCUSSION

4.1 Design base of graphical interface

Starting with the definition of basic interactions, it is taken into account that children will handle the application on a touch screen, and in a short space of time. Mobile Science Center will be made up of several stations through which children will pass, one after another. Due to these characteristics, interactions must be simple and intuitive so as not to waste time understanding how the application works. To do this, they will interact with widgets such as buttons that the user will tap with a finger to choose options and advance with the learning content. To clarify this operation, the graphical interface will have icons to exemplify its use, as well as textual indications.

As it has proven to be very useful for STEM education, storytelling is introduced in all Mobile Science Center applications. The story puts the user in the role of a Polish astronaut who has to take off in a spacecraft to carry out various missions in space. Each application is a different mission that they must complete, and for which they need some spatial knowledge to do it successfully. In this way, their actions are related to real knowledge, so that children get an idea of the importance of astronomy. Narrative guides them through the applications using dialogues between the protagonist astronaut and the workers of the Polish Space Agency. They explain the details of the mission, the problem to be solved and the knowledge that users must learn. In addition, the applications give feedback in the form of points depending on how they perform the actions or the sooner they get the questions right.

Because of the target audience, the graphical interface that represents interactions and narrative will have an artistic style aimed at children, also including spatial motifs. Dialogues, in addition to being narrated by audio, will be displayed in the form of subtitles. Interactions with the touch screen and all feedback the users receive will also be accompanied by different audio effects and visual icons to identify them. In the event that the user shows difficulties in completing the space missions, each application will give him textual advice or clues so that they can solve them and fulfill the education goal.

4.2 Design of astronomy scale representations

From astronomy previous works, certain educational keys can be obtained to transmit the relationship of spatial proportions and distances:

- Assimilate the tools of the cartographic framework, a subject that is taught since primary school and that is widely disseminated [Ged17].
- Need for comparison and reference points to interpret distances and sizes [Ged17].
- Relate the distance between two points with the speed of an agent in motion and the time that this displacement takes [Ged17].

- Use reference points of the real space in which the user lives to understand the relationship between large distances [Cos20].

Based on these points, this paper develops a graphical scale to explain the distance between celestial objects in the applications. Distances shown range from Hohmann's transfer orbit change, to the Solar System, to the Milky Way and other constellations. The user discovers them and interacts with the celestial bodies as if traveling in a spacecraft. The challenge is how to reflect the proportional distances in the trips so that the children can assimilate this astronomical knowledge.

The starting point on which the application is based is the relation of the distance between celestial objects with the speed of the spacecraft and the time it takes. The spacecraft travels at the speed of light (299,792,458 m/s) to establish a measure that users have as a basis. Still, the speed of light is somewhat too fast for human interpretation, so the next step is to equalize the spacecraft's trips with the distance of a person walks in the same time. For example, it takes 8.3 minutes for the light from the Sun to reach Earth, as does a person walks about 830 meters. Therefore, the distance of the Earth from the Sun (AU) corresponds to 830 meters on the applications scale. In this way, space travel is compared with the day-to-day movements of a person in their real space to understand the proportions.

To explain the distance between celestial bodies not only by numbers, since it could be confusing for children, the cartography scales will be applied. Numerous studies have indicated that spatial thinking and numerical reasoning are closely related. Furthermore, children seem to use analog mental transformation strategies for spatial scaling [Mix12, New15]. Therefore, the understanding of scale relations is similar to proportional equivalence. This means that users can understand that the distances displayed in different sizes within the application correspond to the proportions of the real world [Möh18].

Therefore, this project develops a linear scale from the base measure of the distance that light travels for 1 minute and which is equivalent to the 100 meters that a person walks. This value has smaller units created for travel between orbits and larger units for the Solar System and stars. The relationship between these units will follow a decimal system as can be seen in Table 1. There are 2 lower units that are equivalent to 0.1 minutes (6 seconds) and 10 meters traveled, and 0.01 minutes (0,6 seconds) and 1 meter distance. For the higher units, the next step is 10 minutes and 1 km of distance and, the third measure corresponds to 100 minutes of travel of light and 10 km of distance traveled by a person.

Those measurements will be enough for the Solar System, but the trip between stars will need greater distances. Therefore, for Milky Way and Constellations applications, the base measurement of the speed of light in 1 min and 100 meters will become 1 light year and 10 billion (10^{12}) km. In addition, a higher unit is added that is equivalent to a thousand light years and 10,000 billion km. Although the use of a logarithmic scale was considered, the idea was discarded due to the added difficulty for the target audience of this application. There are studies that show the importance of children's familiarity with numbers and the decimal system when creating calculation patterns [Moe09]. Consequently, from the age of 7 they should have no problems understanding this scale [Möh18].

Orbits and Solar System		Color	Milky Way and Constellations	
Time unit [min]	Distance [m]		Time unit [l.y.]	Distance [billion of km]
0.01	1	Red	0.01	0.1
0.1	10	Orange	0.1	1
1	100	Yellow	1	10
10	1,000	Green	10	100
100	10,000	Blue	100	1,000
		Violet	1000	10,000

Table 1. Astronomy scale units of this project

4.3 Graphical proposals for this project

To graphically represent the proposed scale, shapes and geometric objects are used, with relationships between them. Besides, proposals in a two-dimensional or a three-dimensional representation are offered. The interface follows psychology of color principles to help distinguish the closest or farthest distances. The 2D images of the interface have the ability to generate different visual signals according to the differences in size, contrast, luminance and/or color [von15]. When objects have the same contrast, shape and size, warm colors make the observer think that they are closer than cold colors [Egu83]. To represent these scales, the interface also needs uniform backgrounds to improve color stability and related perceptions [Dre20]. In addition, the order in which these elements appear when they are grouped also influences. The object with the lowest position in the plane will have an even greater probability of appearing closer to the human observer [Gui04].

There are 4 proposals which are explained below and shown in Figure 2:

1. Each unit of the scale is represented in 2D with a bar and the color that has been assigned to it.

Each bar has the same length even if the decimal system between them is maintained. The bars are ordered by rows, those of smaller units in the lower ones and the larger units in the upper ones.

2. Smaller units of the scale are represented in 2D and as they increase in the decimal system their composition increases until the last unit is represented in 3D. Each unit is represented by a geometric shape that forms when the next unit is joined. The progression is: point, line, bar, rectangle, square, and cuboid.

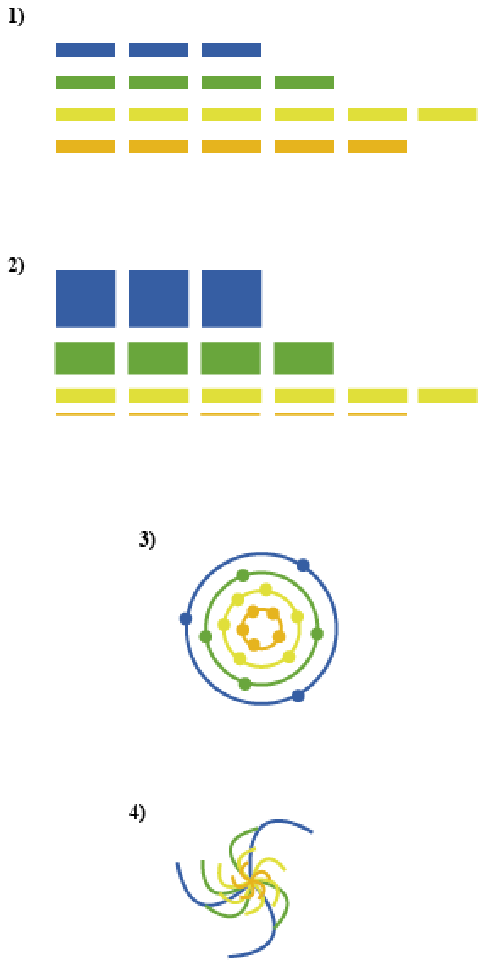


Figure 2. Example of the 4 proposals representing 346.5 light years

3. Each unit of the scale is represented in 2D in the form of concentric circles, the larger units are the outer circles and the smaller units are the inner ones. To represent the number of units that form the distance, each circumference will have the same number of points on it. This representation is proposed because it reminds the icons that are used for planets and their orbits.

4. Each unit of the scale is represented in 2D as a set of spiral lines coming from the same point. The larger units have longer lines and the smaller units have shorter lines. To represent the number of units that form the distance, a line is drawn for each one. This representation is proposed because it reminds the icons that are used for galaxies.

Following psychology of color principles, the smallest step of the scale uses a warm color, red, while it cool downs changing to orange and yellow, green and blue in the middle, and violet in the largest. The chosen colors have the same luminance value with the same background to maintain the contrast and sensation of distance. This relationship of the scale with the necessary time and the equivalence of the distance is explained through the astronaut's dialogues and the explanation of the mission. In each application, users are asked to travel from one celestial body to another that is at a distance. The distance is graphically represented in the interface and the user has to calculate the exact distance that is asked, with the decimal system of the scale always visible to check. For color blind participants, the graphic representations will be accompanied by signs with which they will be able to identify them. An answer is chosen from 3 possible options and the user gets more points doing it right on the first try. Table 2 shows an example of proposal 1 representation of distances from the Sun to the rest of the planets that applies to the Solar System application.

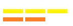
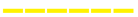
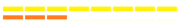
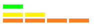
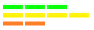
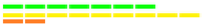
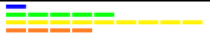
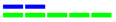
Planet	Time [min]	Reference distance [m]	Scale representation
Mercury	3.2	320	
Venus	6.0	600	
Earth	8.3	830	
Mars	12.4	1,240	
Jupiter	34.2	3,420	
Saturn	79.2	7,920	
Uranus	159.4	15,940	
Neptune	250.0	25,000	

Table 2. Relations of time, distance and scale representation from the Sun to every planet

In addition to this scale and graphic representation, since Mobile Science Center will travel to different cities, the representation will be customized for those citizens. While showing the relative distances between space objects, the applications will also show a real map of the city for short distances and an Earth map for long distances. A representative building of the city will be chosen to place the origin position of the spacecraft and hence the position of the celestial objects following the scale. For example, the users will visualize that the store 320 meters from the building would be equivalent to the displacement from Sun to Mercury, but to go to Neptune they would have to travel 25 km to the nearby town. For the routes within Milky Way and Constellations, the Earth map will be used to simulate the equivalence of distances.

These applications are meant to run one after another in a short space of time. Therefore, when rendering the animations of the spacecraft's journeys it is not possible to take minutes. In order for the scales and the required time to be understood as such, it is necessary to introduce another animation. A clock is included for short trips and a calendar for long trips. In this way, although the animations only take a few seconds, which will be proportional in each one, users will see the passage of time in the change of times and dates.

5. CONCLUSION

This paper has obtained a framework to develop a graphical interface for astronomy applications with educational goals. It has focused on the visual representation of spatial proportions and distances with the aim that children understand them. This framework is used on touch devices that offer easy interaction. Learning is guided by storytelling that introduces the educational content and the actions that the user must perform. The challenges have to be regulated in difficulty and accessibility, always giving feedback through sound and visual elements. Previous works have also highlighted the importance of using shapes and geometric objects, cartographic tools, reference points and comparison with known velocities and spaces.

Applications designed for this project’s Mobile Science Center include the narrative of an astronaut conducting space missions to introduce learning. The representation of spatial proportions occurs when the astronaut travels in a spacecraft from one celestial body to another. The distances are represented following a decimal system scale that compares traveling at the speed of light with a person walking. These measurements are counted by minutes and meters at distances within orbits and the Solar System, and light years and billions of km within Milky Way and constellations. In the graphic representation, each unit of the scale has a color and position assigned according to its proximity or distance. Four proposals are made to represent the units with different geometric shapes and interrelated structures. The applications also offer animations showing elapsed travel time and personalized comparisons of distance traveled.

The framework and the 4 graphical proposals developed in this paper will be programmed and introduced in the current prototypes of the Mobile Science Center. Once its correct functioning is verified, tests will be carried out with children to verify their satisfaction with all the characteristics described. Motivation, ease of use and level of learning will be measured, being especially relevant the division into groups that manage the different proposals. In this way it will be possible to identify which representation of the graphical interface is more useful for learning spatial distances. Finally, the proposal identified will be implemented in the final version of the applications.

6. ACKNOWLEDGMENTS

This work was partially supported by grant FPU18/04688 from the Spanish Ministry of Science, Innovation, and Universities. and grant POIR.01.01.01-00-1753/20 from the Polish National Centre for Research and Development.

7. REFERENCES

[Kam20] Kamińska A. et al. (2020): *Development of pioneering technologies necessary to launch the Mobile Science Center on the market and preparation of a prototype of the solution* (in Polish). Application for project co-financing POIR.01.01.01-00-1753/20.

[Kar19] Karolyi, M., Krejčí, J., ŠčAvnický, J., Vyškovský, R., & Komenda, M. (2019). Tools for development of interactive web-based maps: application in healthcare. *WSCG 2019 - Short papers proceedings*. https://doi.org/10.24132/csrn.2019.2902.2.1

[Tri18] Tripathi, G., Etemad, K., & Samavati, F. (2018). Single image summary of time-varying Earth-features. *WSCG 2018 - Full papers proceedings*. https://doi.org/10.24132/csrn.2018.2801.8

[Har15] Hargitai H., Page D., Canon-Tapia E., Rodrigue C.M. (2015). Classification and characterization of planetary landforms, in: H. Hargitai, A. Kereszturi (Eds.), Encyclopedia of Planetary Landforms, Springer. http://dx.doi.org/10.1007/ 978-1-4614-3134-3.

[Hei07] Heiken, G., & Jones, E. (2007). *On the Moon: The Apollo Journals*. Springer Science & Business Media.

[Bot16] Bottino, A., Martina, A., Strada, F., & Toosi, A. (2016). GAINE – A portable framework for the development of edutainment applications based on multitouch and tangible interaction. *Entertainment Computing*, *16*, 53–65. https://doi.org/10.1016/j.entcom.2016.04.001

[Ber18] Bers, M. U. (2018). Coding as a playground: programming and computational thinking in the early childhood classroom. Routledge.

[Str20] Strawhacker, A., Verish, C., Shaer, O., & Bers, M. (2020). Young Children’s Learning of Bioengineering with CRISPEE: a Developmentally Appropriate Tangible User Interface. *Journal of Science Education* *and Technology*, *29*(3), 319–339. https://doi.org/10.1007/s10956-020-09817-9

[Bay18] Baykal, G., Alaca, I. V., Yantaç, A., & Göksun, T. (2018). A review on complementary natures of tangible user interfaces (TUIs) and early spatial learning. *International Journal of Child-Computer Interaction*, *16*, 104–113. https://doi.org/10.1016/j.ijcci.2018.01.003

[Alm17] Almurayh, A., & Semwal, S.K. (2017). CoUIM: crossover user interface model for inclusive computing. *WSCG 2017 - Short papers proceedings*.

[Pir21] Pires, A. C., Bakala, E., González-Perilli, F., Sansone, G., Fleischer, B., Marichal, S., & Guerreiro, T. (2021). Learning maths with a tangible user interface: Lessons learned through participatory design with children with visual impairments and their educators. *International Journal of Child-Computer Interaction*, 100382. https://doi.org/10.1016/j.ijcci.2021.100382

[Kur12] Kurniawan, R., Rohman, A. S., & Husni, E. M. (2012). The design and analysis of the Space Exploration 3D simulation game. *2012 International Conference on System Engineering and Technology (ICSET)*. https://doi.org/10.1109/icsengt.2012.6339307

[Ged17] Gede, M., & Hargitai, H. (2017). An online planetary exploration tool: "Country Movers". *Acta Astronautica*, *137*, 334–344. https://doi.org/10.1016/j.actaastro.2017.04.028

[Sim11] Simonné-Dombóvári, E. (2011). *Development of interactive web applications in teaching cartographical skills (for 4th, 6th and 8th grades of high schools)* (Doctoral dissertation, PhD thesis. Eötvös Loránd University, Budapest).

[Cos20] Costa, M.C., Manso, A., Santos, P., Patrício, J., Vital, F.M., Rocha, G.M., & Alegria, B.M. (2020). An Augmented Reality Information System Designed to Promote STEM Education. *SIIE*.

[Mix12] Mix, K. S., & Cheng, Y. L. (2012). The Relation Between Space and Math. *Advances in Child Development and Behavior Volume 42*, 197–243. https://doi.org/10.1016/b978-0-12-394388-0.00006-x

[New15] Newcombe, N. S., Levine, S. C., & Mix, K. S. (2015). Thinking about quantity: the intertwined development of spatial and numerical cognition. *WIREs Cognitive Science*, *6*(6), 491–505. https://doi.org/10.1002/wcs.1369

[Möh18] Möhring, W., Frick, A., & Newcombe, N. S. (2018). Spatial scaling, proportional thinking, and numerical understanding in 5- to 7-year-old children. *Cognitive Development*, *45*, 57–67. https://doi.org/10.1016/j.cogdev.2017.12.001

[Moe09] Moeller, K., Pixner, S., Kaufmann, L., & Nuerk, H. C. (2009). Children's early mental number line: Logarithmic or decomposed linear? *Journal of Experimental Child Psychology*, *103*(4), 503–515. https://doi.org/10.1016/j.jecp.2009.02.006

[von15] von der Heydt, R. (2015). Figure–ground organization and the emergence of proto-objects in the visual cortex. *Frontiers in Psychology*, *6*. https://doi.org/10.3389/fpsyg.2015.01695

[Egu83] Egusa, H. (1983). Effects of Brightness, Hue, and Saturation on Perceived Depth between Adjacent Regions in the Visual Field. *Perception*, *12*(2), 167–175. https://doi.org/10.1068/p120167

[Dre20] Dresp-Langley, B., & Reeves, A. (2020). Color for the perceptual organization of the pictorial plane: Victor Vasarely's legacy to Gestalt psychology. *Heliyon*, *6*(7), e04375. https://doi.org/10.1016/j.heliyon.2020.e04375

[Gui04] Guibal, C. R. C., & Dresp, B. (2004). Interaction of color and geometric cues in depth perception: When does red mean near? *Psychological Research Psychologische Forschung*, *69*(1–2), 30–40. https://doi.org/10.1007/s00426-003-0167-0

Balanced Feature Fusion for Grouped 3D Pose Estimation

Jihua Peng[1,2], *Yanghong Zhou*[1], *P.Y. Mok*[1,2,*]

[1]*Institute of Textiles & Clothing, The Hong Kong Polytechnic University, Hong Kong.*
[2]*Laboratory for Artificial Intelligence in Design, Hong Kong Science Park, Hong Kong.*
* tracy.mok@@polyu.edu.hk

ABSTRACT

3D human pose estimation by grouping human body joints according to anatomical relationship is currently a popular and effective method. For grouped pose estimation, fusing features of different groups together effectively is the key step to ensure the integrity of whole body pose prediction. However, the existing methods for feature fusion between groups require a large number of network parameters, and thus are often computational expensive. In this paper, we propose a simple yet efficient feature fusion method that can improve the accuracy of pose estimation while require fewer parameters and less calculations. Experiments have shown that our proposed network outperforms previous state-of-the-art results on Human3.6M dataset.

Keywords
3D Human Pose Estimation, Grouping Feature Fusion, Anatomical Relationships

1 INTRODUCTION

3D human pose estimation is the computer vision task of estimating the articulated 3D joint locations of a human body from an input image or video, which has received much research attention because it supports many applications including cloth parsing [Don14a], surveillance [Liu18a], augmented reality [Lin10a], action prediction [Luv18a].

The main stream convolutional neural networks-based methods for 3D human pose estimation mainly follow two approaches: (1) directly regressing 3D coordinates of each joint from input images or sequences; (2) first predicting 2D joint coordinates from images and then matching these 2D key points to 3D coordinates. The second approach significantly outperforms the first one, since these methods benefit from the high performance of intermediate 2D pose detectors [Che18a, Sun19a].

Some researchers proposed to group human joints into parts, such as arms, legs and torso, based on kinematics and human anatomy to improve prediction accuracy [Cai19a, Fan18a, Lee18a, Wan19a]. They first encode each part to get the corresponding features, and then fuse these features together to get a complete human pose. Although the features of each part are independently predicted, different parts affect each other and their features are also related mutually. For example, there is interaction between the arm and the head for poses in "Eating", while the hand and the foot are closely related when performing poses "Running". In these methods, a feature fusion module is used to fuse the features of different parts together; while many existing feature fusion methods just use fully connected layers, resulting in large number of parameters and high computation costs.

In this paper, we propose a network framework with an optimized feature fusion (OFF) module for 3D pose estimation, as shown in Figure 1. Our method improves the accuracy of 3D pose estimation while requires fewer parameters and has lower computational complexity.

The remaining of this paper is organized as follows. We first review the related works on 3D human pose estimation. Then, we describe the proposed method in detail and experimentally demonstrate the effectiveness of our method by comparing with state-of-the-art methods and ablation studies. Finally, we conclude our work, and discuss limitations and future research directions.

2 RELATED WORK

3D human pose estimation has been studied since a very early time. Early traditional methods utilized pictorial structures to predict 3D coordinates of human joints. These methods usually require tedious manual operations and a large amount of calculations, and get bad results when encountering some complex poses. Thanks to the rapid development of deep learning methods, convolutional neural networks-based methods have become the main stream and achieved very promising results in recent years. These methods can be classified into two categories. Some early works

https://www.doi.org/10.24132/CSRN.3201.13

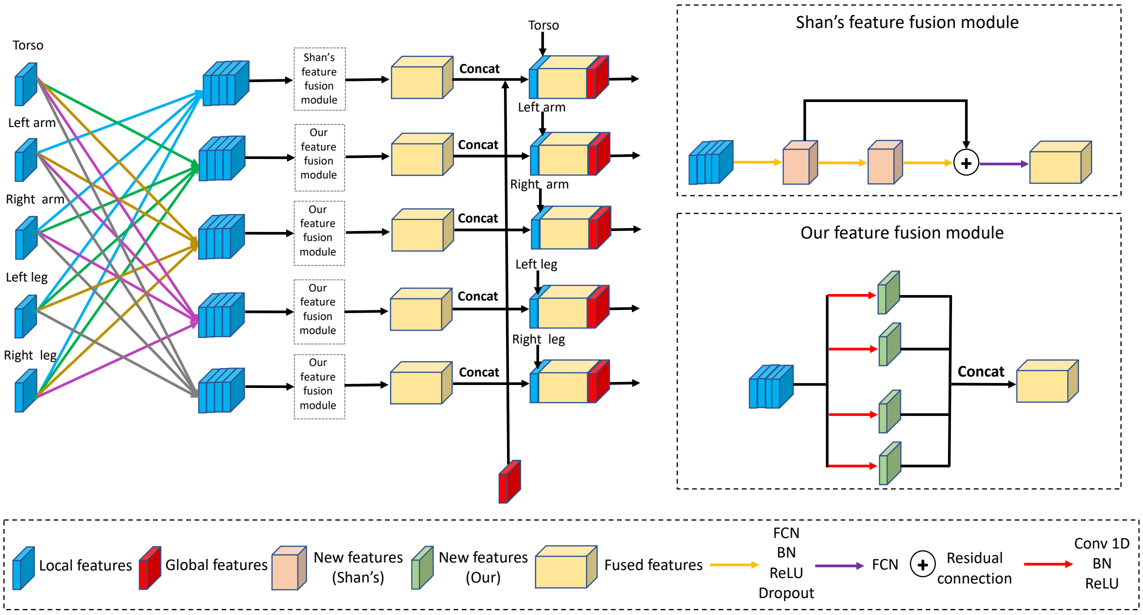


Figure 1: Our and Shan's [Sha21a] feature fusion modules. There are five groups of local features. We both concatenate the four groups of local features together. Then Shan [Sha21a] uses three modules based on FCN to raise the feature dimension and outputs fused features through a FCN, while we use four modules based on 1D convolution to conduct discriminative dimensionality reduction and obtain four new features respectively. We concatenate them together to get the fused features. For the torso part, we still use the feature fusion method proposed by [Sha21a]. Finally, the remaining local features, fused features, and global features are concatenated together to become a new feature of the body part. After performing such feature fusion five times, five new features of five body parts are obtained. (FCN) - Fully Connected Layer. (BN) - 1D Batch Normalization. (Conv 1D) - 1D convolution.

[Pav17a, Tek16a] directly predict the 3D human joints from the input image or video through neural networks, which is called one-step regression method. Recently, benefit from recent advancement in 2D pose detectors [Che18a, Liu18a, Liu19a, Sun19a, Hua20a], many methods first detect the coordinates of the 2D pose joints, and then use these 2D coordinates to regress the human pose joint in the 3D space. Among the methods of regressing 3D coordinates, the method using temporal information has the promising results.

The Long Short-Term Memory (LSTM) model is the first sequence-to-sequence model that extract 3D human pose information from videos. It mainly encodes the coordinates of 2d human joint, and then decodes them into coordinates in 3d space [Hos18a]. Lee et al. [Lee18a] design a propagating LSTM structure based on joint interdependency to learn the spatial position relationship of each joint of the human body. However, LSTM cannot process multiple frames in parallel, but store them sequentially in memory resulting in many parameters. To address this problem, temporal convolutional network (TCN) is proposed for 3D human pose estimation [Pav19a]. It performs 1D convolutions over 2d input pose sequences with fewer parameters. Liu et al. [Liu20a] apply attention mechanism to TCN, which determine key frames and output tensor in every layer. Chen et al. [Che21a] transform TCN to predict both direction and length of the bones.

Many current methods [Par18a, Wan19a, Zhe20a, Zen20a, Sha21a] group body parts to predict 3d human pose according to the anatomical relationship. Park et al. [Par18a] propose to divide human joints into 5 non-overlapping groups (torso, left arms, right arms, left legs, right legs). Then the features learned by these 5 groups are averaged to produce the feature of the whole pose. Wang et al. [Wan19a] define the degrees of freedom (DOF) and model limbs with higher DOFs and torso with lower DOFs. In this way, various body parts with different DOFs can supervise each other, leading to more reasonable prediction results. Zheng et al. [Zhe20a] treat each joint as a group and design a dual attention module to learn the feature relationship between each group. Zeng et al. [Zen20a] divide the human body into local groups of joints and develop a network to learn internal dependencies within each group and weak dependencies among groups. Shan et al. [Sha21a] split the human body into five groups (torso, left/right arms, left/right legs), encode the features of each group separately, and design a feature fusion module to fuse the 5 groups of features to obtain a complete human body.

3 METHOD

We use the structure of [Sha21a] as the baseline from our network framework (Figure 1). The input of the network is the coordinate of the 2D pose that have been predicted from the images. It first processes the input 2D pose joints of target pose and other poses to obtain the positional and temporal information, and then divides these information into five groups (torso, left arm, right arm, left leg, right leg). After all the information encoded by the TCN network, we propose a new optimized feature fusion (OFF) module to fuse the five groups of features. We use the OFF module to fuse four groups of features (e.g., left and right hands, left and right legs) and then concatenate them with the remaining group of features (torso) to form a new torso feature. After performing five times of feature fusion, we get five new groups of features (torso, left/right hands, left/right legs). Finally, we decode the five new groups of features to get the coordinates of the joints of the five body parts and concatenate them together to get the complete human pose.

Our proposed feature fusion module is compared with that of [Sha21a] in Figure 1. Both of our inputs are five local features of five body parts, four of which are concatenated together. Shan et al. [Sha21a] use three modules composed of fully connected layer, 1D batch normalization [Iof15a], rectified linear units [Nai10a] and dropout [Sri14a] to raise the feature dimension to obtain the fused features. Shan et al. [Sha21a] also use residual connections to solve the problem of gradient explosion and disappearance when the number of network layers is deep. However, Cheng et al. [Che15a] illustrate that fully connected layers generally involve over 90% of the network parameters and generate redundancy of parameters in deep neural networks. Similarly, the method of [Sha21a] also generate some redundant information, since the features of each group are connected to each other when using fully connected layers to fuse the features of the four body parts. But in fact some body parts are not strongly related in some actions. For example, there is no strong correlation between head and hands in the action âWalkingâ. Compared with the method of [Sha21a], we only use four different 1D convolutions followed by 1D batch normalization [Iof15a] and rectified linear units [Nai10a], which can reduce the amount of parameters greatly and ensure that each part learns enough information.

Specifically, we denote each group of local features as F_i, $F_i \in \mathbb{R}^{B\times C}$, where B and C are the batch size and the number of channels. We concatenate the four groups of local features among them together according to the channel dimension as $[F_1,...,F_4]$. Then we aim to use the discriminative dimensionality reduction method [Sub17a, Gao19a] to make the features separate and obtain a new discriminative feature $F_i^{'}$ for group i by aggregating features from four groups. Therefore, we need to learn the transformation W for concatenated four groups of features $[F_1,...,F_4]$. We denote W as a 1D convolution with 1 stride and 1 kernel size. The input and output channels of this 1D convolution are $4C$ and C respectively. Therefore, this formula for obtaining the discriminative feature $F_i^{'}$ is as follows

$$F_i^{'} = BatchNorm1D\left(Conv1D\left([F_1,...,F_4]\right)\right) \tag{1}$$

After performing four different groups of 1D convolution, batch normalization [Iof15a] and rectified linear units [Nai10a], we get four new groups of features $F_1^{'},F_2^{'},F_3^{'},F_4^{'}$. These new features are concatenated to form the fused features $\left[F_1^{'},F_2^{'},F_3^{'},F_4^{'}\right]$. Subsequently, the remaining local body features F_5 and global features (target pose) are concatenated with the fused feature and all the features are sent to the decoding layer to predict the coordinates of the 3D human pose. In addition, considering that the number of joints in the torso is the largest among the five groups, we still use the feature fusion method based on the fully connected layer proposed by [Sha21a] for the torso group. The remaining four groups of joints adopt our proposed optimized feature fusion (OFF) module.

4 EXPERIMENTS

4.1 Datasets and Evaluation

Dataset We evaluate our model on the public dataset Human3.6M [Ion13a]. Human3.6M is an indoor scenes dataset collected by motion capture system with 3.6 million video frames. It has 11 professional actors wearing clothes with markers which record the coordinates of each human body joint. These actors perform 15 actions in daily life under 4 synchronized camera views, such as walking dogs, photoing, sitting, greeting, eating and so on.

Following previous studies [Mar17a, Pav17a, Fan18a, Pav19a, Liu20a, Sha21a], we adopt five subjects (S1, S5, S6, S7, S8) for training and two subjects (S9 and S11) for testing. We use the commonly used protocols to evaluate our experimental results. Our model is trained in the PyTorch framework on one GeForce RTX 3070 GPU.

Evaluation protocol is denoted as Mean Per Joint Position Error (MPJPE) that is the average Euclidean distance between estimated human joint coordinates and ground-truth human joint coordinates. It is the most popular standard for evaluating the 3D human pose estimation.

Table 2 illustrates the computational complexity of different models. We compare our method with [Pav19a]

Method	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Lee et al. [Lee18a]	32.1	36.6	34.4	37.8	44.5	49.9	40.9	36.2	44.1	45.6	35.3	35.9	37.6	30.3	35.5	38.4
Pavllo et al. [Pav19a]	35.2	40.2	32.7	35.7	38.2	45.5	40.6	36.1	48.8	47.3	37.8	39.7	38.7	27.8	29.5	37.8
Liu et al. [Liu20a]	34.5	37.1	33.6	34.2	32.9	37.1	39.6	35.8	40.7	41.4	33.0	33.8	33.0	26.6	26.9	34.7
Zeng et al. [Zen20a]	34.8	32.1	**28.5**	30.7	31.4	36.9	35.6	30.5	38.9	40.5	32.5	31.0	29.9	**22.5**	24.5	32.0
Shan et al. [Sha21a]	29.5	30.8	28.8	29.1	30.7	**35.2**	31.7	27.8	**34.5**	**36.0**	30.3	29.4	28.9	24.1	24.7	30.1
Ours (T = 243 GT)	**27.8**	**28.2**	28.7	**27.7**	**30.4**	35.8	**31.1**	**26.6**	34.6	36.8	**30.2**	**28.3**	**28.0**	23.6	**24.1**	**29.5**

Table 1: Reconstruction error on Human3.6M under **evaluation protocol** with MPJPE (mm). The input 2d pose is ground truth. The lowest reconstruction error is bold, and the second lowest is underlined. (GT) - ground-truth.

Method	Parameters	≈ FLOPs	MPJPE
Pavllo et al. [Pav19a]	16.95M	33.87M	37.8
Shan et al. [Sha21a]	41.78M	36.03M	30.1
Ours	28.38M	22.39M	29.5

Table 2: Comparison with the computational complexity of different models. All models are trained on ground-truth 2D poses under **evaluation protocol** with MPJPE (mm). The input of all models is a 2D pose sequence of 243 frames. Both parameters and FLOPs of us and [Sha21a] are calculated at stage 2 or 3.

and [Sha21a] in terms of the number of model parameters and an estimate of the floating-point operations (FLOPs) because we all use the TCN as the baseline. FLOPs are the computational power required for forward propagation, which reflects the level of performance required for hardware such as GPU. In our experiments, we mainly calculate the computational power consumption of the convolutional layer, fully connected layer, BatchNorm and ReLU when the model is forward propagated. Both parameters and FLOPs of us and [Sha21a] are calculated at stage 2 or 3. Although our model has more parameters than that of [Pav19a], our MPJPE result is 8.3mm lower. In addition, the number of our model parameters is much less than that of [Sha21a] and our reconstruction error (MPJPE) is also lower. As for FLOPs, our model only has 22.39M which is the least among all models, almost half of the model proposed by [Sha21a].

4.2 Comparison with State-of-the-Art Methods

We compare our results with state-of-the-art works in recent years on the public dataset Human3.6M. Table 1 shows the comparison results between our method and recent methods on Human3.6M under **evaluation protocol**. We use ground-truth 2D poses as input and train under the recpetive field of 243 frames. Our method reaches 29.5mm in MPJPE, which is 0.6mm better than the best result. Besides, our model achieves the state-of-the-art in terms of multiple actions and also reaches the second best result in some complex actions, such as "Eat", "Photo", "Sitting", "Sitting Down", "Walk". The reason why we do not achieve the best results in these complex actions may be that these actions have severe deep ambiguity and occlusion, which requires other groups of joints to provide more information to the occluded or deeply ambiguous joints during feature fusion. Our method reduces nearly half of the parameters in the feature fusion stage, so it may lose some information for the learning of these complex actions.

4.3 Ablation Studies

In order to verify the validity of each part in our model, we perform ablation experiments on whether the torso part uses 1D convolution or fully connected layers on Human3.6M under **evaluation protocol** with MPJPE (mm). Our network takes the ground truth of 2D poses as input. Table 3 shows ablation experiments of different methods used in our network at stage 3. If all feature fusion modules use 1D convolution, although the parameter amount is the lowest, only 24.96M, the MPJPE is only reduced by 0.1mm. However, if the fully connected layer is used for feature fusion for the torso part and the 1D convolution is used for other parts for fusion, the MPJPE is reduced by 0.6mm and the parameter quantity is also much lower than the baseline. This is because the torso part has more joints than other parts and needs to learn more information during feature fusion. 1D convolution feature fusion method makes it not enough to learn features. The baseline uses FCN to perform feature fusion on five parts, so redundant information is learned for the part with few joints. Therefore, in order to balance the number of features that need to be learned in each part, we use the feature fusion module consisting of 1D convolution and FCN.

Method	MPJPE(mm)	△	Parameters
Baseline (FCN (all parts))	30.1	–	41.78M
+OFF(Conv 1D)	30.0	0.1	24.96M
+OFF(Conv 1D + FCN (torso))	29.5	0.6	28.38M

Table 3: Ablation study of different methods in our model at stage 3. OFF refer to optimized feature fusion proposed by us. The MPJPE results takes ground-truth 2D poses as input and is trained on Human3.6M under **evaluation protocol**. (FCN) - Fully Connected Layer. (Conv 1D) - 1D convolution.

5 CONCLUSION

We propose an optimized feature fusion module for grouped human pose in this paper. Compared with the feature fusion module [Sha21a] composed of fully connected layer, our optimized feature fusion module can use fewer parameters to fuse different groups of human pose features and it can also reduce reconstruction errors. Experimental results prove that our method advances state-of-the-art performance on Human3.6M dataset. However, there are still limitations in our current method: we still require three stages of training and the improvement in prediction accuracy is observed but not in a significant percentage. In the future, we will research on one-stage training of the entire network to further improve prediction accuracy.

6 ACKNOWLEDGMENTS

This research is funded by the Laboratory for Artificial Intelligence in Design (Project Code: RP1-1) under the InnoHK Research Clusters, Hong Kong Special Administrative Region Government.

7 REFERENCES

[Don14a] Dong, J., Chen, Q., Shen, X., Yang, J., Yan, S. Towards unified human parsing and pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 843-850). 2014.

[Liu18a] Liu, J., Ni, B., Yan, Y., Zhou, P., Cheng, S., Hu, J. Pose transferrable person re-identification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4099-4108). 2018.

[Lin10a] Lin, H. Y., Chen, T. W. Augmented reality with human body interaction based on monocular 3D pose estimation. In International Conference on Advanced Concepts for Intelligent Vision Systems (pp. 321-331). Springer, Berlin, Heidelberg. 2010.

[Luv18a] Luvizon, D. C., Picard, D., Tabia, H. 2d/3d pose estimation and action recognition using multitask deep learning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5137-5146). 2018.

[Che18a] Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J. Cascaded pyramid network for multi-person pose estimation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7103-7112). 2018.

[Sun19a] Sun, K., Xiao, B., Liu, D., Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5693-5703). 2019.

[Cai19a] Cai, Y., Ge, L., Liu, J., Cai, J., Cham, T. J., Yuan, J., Thalmann, N. M. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2272-2281). 2019.

[Fan18a] Fang, H. S., Xu, Y., Wang, W., Liu, X., Zhu, S. C. Learning pose grammar to encode human body configuration for 3d pose estimation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1). 2018.

[Lee18a] Lee, K., Lee, I., Lee, S. Propagating lstm: 3d pose estimation based on joint interdependency. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 119-135). 2018.

[Wan19a] Wandt, B., Rosenhahn, B. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7782-7791). 2019.

[Pav19a] Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7753-7762). 2019.

[Che21a] Chen, T., Fang, C., Shen, X., Zhu, Y., Chen, Z., Luo, J. Anatomy-aware 3d human pose estimation with bone-based pose decomposition. IEEE Transactions on Circuits and Systems for Video Technology, 32(1), 198-209. 2021.

[Liu20a] Liu, R., Shen, J., Wang, H., Chen, C., Cheung, S. C., Asari, V. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5064-5073). 2020.

[Zen20a] Zeng, A., Sun, X., Huang, F., Liu, M., Xu, Q., Lin, S. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In European Conference on Computer Vision (pp. 507-523). Springer, Cham. 2020.

[Iqb20a] Iqbal, U., Molchanov, P., Kautz, J. Weakly-supervised 3d human pose learning via multi-view images in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5243-5252). 2020.

[Mar17a] Martinez, J., Hossain, R., Romero, J., Little, J. J. A simple yet effective baseline for 3d human pose estimation. In Proceedings of the IEEE international conference on computer vision (pp. 2640-2649). 2017.

[Sha21a] Shan, W., Lu, H., Wang, S., Zhang, X., Gao, W. Improving Robustness and Accuracy via Relative Information Encoding in 3D Human Pose Estimation. In Proceedings of the 29th ACM International Conference on Multimedia (pp. 3446-3454). 2021.

[And09a] Andriluka, M., Roth, S., Schiele, B. Pictorial structures revisited: People detection and articulated pose estimation. In 2009 IEEE conference on computer vision and pattern recognition (pp. 1014-1021). IEEE. 2009.

[Ami13a] Amin, S., Andriluka, M., Rohrbach, M., Schiele, B. Multi-view pictorial structures for 3d human pose estimation. In Bmvc (Vol. 1, No. 2). 2013.

[Bel14a] Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., Ilic, S. 3D pictorial structures for multiple human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1669-1676). 2014.

[Hos18a] Hossain, M. R. I., Little, J. J. Exploiting temporal information for 3d human pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 68-84). 2018.

[Pav17a] Pavlakos, G., Zhou, X., Derpanis, K. G., Daniilidis, K. Coarse-to-fine volumetric prediction for single-image 3D human pose. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7025-7034). 2017.

[Tek16a] Tekin, B., Rozantsev, A., Lepetit, V., Fua, P. Direct prediction of 3d body poses from motion compensated sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 991-1000). 2016.

[Par18a] Park, S., Kwak, N. 3d human pose estimation with relational networks. arXiv preprint arXiv:1805.08961. 2018.

[Wan19a] Wang, J., Huang, S., Wang, X., Tao, D. Not all parts are created equal: 3d pose estimation by modeling bi-directional dependencies of body parts. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 7771-7780). 2019.

[Zhe20a] Zheng, X., Chen, X., Lu, X. A joint relationship aware neural network for single-image 3D human pose estimation. IEEE Transactions on Image Processing, 29, 4747-4758. 2020.

[Iof15a] Ioffe, S., Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). PMLR. 2015.

[Nai10a] Nair, V., Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In Icml. 2010.

[Sri14a] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958. 2014.

[Che15a] Cheng, Y., Yu, F. X., Feris, R. S., Kumar, S., Choudhary, A., Chang, S. F. An exploration of parameter redundancy in deep networks with circulant projections. In Proceedings of the IEEE international conference on computer vision (pp. 2857-2865). 2015.

[Ion13a] Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence, 36(7), 1325-1339. 2013.

[Liu18a] Liu, S., Li, Y., Hua, G. Human pose estimation in video via structured space learning and halfway temporal evaluation. IEEE Transactions on Circuits and Systems for Video Technology, 29(7), 2029-2038. 2018.

[Hua20a] Hua, G., Li, L., Liu, S. Multipath affinage stackedâhourglass networks for human pose estimation. Frontiers of Computer Science, 14(4), 1-12. 2020.

[Liu19a] Liu, S., Hua, G., Li, Y. 2.5 D human pose estimation for shadow puppet animation. KSII Transactions on Internet and Information Systems (TIIS), 13(4), 2042-2059. 2019.

[Sub17a] Su, B., Ding, X., Wang, H., Wu, Y. Discriminative dimensionality reduction for multidimensional sequences. IEEE transactions on pattern analysis and machine intelligence, 40(1), 77-91. 2017.

[Gao19a] Gao, Y., Ma, J., Zhao, M., Liu, W., Yuille, A. L. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3205-3214). 2019.

BabyX: Transferring 3D facial expressions from adults to children

Antonia Alomar
Universitat Pompeu
Fabra, Barcelona, Spain
antonia.alomar@upf.edu

Marius G. Linguraru
Children's National
Hospital & George
Washington University,
Washington, D.C., U.S.A.
mlingura@childrensnational.org

Araceli Morales
Universitat Pompeu
Fabra, Barcelona, Spain
mariadearaceli.morales@upf.edu

Gemma Piella
Universitat Pompeu
Fabra, Barcelona, Spain
gemma.piella@upf.edu

Antonio R. Porras
University of Colorado &
Children's Hospital
Colorado, Aurora, CO,
U.S.A.
antonio.porras@cuanschutz.edu

Federico Sukno
Universitat Pompeu
Fabra, Barcelona, Spain
federico.sukno@upf.edu

Abstract

Diagnosis of craniofacial conditions is shifting towards *pre-* and *peri-natal stages*, since early assessment has shown to be crucial for the effective treatment of functional and developmental aspects of children. 3D Morphable Models are a valuable tool for such evaluation. However, limited data availability on 3D newborn geometry, and highly variable imaging environments, challenge the construction of 3D baby face models. Our hypothesis is that constructing a bi-linear baby face model that allows identity and expression decoupling, enables to improve craniofacial and brain function assessments. Thus, given that adult and infants facial expression configurations are very similar and that 3D facial expressions in babies are difficult to be scanned in a controlled manner, we propose transferring the facial expressions from the available FaceWarehouse (FW) database to baby scans, to construct a baby-specific bi-linear expression model. First, we defined a spatial mapping between the BabyFM and the FW. Then, we propose an automatic neutralization to remove the expressions from the facial scans. Finally, we apply expression transfer to obtain a complete data tensor. We test the performance and generalization of the resulting bi-linear model with a test set. Results show that the obtained model allow us to successfully and realistically manipulate facial expressions of babies while keeping them decoupled from identity variations.

Keywords

Baby facial expressions, bi-linear model, automatic scan neutralization, face transfer.

1 INTRODUCTION

Craniofacial dysmorphology has been highlighted as an index of developmental disturbance at early stages of life [LMLP$^+$06; TPBL18; TPM$^+$19], encompassing a wide range of heterogeneous conditions associated with many genetic syndromes [HAK$^+$20]. Early recognition and assessment of craniofacial conditions are often crucial for the effective treatment of functional and developmental aspects of children [LMLP$^+$06]. For this reason, diagnosis is shifting towards *pre-* and *peri-natal stages*.

Normative population references are needed for craniofacial analysis and 3D Morphable Models (3DMM) have been proven as a valuable tool for their construction [KPB$^+$19]. Nonetheless, a crucial aspect to consider when using a 3DMM is that the demographics of the data used to train the model (e.g., ethnicity, gender and, especially important for our application, age) must match those of the target population. In addition to the limited availability of 3D newborn geometries to train such models, the main challenge is related to a highly variable imaging environment to obtain such geometries. Specifically, the variable baby facial expression during scan acquisition is one of the main factors affecting model quality and their neutralization is essential to build robust methods. Moreover, fetal facial expression is believed to be important to investigate the development of the fetal brain and the central nervous system [KHN$^+$13].

In this paper, we propose a method to decouple the baby identity from the expression on a non-controlled expression dataset. Using the BabyFM [MPT$^+$20], which was the first 3DMM constructed exclusively from ba-

https://www.doi.org/10.24132/CSRN.3201.14

bies and newborns, we augment the available 3D data by transferring the 46 FaceWarehouse (FW) facial expressions to the original 3D scans used to train the BabyFM. Our hypothesis is that adding variability and control over the expression makes the model more generalizable and reduces the error when performing the 2D-3D fitting from multiple images in which the baby changes his/her expression.

1.1 Facial 3D Morphable Models

3DMMs are a powerful tool exploited in a large variety of applications for face analysis [EST$^+$20], face recognition [HV13] and computer vision [MPS21], as well as for computer graphics [BRV$^+$18], animation [YF20] and even health [LHCZ21; KPB$^+$19].

The first 3DMM was presented by Blanz and Vetter in 1999 and was built from a limited number of 3D face scans, mostly in a neutral expression [BV99]. Since then, numerous approaches have been proposed to build 3DMMs. The simplest one consists in modeling the facial geometry variation by linear subspaces such as principal component analysis. However, these methods encode all geometric variations in the same subspace, and do not allow modeling different variations independently.

Multi-linear models were later presented to decouple different factors, such as identity and expression. The most widely used 3D facial expression model is the FaceWarehouse (FW) [CWZ$^+$14], built from a database of 150 individuals with 47 blendshapes corresponding to the 46 Action Units (AU) as described in the Facial Action Coding System (FACS) [EF02], plus the neutral blendshape. Vlasic et al. [VBPP05] presented two models: a bi-linear model trained with 15 subjects with the same ten facial expressions, and a tri-linear one containing 16 subjects with five visemes in five different expressions. Yin et al. [YWS$^+$] developed a 3D facial expression database that can be used to construct bi-linear models. It includes both 3D facial expression shapes and 2D facial textures of 100 subjects with seven universal expressions.

Unfortunately, multi-linear models require a complete data tensor, i.e. there must be a facial scan for each identity and expression pair. Therefore, multi-linear models are limited by data availability and they are not suitable when targeting datasets with sparse or nonuniform facial expression. We presented a method that enables data augmentation to construct a multi-linear model from a dataset that only contains a single scan per identity, with an unknown facial expression that, in most cases, is not neutral.

1.2 Face Transfer

The acquisition of time-varying 3D face scans has become an increasingly popular technology for transferring real facial human expressions to virtual avatars in movies and video games [CWLZ13; WBLP11; VBPP05]. In addition, this technology has also been used for face transfer and reenactment of RGB videos [TZS$^+$16; TZN$^+$15; RLMNA18] and for data augmentation to train 3DMMs and expression recognition algorithms [WBZB20; TMA19].

The core idea behind the facial expression transfer is that given two 3D scans of the same person with different expressions, we can locally transfer the expression to a different person using the displacement vector (or *deformation field*) from the reference expression to the desired one. The limitation of this procedure is that we need 3D meshes of the source and the target with the same expression (usually neutral) plus another 3D mesh with the target expression of the same identity as the source. Moreover, the target and the source require the same mesh triangulation to apply the deformation vector. When transferring expressions or deformations between two different databases with different triangulation, a mapping between both spaces is required [SP04; LBB$^+$17]. As we have mentioned, we cannot control newborns' facial expressions their expression while being scanned, which hampers the use of the above ideas.

1.3 Facial Expressions

Facial expressions are configurations of different small muscle (micromotor) movements in the face [Har16]. Ekman and Friesen's FACS was the first widely used and empirically validated approach to classifying a person's emotional state from their facial expressions [EO79].

There are variations of FACS, such as the Baby FACS, which code the facial expressions in infants. Both of them represent an exhaustive catalogue of possible movements of the human facial musculature [EF02; Ost06]. Despite the large number of different possible facial configurations, Ekman et al. [EF02] described a limited set of 46 AUs that are widely recognized. Any human expression can be characterized by a specific combination of AUs.

1.3.1 Facial Expression in Babies

Facial musculature is fully formed and functional at birth [EO79]. 2-3 weeks old neonates can imitate different actions such as mouth opening and tongue or lip protrusion. Despite the large debate about the similarity between early infant and adults emotions, there is agreement over the fact that the muscle movements and infant facial expressions configurations are very similar to the adult ones even at early stages [IHH87; OHN92; CSM93]. However, distinctive facial structures in babies should be considered.

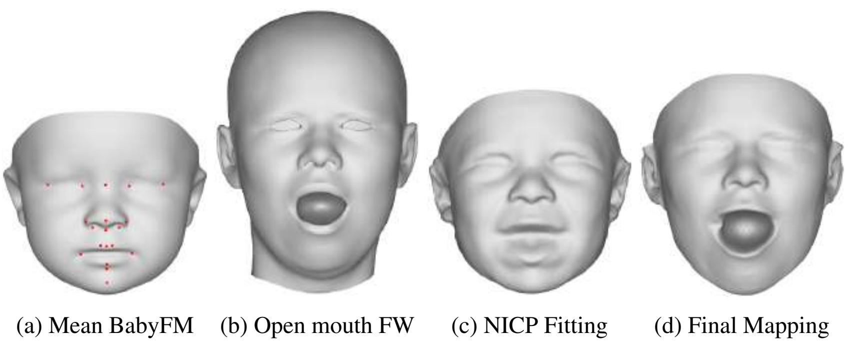


Figure 1: Mapping FW-BabyFM. (a) and (b) show the models average meshes. Whereas, (c) and (d) show the two different model fitting stages.

Given that adult and infants facial expression configurations are very similar, and that 3D facial expressions in babies are difficult to be scanned in a controlled manner, we propose transferring the facial expressions from the available 3D adult expression databases to baby scans. We base our work on the FW model, as is the most widely used model for face transfer and animation, and it can represent any combinations of facial muscle movements. Other existing databases provide overlapping facial movements that substantially reduce the expression space and model versatility.

2 BABY DATA

Our data consist on 115 3D photogrammetries of the baby head obtained at the Children's National Hospital in Washington D.C. The 3D scans have non-neutral expression and the surface that was not strictly face was removed [MPT+20].

3 METHODOLOGY

We obtained a data tensor with a complete set of expressions for all identities from a group of non-neutral newborn scans, to construct a bi-linear model. The spectral correspondence framework was used to establish the correspondences between the baby scans as detailed in Morales et al. [MPT+20]. Then, we implemented the following algorithm to obtain a complete data tensor from the 3D baby scans: 1) we defined a mapping between the BabyFM and the FW, 2) we trained 3D facial AUs classifiers to perform expression recognition; 3) we automatically neutralized the baby scan expressions; and 4) we apply expression transfer from FW to the neutralized baby scans. Each of this step is details in the following subsections.

3.1 Creating a complete data tensor

3.1.1 Mapping definition

We found a mapping transformation ($\mathscr{M}$) to represent surfaces from the FW with the BabyFM triangulation. Note that both models differ in the number of vertices (36K in BabyFM vs 11K in FW) and cover the face and head to a different extent. To facilitate calculating an accurate mapping ($\mathscr{M}$), we first triangulated the mouths from FW using a paraboloid-based approximation (see Figure 1b), closing the hole and making the registration more stable. Then, based on the procedure described in Dai et al. [DPSD20], we performed a multi-stage fitting that allowed overcoming the possible inaccuracies derived from the challenges of establishing correspondences between baby and adult faces given their anatomical differences. The multi-stage fitting can be divided in: i) a template adaptation, and ii) an Iterative Coherent Point Drift (ICPD), each one followed by a Laplace-Beltrami Regularized Projection (LBRP).

Using the average BabyFM ($\mu_{\mathbf{BB}}$) and FW mesh ($\mu_{\mathbf{FW}}$) with the mouth open as references, we first performed the template adaptation through a first rigid global alignment followed by a dynamic adaptation. The global alignment was based on the 23 anatomical landmarks of the BabyFM (see Figure 1a), whose corresponding locations were also manually annotated in $\mu_{\mathbf{FW}}$ using ParaView-5.4.1[1]. Then, a Non-Rigid Iterative Closest Point (NICP) algorithm with a non-linear optimization was used to refine the aligment, where $\mu_{\mathbf{BB}}$ was deformed to fit $\mu_{\mathbf{FW}}$. This was followed by a local regularization step using the LBRP.

The above steps yielded the baby-looking facial reconstruction shown in Figure 1c, due to the constraints imposed by the BabyFM. To improve the accuracy of our reconstruction of $\mu_{\mathbf{FW}}$, we displaced the BabyFM vertices to the nearest surface point (in barycentric coordinates) in the FW using ICPD, followed by consecutives LBRP with increasing strength.

The use of barycentric coordinates allows more accurate correspondences, not limited to common landmarking establishing point-to-point correspondences, and it also facilitates the mapping between surfaces of different numbers of vertices, as is our case. Using this representation, we represent the coordinates $\mathbf{p}_i$ of each

[1] https://www.paraview.org/

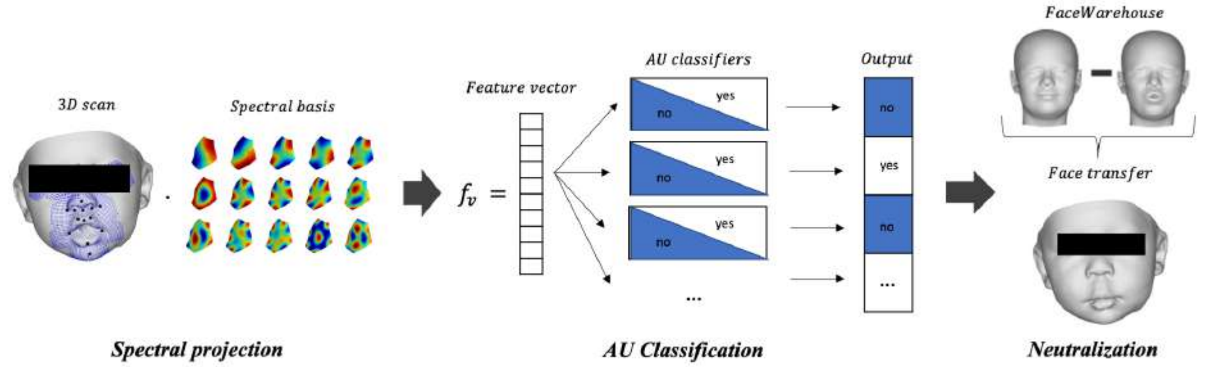

[redacted]
[redacted]

Figure 2: Neutralization pipeline. Considering a non-neutral 3D scan, each patch centered at the anatomical landmarks considered is projected in the spectral domain and concatenated to form the feature vector. This feature vector is fed into the AU classifiers and the output is used to synthesize the facial expression of the baby scan in the FW coordinates by combining the detected AUs ($\mathbf{V_{BB,s}}$). The last step is to apply face transfer to neutralize the scan using $\mathbf{V_{BB,s}}$.

vertex i of the FW as a function of the nearest BabyFM vertices as:

$$\mathbf{p}_i = \alpha_1 \mathbf{v_1} + \alpha_2 \mathbf{v_2} + \alpha_3 \mathbf{v_3} \qquad (1)$$

where $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3 \in \mathbb{R}^3$ are the triangle vertices in BabyFM coordinates and α_1, α_2, and $\alpha_3 \in \mathbb{R}$ are the barycentric coordinate parameters, such that $\alpha_1 + \alpha_2 + \alpha_3 = 1$.

Given $\mu_{\mathbf{BB}} \in \mathbb{R}^{3\times N}$ and $\mu_{\mathbf{FW}} \in \mathbb{R}^{3\times M}$, we defined a mapping $\mathscr{M}$ of dimension $M \times N$ to express $\mu_{\mathbf{FW}}$ in baricentric coordinates of $\mu_{\mathbf{BB}}$, where each column corresponds to the affine combination of $\mu_{\mathbf{FW}}$ vertices to define $\mathbf{p}_i \in \mu_{\mathbf{BB}}$. The values α of the triangle on which $\mathbf{p}_i$ lies are scalar values $\in [0,1]$ and the rest are zero.

3.1.2 3D Facial Expression Recognition

Before transferring a FW facial expression to a given baby scan, we first need to identify and remove any expression present in the baby scan, which we refer to as *neutralization*. We follow the method proposed by Derkach et al. [DS18] based on local shape spectral analysis to conduct automatic 3D facial expression recognition.

As the facial expressions are located in the frontal part of the face, we compute the spectral representation of local patches centered at the first 19 anatomical landmarks of the BabyFM (landmarks of the ears are not considered as do not add any information for expression recognition). In our implementation, we define these patches using a 5-ring neighbourhoods (see Figure 3). Given a landmark mesh patch $\mathscr{P} = (\mathbf{V}, \mathbf{E})$ with n vertices where $\mathbf{V}$ are the vertices and $\mathbf{E}$ the edges connections, we compute the graph Laplacian $\mathbf{L^G}$ as an $n \times n$ matrix defined by a discrete Schrödinger operator as:

$$\mathbf{L^G}_{ij} = \begin{cases} -1 & if \quad (i,j) \in \mathbf{E} \\ d_i & if \quad i = j \\ 0 & otherwise \end{cases} \qquad (2)$$

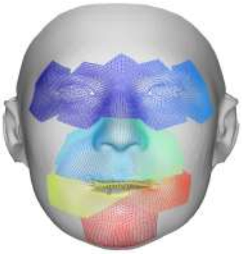


Figure 3: Local patches used for automatic AU recognition, centered at 19 anatomical facial landmarks, as defined in Morales et al. [MPT+20].

where d_i is the valence or degree of vertex i and $\mathbf{E}$ are the edges of the 1-ring neighborhood.

We performed eigen-decomposition of $\mathbf{L^G}$ to obtain a local spectral representation of the spatial information around every landmark. Then, we defined the τ-dimensional embedding $\Phi_\tau = [\phi_\mathbf{1} \phi_\mathbf{2} ... \phi_\tau]$ to represent the global patch structure. This embedding contained the eigenvectors ($\phi_\mathbf{1}, ... \phi_\tau$) associated with the smallest eigenvalues, which represent the low frequency information. Eliminating high frequencies adds robustness to local noise.

The mesh coordinates of each patch can be projected into the spectral domain as

$$\tilde{\mathbf{x}} = \Phi_\tau^T \mathbf{x} \qquad (3)$$

where $\tilde{\mathbf{x}}$ are the obtained spectral coefficients and $\mathbf{x}$ the patch coordinates.

We describe each mesh using a feature vector defined as the concatenation of the spectral coefficients obtained for its 19 local patches using a 50-dimensional embedding, which has shown to perform well for facial expression recognition [DS18]. Then, we use these embeddings to train the 46 binary support vector machine (SVM) classifiers using the LIBSVM [CL11].

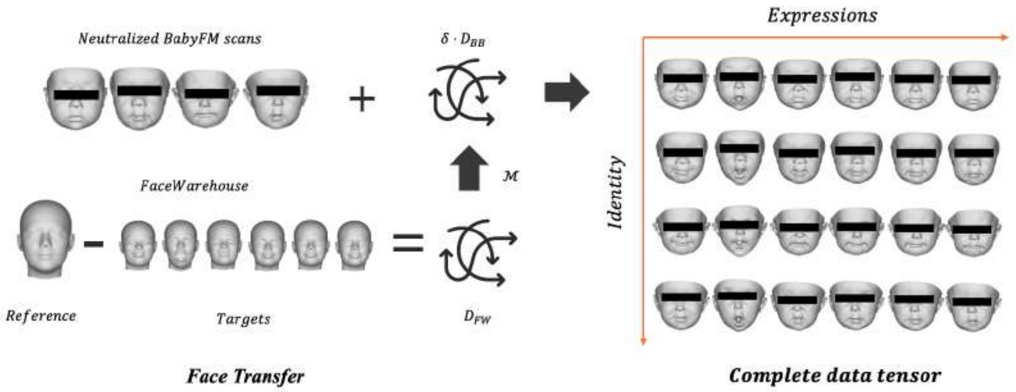


Figure 4: Completion of the data tensor. On the right it is illustrated the computation and transferring of the deformation fields ($\mathbf{D_{FW}}$) of 6 different facial expressions from the FW database to 4 neutralized babies using the mapping found between the FW and the BabyFM coordinates. And, on the right side it can be seen the completion of the data tensor after transferring the 6 different facial expressions (columns) to the 4 neutralized babies (rows).

3.1.3 Face transfer

Once we have computed the mapping ($\mathscr{M}$) between the FW and the BabyFM, we can express facial expression deformation vectors from the FW ($\mathbf{D_{FW}}$) under the BabyFM representation ($\mathbf{D_{BB}}$).

To transfer an expression e to a baby scan with expression s (i.e. with vertices $\mathbf{V_{BB,s}}$), we randomly select two scans from an individual of the FW database with expressions e and s, with vertices $\mathbf{V_{FW,e}}$ and $\mathbf{V_{FW,s}}$, respectively. The deformation vector in the FW triangulation ($\mathbf{D_{FW}}$) from s to e is computed as:

$$\mathbf{D_{FW}} = \mathbf{V_{FW,e}} - \mathbf{V_{FW,s}}. \tag{4}$$

To achieve expression transfer from the baby mesh $\mathbf{V_{BB,s}}$ to $\mathbf{V_{BB,s}}$, we perform the following operations:

$$\mathbf{V_{BB,e}} = \mathbf{V_{BB,s}} + \delta \mathbf{D_{BB}} \tag{5}$$

$$\mathbf{D_{BB}} = \mathbf{A_{in}} \cdot \mathscr{M} \cdot \mathbf{D_{FW}} \tag{6}$$

where δ is the weight (strength) of the deformation vector and $\mathbf{D_{BB}}$ is the equivalent to expression deformation $\mathbf{D_{FW}}$ expressed in the coordinates of the BabyFM.

To this end, we need to firstly map the deformation vector $\mathbf{D_{FW}}$ to the BabyFM coordinates by means of the mapping $\mathscr{M}$ (Section 3.1.1) and then align each deformation vector to adapt to the local geometry of the target individual. The latter is implemented as a set of local similarity transformations $\mathbf{A_{in}}$ that are computed by Procrustes alignment of the local neighborhood of each vertex in $\mathbf{V_{BB,s}}$ with respect to $\mathscr{M} \cdot \mathbf{V_{FW,s}}$.

To avoid that the expression transfer depends too much on a given individual, we repeat the process for $\kappa = 40$ randomly selected individuals from the FW database. The final result, obtained by averaging the obtained $\mathbf{V_{BB,e}}$, is added to the data tensor. A visual illustration of this procedure can be seen in Figure 4.

Neutralitzation

The 46 trained classifiers are applied to all the 3D baby scans, obtaining an automatic estimate of all the AUs present in each mesh. To neutralize a given scan $\mathbf{V_{BB,s}}$, we synthesize the facial expression of the baby scan in the FW coordinates ($\mathbf{V_{FW,s}}$) by combining the detected AUs. Then, face transfer is applied (Equations 4 - 5), setting e = neutral, to select $\mathbf{V_{FW,e}}$. An illustration of this procedure can be seen in Figure 2.

Baby Tensor Expressions

For all the 3D scans that we were able to automatically neutralize (see Section 4.3), we transferred all the 46 AUs available in the FW database, the six universal expressions, the compound expressions [Shi15] and some characteristic baby facial expressions (such as crying with closed eyes). The detailed list of facial expressions considered to form the baby data tensor can be found in Table 2 of the Supplementary Material.

We always started from the neutralized scans, i.e. $\mathbf{V_{BB,s}}$ with s = neutral, while the target expressions e were those mentioned above, which were transferred to each available baby subject.

3.2 Bi-linear baby model

After performing face transference we achieved a data tensor $\mathbf{T} \in \mathbb{R}^{3n_{verts} \times n_{id} \times n_{ex}}$ where n_{verts} is the number of vertices of the BabyFM triangulation, n_{id} is the number of baby identities considered after automatic neutralization and n_{ex} is the number of facial expressions that we choose for our expression space. The obtained data tensor is a mode-three tensor with dimension $93098 \times 95 \times 76$.

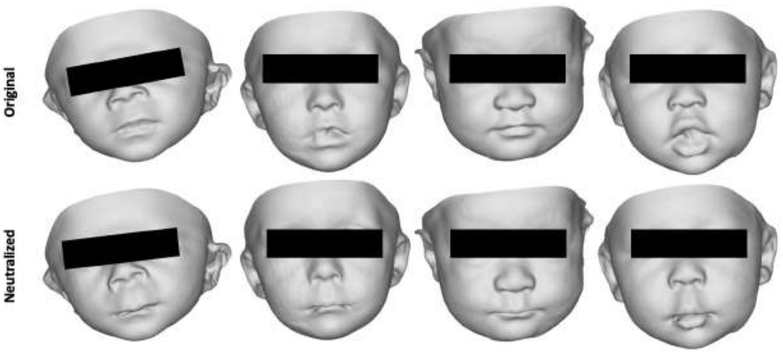


Figure 5: Automatic neutralization examples. The first row shows four different original 3D scans and the second row displays their respective neutralized scan achieved after the proposed automatic neutralization.

Higher-Order Singular Value Decomposition (HOSVD) is used to decompose the data tensor, obtaining the following representation:

$$\mathbf{T} = \mathbf{S} \otimes \mathbf{U_{verts}} \otimes \mathbf{U_{id}} \otimes \mathbf{U_{ex}} \tag{7}$$

where S is the core tensor and the U matrices are the orthogonal bases for the η different subspaces of the mode-η tensor.

The decomposition enables us to model and decouple identity and expression. Thus, a 3D face can be described as:

$$\mathbf{x} = \mathbf{C} \otimes \mathbf{w_{id}} \otimes \mathbf{w_{ex}} + \bar{\mathbf{x}} \tag{8}$$

where $\mathbf{C} = \mathbf{S} \otimes U_{verts}$, $\bar{\mathbf{x}}$ is the mean of the bi-linear, $\mathbf{w_{id}}$ and $\mathbf{w_{ex}}$ are the identity and expression parameters.

3.2.1 Iterative Fitting

To evaluate the constructed bi-linear model, we reconstruct a collection of the 3D scans that are independent to the training scans. A total of 20 3D scans were used to test the capacity of the bi-linear model to represent babies that were not included in the training data.

First we perform a Procrustes alignment between the model and the 3D scan vertices ($\mathbf{x}$). Afterwards, a non-linear optimization with regularization is used. In each iteration, we first estimate the identity parameters and secondly the expression parameters. To initialize the fitting, expression parameters are set to those corresponding to the neutral expression and the vertices to the mean of the model. We truncate the model and kept $d_{id} = 90$ and $d_{ex} = 50$ dimensions.

The shape parameters of the model are obtained using the following non-linear optimization:

$$E(\mathbf{w}) = \beta_1 E_{verts}(\mathbf{w}) + \beta_2 E_{Prior}(\mathbf{w}) \tag{9}$$

where β_1 and β_2 are the corresponding weights of each error term ($\beta_1 + \beta_2 = 1$). E_{verts} refers to the reconstruction error of the vertices of the mesh and E_{Prior} to the error due to the statistical prior that constrains the solution to lie within a hyper-ellipsoid estimated from the training set (i.e. multi-variate Gaussian assumption). Each error term depends on the parameters that we are estimating ($\mathbf{w}$). Thus, we have:

$$\mathbf{E}_{verts}(\mathbf{w_{id}}) = ||(C \otimes \mathbf{w_{id}} \otimes \mathbf{w_{ex}}) - (\bar{\mathbf{x}} - \mathbf{x})||^2 \tag{10}$$

$$\mathbf{E}_{verts}(\mathbf{w_{ex}}) = ||(C \otimes \mathbf{w_{id}} \otimes \mathbf{w_{ex}}) - (\bar{\mathbf{x}} - \mathbf{x})||^2 \tag{11}$$

$$\mathbf{E}_{Prior}(\mathbf{w_{id}}) = (\frac{\mathbf{w_{id}}}{\sqrt{\lambda_{\mathbf{id}}}})^2 \tag{12}$$

and

$$\mathbf{E}_{Prior}(\mathbf{w_{ex}}) = (\frac{\mathbf{w_{ex}}}{\sqrt{\lambda_{\mathbf{ex}}}})^2 \tag{13}$$

where $\mathbf{x}$ are the 3D scan vertices, $\lambda_{\mathbf{id}}$ and $\lambda_{\mathbf{ex}}$ are the identity and expression eigenvalues of the corresponding sub-spaces.

4 RESULTS

In this section, we show the results of the above proposed methodology and also, perform different experiments to test the resulting bi-linear model.

4.1 Mapping

The results of the multi-stage procedure implemented to find an accurate mapping between the BabyFM and the FW are shown in Figure 1. In (c) it can be seen the template adaptation result that is still far from the FW mesh (b). The final mesh obtained after NICP fitting applying the mapping computed using the barycentric coordinate representation is shown in (d). In the latter it can be appreciated the similarity between the FW original mesh (b) and the one obtained in the BabyFM coordinates using the computed mapping. Thus, it can be said that an accurate mapping was found.

4.2 Automatic AU classification

The first 120 identities from the FW database with their 46 respective facial expressions were mapped to the BabyFM coordinates and then, used for training the AU classifiers. The remaining identities were used as test set to evaluate the performance of the classifiers.

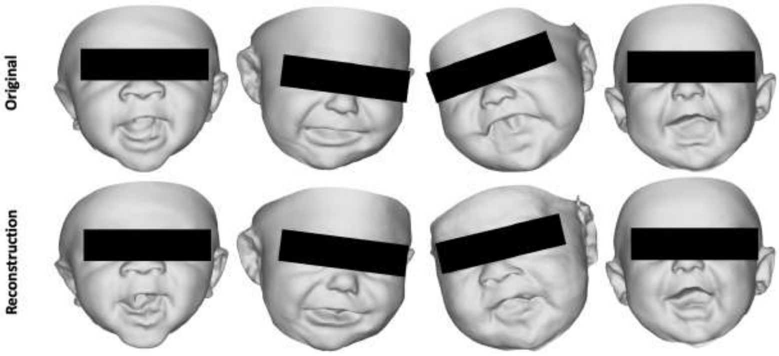


Figure 6: Test Fitting. The first row shows four different babies not included in the training of the model, and the second row displays their respective reconstruction obtained after fitting the bi-linear model to the original scans.

A total of 46 AU classifiers were trained, obtaining a mean accuracy of 99.62 ± 0.97% in the test set. The classifiers with less accuracy were the ones corresponding to the Upper Left and Right Lid Raiser (AU 13 and 14 in the FW database) with an accuracy of 95.67% and 96.31%, respectively. An accuracy of 100% in the test set was achieved in 30 out of the 46 trained classifiers. The accuracy of each classifier can be found in Table 1 of the Supplementary Material.

4.3 Neutralization

The evaluation of the 3D scan neutralization was done by visual inspection, where 3 independent observers classified the obtained scans as neutral or not neutral. An agreement of 94.52% between the 3 independent observers on the classification of the neutralized scans was archived. As a results 95 scans out of 115 were considered correctly neutralized by the proposed method and were thus used to construct our data tensor.

A few examples of the results obtained using the proposed automatic neutralization procedure are shown in Figure 5. The first row shows four different original 3D scans, and the second row displays their respective neutralized scan achieved. Note that using the pipeline proposed in Figure 2 we are able to obtain acceptable neutralizations. The most distinguishable changes between the original and the neutralized scan are located in the mouth and cheeks, e.g the closing of the mouth.

4.4 Bi-lienar Model

Once we obtained the complete data tensor, we applied HOSVD to decompose the tensor and decouple the identity from the expression in the baby facial geometry. To test the performance of the constructed bi-linear model the following experiments were performed.

4.4.1 Synthetization of Baby Identities

Using the statistics encoded in the identity subspace of the bi-linear model, the identities with neutral faces of Figure 7 were randomly synthesized, while the expression parameters were set to the neutral face parameters of the model. It can be observed that the constructed bi-linear model creates new realistic and plausible identities.

To test the generalization of the model, we change the expression of the randomly synthesized identities to evaluate if the model succeed in maintaining the identity and changing the expression to plausible shapes while avoiding distortions. As shown in Figure 7, the obtained changes of expressions are a good representation of plausible facial expressions for each identity. Moreover, the expressions can be easily identified.

4.4.2 Interpolation in the expression space

To test if the constructed model is able to represent expressions that were not part of the training, and if the expression space is stable to changes, we interpolate the expression parameters between different expressions present in the bi-linear model. Figure 9 shows the interpolation considering different strengths (from 0 to 1) between happily surprised and happily disgusted; and between happy and surprise. In the latter, it can be seen how the baby progressively and realistically raises the eyebrows and opens the mouth.

4.4.3 Reconstruction error in the test set

Figure 6 shows some qualitative examples of the reconstructions obtained by fitting the bi-linear model to a collection of 3D scans that are independent to the training scans. The main discrepancies between the reconstructed and the original scans can be located in the region corresponding to the mouth, which usually is the noisiest region in the 3D scans. The rest of the facial shape is reconstructed with high accuracy.

A quantitative analysis was performed by computing the error between the reconstruction achieved and the original 3D scan. A mean reconstruction error of 0.928 ± 0.142 mm was achieved. In Figure 8, we show the mean error per vertex across the 20 scans from the test

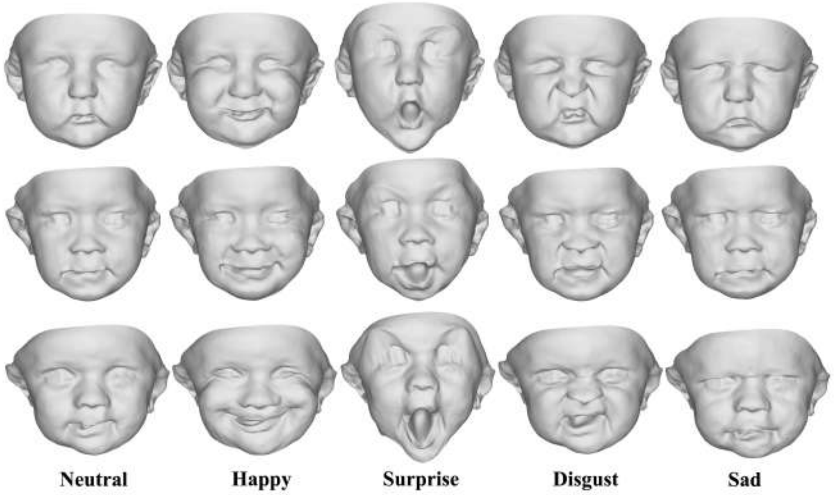


Figure 7: Generating synthetic identities and changing their expression. Each row correspond to a randomly synthesized identity ($\mathbf{w_{id}}$), while each column correspond a different facial expression using the learned expression parameters of the bi-linear model ($\mathbf{w_{ex}}$).

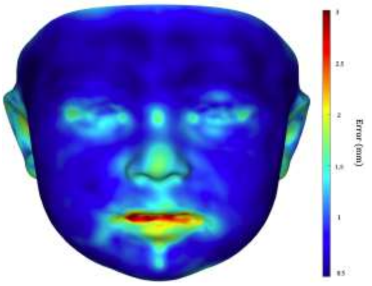


Figure 8: Test reconstruction error. It shows the mean error per vertex across the 20 scans from the test set fitting.

set. It can be observed that the highest error (colored in red) is obtained in the region of the mouth, which is in agreement with the qualitative analysis.

4.4.4 Data Augmentation

As the reconstructions obtained have a high accuracy, we can use our approach for data augmentation to improve the created bi-linear model. Once we have the bi-linear model fitted to a 3D scan, we have its identity and expression parameters. Thus, if we maintain the identity parameters ($\mathbf{w_{id}}$) and change the expression parameters ($\mathbf{w_{id}}$) to the expressions that conform the data of the model, we obtain new synthetic expressions on identities different from those used to create the model. The advantage of this procedure is that no face transfer is needed to change the expression, only the corresponding expression parameters obtained in the model ($\mathbf{U_{ex}}$). So, in this way we can easily add more variability to the model just adding more real identities to the data tensor and recomputing the bi-linear model. Figure 2 in the Supplementary Materials shows the resulting changes of expression. Note that plausible expressions are obtained and the identity of the subjects is kept.

5 CONCLUSIONS

In this paper, we presented a methodology to achieve a complete baby expression data tensor transferring the facial expression from the FW adult database to the BabyFM training set.

The accurate mapping found between the FW and the BabyFM enable us to transfer facial expressions to the babies scans and to enrich the baby dataset allowing the construction of a complete data tensor. Note that thanks to the AU classifiers, we are able to construct a synthetic expression with the FW coordinates that imitates the baby expression in the 3D scan and then apply face transfer from the FW to the baby scan to neutralize the expression.

Moreover, the different experiments performed in the obtained bi-linear model show that the obtained model is a good estimation of the facial shape and expression of the babies space. It allows us to successfully decouple identity and expression.

As future work, we will use the constructed bi-linear model to reconstruct the 3D-2D reconstruction from multiple 2D images taken from different views. Also, the proposed methodology can be extended to perform data augmentation of existing non-neutral datasets, adding facial expressions to each subject.

6 ACKNOWLEDGMENTS

This work is partly supported the eSCANFace project (PID2020-114083GB-I00) funded by the Spanish Ministry of Science and Innovation, and the Eunice

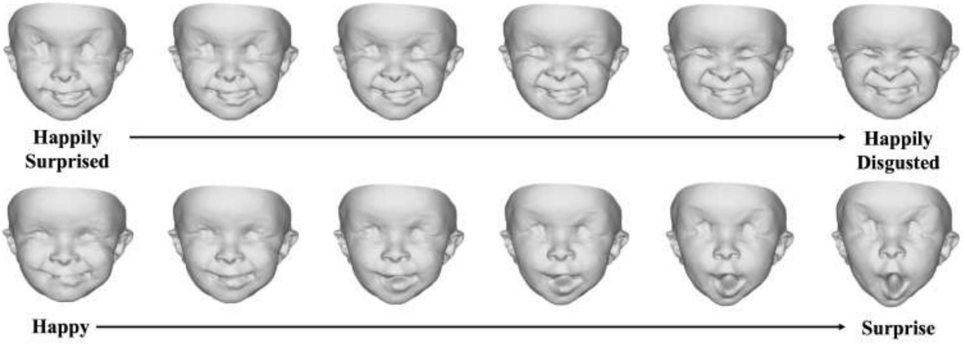


Figure 9: Interpolations between different facial expressions (w_{ex}).

Kennedy Shriver National Institute of Child Health and Human Development grant R42HD081712. A. Alomar was supported by AGAUR under the FI scholarship and G. Piella was supported by ICREA under the ICREA Academia programme.

REFERENCES

[BRV+18] Booth, J., Roussos, A., Ververas, E., et al. 3D reconstruction of In-the-Wild faces in images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2638–2652, 2018.

[BV99] Blanz, V. and Vetter, T. A morphable model for the synthesis of 3D faces. *SIGGRAPH '99: Proceedings of the 26th annual conference.*, pages 187–194, 1999.

[CL11] Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[CSM93] Camras, L. A., Sullivan, J., and Michel, G. Do infants express discrete emotions? Adult judgments of facial, vocal, and body actions. *Journal of Nonverbal Behavior*, 17:171–186, 1993.

[CWLZ13] Cao, C., Weng, Y., Lin, S., and Zhou, K. 3D shape regression for real-time facial animation. *ACM Transactions on Graphics*, 32:1–10, 2013.

[CWZ+14] Cao, C., Weng, Y., Zhou, S., et al. FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE Transactions on Visualization and Computer Graphics*, 20:413–425, 2014.

[DPSD20] Dai, H., Pears, N., Smith, W., and Duncan, C. Statistical Modeling of Craniofacial Shape and Texture. *International Journal of Computer Vision*, 128(2):547–571, 2020.

[DS18] Derkach, D. and Sukno, F. M. Automatic local shape spectrum analysis for 3D facial expression recognition. *Image and Vision Computing*, 79:86–98, 2018.

[EF02] Ekman, P. and Friesen, W. V. Facial Action Coding System (FACS): A technique for the measurement of facial action. *Palo Alto CA Consulting*, 3, 2002.

[EO79] Ekman, P. and Oster, H. Facial Expressions of Emotion. *Annual Review of Psychology*, 30:527–554, 1979.

[EST+20] Egger, B., Smith, W. A. P., Tewari, A., et al. 3d morphable face modelsâpast, present, and future. *ACM Transactions on Graphics*, 39:1–38, 2020.

[HAK+20] Hallgrimsson, B., Aponte, J. D., Katz, D. C., et al. Automated syndrome diagnosis by three-dimensional facial imaging. *Genetics in Medicine*, 22:1682–1693, 2020.

[Har16] Harley, J. M. Measuring emotions. *Emotions, Technology, Design, and Learning*, pages 89–114, 2016.

[HV13] Haar, F. B. and Veltkamp, R. 3D Morphable Models for Face Surface Analysis and Recognition. *3D Face Modeling, Analysis and Recognition*, pages 119–147, 2013.

[IHH87] Izard, C. E., Hembree, E. A., and Huebner, R. R. Infants' emotion expressions to acute pain: Developmental change and stability of individual differences. *Developmental Psychology*, 23:105–113, 1987.

[KHN+13] Kanenishi, K., Hanaoka, U., Noguchi, J., et al. 4D ultrasound evaluation of fetal facial expressions during the latter stages of the second trimester. *International Journal of Gynecology Obstetrics*, 121:257–260, 2013.

[KPB$^+$19] Knoops, P. G. M., Papaioannou, A., Borghi, A., et al. A machine learning framework for automated diagnosis and computer-assisted planning in plastic and reconstructive surgery. *Scientific Reports*, 9:13597, 2019.

[LBB$^+$17] Li, T., Bolkart, T., Black, M. J., et al. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics*, 36:1–17, 2017.

[LHCZ21] Lan, K.-C., Hu, M.-C., Chen, Y.-Z., and Zhang, J.-X. The Application of 3D Morphable Model (3DMM) for Real-Time Visualization of Acupoints on a Smartphone. *IEEE Sensors Journal*, 21:3289–3300, 2021.

[LMLP$^+$06] Learned-Miller, E., Lu, Q., Paisley, A., et al. Detecting Acromegaly: Screening for disease with a Morphable Model. 4191 LNCS - II:495–503, 2006.

[MPS21] Morales, A., Piella, G., and Sukno, F. M. Survey on 3D face reconstruction from uncalibrated images. *Computer Science Review*, 40, 2021.

[MPT$^+$20] Morales, A., Porras, A. R., Tu, L., et al. Spectral Correspondence Framework for Building a 3D Baby Face Model. pages 708–715. IEEE, 2020.

[OHN92] Oster, H., Hegley, D., and Nagel, L. Adult judgments and fine-grained analysis of infant facial expressions: Testing the validity of a priori coding formulas. *Developmental Psychology*, 28:1115–1131, 1992.

[Ost06] Oster, H. Baby FACS: Facial Action Coding System for infants and young children. . *Unpublished monograph and coding manual. New York University.*, 2006.

[RLMNA18] Rotger, G., Lumbreras, F., Moreno-Noguer, F., and Agudo, A. 2D-to-3D Facial Expression Transfer. pages 2008–2013. IEEE, 2018.

[Shi15] Compound facial expressions of emotion: from basic research to clinical applications. *Dialogues in Clinical Neuroscience*, 17:443–455, 2015.

[SP04] Sumner, R. W. and Popović, J. Deformation transfer for triangle meshes. *ACM Transactions on Graphics*, 23:399–405, 2004.

[TMA19] Trimech, I. H., Maalej, A., and Amara, N. E. B. Data Augmentation using non-rigid CPD Registration for 3D Facial Expression Recognition. pages 164–169. IEEE, 2019.

[TPBL18] Tu, L., Porras, A. R., Boyle, A., and Linguraru, M. G. Analysis of 3D Facial Dysmorphology in Genetic Syndromes from Unconstrained 2D Photographs. pages 347–355, 2018.

[TPM$^+$19] Tu, L., Porras, A. R., Morales, A., et al. Three-dimensional face reconstruction from uncalibrated photographs: Application to early detection of genetic syndromes. *Springer International Publishing*, pages 182–189, 2019.

[TZN$^+$15] Thies, J., Zollhöfer, M., Nießner, M., et al. Real-time expression transfer for facial reenactment. *ACM Transactions on Graphics*, 34:1–14, 11 2015.

[TZS$^+$16] Thies, J., Zollhofer, M., Stamminger, M., et al. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2387–2395, 2016.

[VBPP05] Vlasic, D., Brand, M., Pfister, H., and PopoviÄ, J. Face transfer with multilinear models. *ACM Transactions on Graphics*, 24:426–433, 2005.

[WBLP11] Weise, T., Bouaziz, S., Li, H., and Pauly, M. Realtime performance-based facial animation. *ACM Transactions on Graphics*, 30:1–10, 2011.

[WBZB20] Wang, M., Bradley, D., Zafeiriou, S., and Beeler, T. Facial Expression Synthesis using a Global−Local Multilinear Framework. *Computer Graphics Forum*, 39:235–245, 2020.

[YF20] Ye, D. and Fuh, C.-S. 3D Morphable Face Model for Face Animation. *International Journal of Image and Graphics*, 20:2050003, 2020.

[YWS$^+$] Yin, L., Wei, X., Sun, Y., et al. A 3D Facial Expression Database For Facial Behavior Research. pages 211–216. IEEE.

ContourNet : a new deep convolutional neural network for 2D contour classification

Mhedhbi Makrem
CRISTAL LABORATORY
National School of Computer Sciences
Tunisia (2010), Manouba, Manouba
Makrem.mhedhbi@ensi-uma.tn

Mhiri Slim
CRISTAL LABORATORY
National School of Computer Sciences
Tunisia (2010), Manouba, Manouba
Slim.mhiri@ensi-uma.tn

Ghorbel Faouzi
CRISTAL LABORATORY
National School of Computer Sciences
Tunisia (2010), Manouba, Manouba
Faouzi.ghorbel@ensi-uma.tn

ABSTRACT

In this paper, we present a new deep convolutional neural network to classify 2d contours, described by a sequence of points coordinates representing the boundary of objects. Several works dealt with this subject, even those using learning, but few works use deep learning. This is due to the fact that contours data are very narrow and inappropriate for convolution. To enrich this representation, we use curve evolution and consider simultaneously a multi-scale representation of a contour. Associated with coordinates, curvature estimated at each point is the most used descriptor who can help distinguishing objects. Despite deficiency of large 2d contour datasets, required for a convergent learning, the use of several additional techniques, such as data augmentation, lead to results outperforming the state of the art. We train ContourNet on MPEG-7 database CE-1 part B, witch achieves 100% for Top-1 accuracy rate on MPEG-7 test set, and 91.78% on Kimia216 dataset.

Keywords

Contour description, Shape classification, CNN, Deep learning, Data augmentation, curve evolution.

1. INTRODUCTION

Classification and clustering of planar shapes described by their contour are of great importance for automation of several tasks such as the manufactory control (fault detection, automatic assembly), text recognition (sorting postal, RADAR control), security (digital fingerprints, facial, retinal)…. Given a set of planar shapes such as MPEG-7, KIMIA, ANIMAL bases, it is desirable to divide them into homogeneous groups sharing common characteristics, in order to facilitate the retrieval of images with similar content. When the number of groups is not known, it is a clustering, but when the number of groups is known, it is a classification. In both cases, it will be necessary to design discriminating descriptors making it possible to assign the form to a specific group in the first case, and to assign a label to a form in the second one. Several criteria should be verified by these descriptors, namely, invariance according to geometric transformations (translation, rotation and scale), robustness with respect to noise, occlusion and field of view. These descriptors must also be quantified and be part of a metric space, allowing measurement of similarity between shapes. Approaches cited in literature can be divided into main categories: category of approaches dealing with a single scale of a contour, and category of approaches using a multi-scale representation of contours.

In the first category, we can mention works like shape context [Bel02], curve edit distance [Seb03], the phase of Fourier descriptors [Bar05], the inner distance [Lin07], etc. With shape context [Bel02], authors introduce a new shape descriptor and a similarity distance measure to evaluate likeness between shapes. At each point of the shape (reference point), a local descriptor is created, based on the distribution of all remaining points, with regards to reference point. All obtained vectors are then embedded in a log-polar space, divided in many bins, according to length, and angle between a vector and a reference direction. This resulting histogram is called shape context descriptor, and used to evaluate correspondence between two points on two different shapes. Then, authors estimate the transformation that best align the two shapes. Distance between two

https://www.doi.org/10.24132/CSRN.3201.15

shapes is computed as the sum of matching errors between all matched points, added to the magnitude of aligning transformation. In [Lin07], Ling use also shape context but replace Euclidean distance by geodesic distance under some hypothesis to reduce computing complexity. This assumptions allow method to gain about 9% in accuracy rate, compared to original shape context. In [Seb03], Curve edit distance is used to estimate similarity of shapes. Based on curve length, arc-length parameterization and curvature, authors look for the function, from functions space, that minimize required energy to align curves. This function must satisfy the additivity property so that global alignment is the sum of many local alignments. Bartolini [Bar05] use phase of Fourier descriptors with dynamic time warping to compute similarity between shapes. This contribution focuses on importance of information contained in phase of Fourier coefficients, often neglected in favor of amplitude information, to ensure rotation invariance of descriptor. Using phase of Fourier descriptor makes inappropriate the use of Euclidean distance between shapes, thus dynamic time warping is well suited to measure similarity between shapes, as it allows matching of elastic deformations of a part of shapes. Best performances achieved for this category of approaches are those of shape context with inner distance [Lin07], with error rate of 14.60% for MPEG-7 dataset, and 2.63 for Kimia216 dataset.

Among approaches from multi-scale category, we can mention visual parts [Lat00], curvature scale space and its derivatives [6,7,8], beam angle statistics [Ari03], convexity-concavity multi scale representation [Ada04], triangle area representation [Ala07], Eigen and Fisher barycenter [Tho09], etc. In visual parts contribution [Lat00], Latecki aim to measure similarity between 2d shapes, by matching significant parts of shapes instead of matching the whole shapes. A shape is considered as a union of several concave or convex sub parts. The use of digital curve evolution is performed by substituting two consecutive line segments with a single line segment, joining the endpoints of initial segments. This contour simplification is done to eliminate noise digitization and segmentation errors. Then, each shape is represented by a tangent function, which is a step function. Similarity measure is deduced by computing area difference between tangent function of two shapes, after aligning their functions. Another contribution, part of current category of approaches, is Curvature Scale Space CSS [Mok03]. In this contribution, authors consider the normalized arc-length parameterization of the initial contour, with several variants extracted using curve evolution process. It consists of smoothing original contour shape by Gaussian functions with progressive σ parameter. These successive convolutions eliminate gradually points with high absolute value of curvature estimated, until reaching a convex shape (ellipse or circle in most cases). Shrinkage of evolved contours caused by curve evolution process is compensated with a motion vector to obtain evolved curves with same length as the original contour. CSS image descriptor is then obtained as solution of equation $\kappa(\mu,\sigma) = 0$, κ is curvature value, σ is the curve evolution parameter, and μ is the point coordinate. To compute similarity between two shapes, authors begin by shifting one shape until the two maximum of curvature coincides, and retain the sum of Euclidean distances between matched maxima. Another contribution in this category is the beam angle statistics BAS [Ari03]. In this work, a BAS descriptor is computed as follow: from each boundary point on the shape, a set of beams is considered, linking the reference point to all remaining points on the shape. Angles between each pair of beams help extracting the topological structure of contour. Use of multi-scale information in this context is realized by considering multiple levels of neighbors with a function called K-neighborhood. This leads to K-curvature function, regarded as a random variable. The BAS descriptor is a vector of third order statistical moments, which is invariant to translation, rotation, and scale and insensitive to distortions. In [Ada04], Adamek use curve evolution to generate a multi-scale representation of a contour. An estimation of convexity/concavity is done on each point of contour on original shape and all of its evolved versions, and result is a two-dimensional matrix descriptor (MCC). Columns of this matrix represents contour points while rows represents levels of evolution: σ. A dynamic time warping is then used to find the optimal global alignment between contours, and measure distance between corresponding shapes. In [Ala07], inspired by works on 3D shapes, Alajlan al introduce a new descriptor, called TAR, to characterize 2D shapes. Unlike previous works, where triangle area was normalized by global signature, alajlan and al locally normalize signature by dividing it by length of hypotenuse triangle. By using multiple neighborhood scales, TAR descriptor highlights both local and global details, making thus higher the discrimination power of shapes. Inspired by DTW, authors use a similar technique, called Dynamic Space Warping to match starting points of two shapes, before measuring similarity. To emphasize intuition behind matching, distance measurement is divided by a complexity term, which is the mean of absolute differences between highest and lowest signatures, through all points and all scales. Considering global features such as circularity, eccentricity and aspect ratio to increase discrimination shapes. In [Tho09], authors improve works done in [Seb03], by using barycentric points coordinates, instead of original points coordinates of the boundary shape, to generate several shapes of the contour: making a multi-scale representation. To ensure invariance to reflection and starting point, a phase normalization of descriptor is realized on spectral domain, giving a matrix N*M, where N represent points number and M represent scales number. Then, a linear discriminant analysis is processed on Fisher matrix to reach final descriptor.

The remainder of this paper is organized as follow: in Section II, we introduce convolutional neural networks architecture, their components, and some of the most famous ones in literature, mainly works dealing with contours. In Section III, we describe datasets used in this work, and how we preprocessed their contents. Section IV contains proposed architecture, hyper parameters settings and details of learning. Results and discussion will be in Section V and VI, and we conclude with some future works in Section VII.

2. RELATED WORKS

Since 2012, and with the advent of deep learning, as well as the impressive results it achieves, we thought about applying these solutions to the classification of plane contours. Convolutional neural networks have proved their capability to solve complex tasks such as segmentation, detection and clustering and labeling. A CNN is an artificial neural network with several hidden layers. To overcome the huge number of parameters included, and who increases dramatically each time a layer is added, convolution has been used. Unlike fully connected layers, where each neuron in layer j is connected to all neurons in the previous layer j-1, convolutional layers take the form of a stack of small-scale filters. Therefore, the input of a neuron is no longer equal to the weighted sum of all neurons in the previous layer, but it is equal to the weighted sum of a narrow neighborhood surrounding the affected neuron. In addition, the sharing of parameters results in a modest number of them. As cited above, these models can fit linear functions with millions of parameters, but it remains inefficient when applied to complex data. Therefore, non-linearity was introduced to expand functions space covered by CNN approximations. Moreover, it makes sense to go through depth. From layer to layer, CNNs learn more and more complex discriminant features, leading to better results in classification. Activations functions used to introduce non linearity in CNNs are of two types, saturating functions and non-saturating functions. The former translate neuron's information inside a bounded interval and the latter inside unbounded one. To further reduce number of parameters, a pooling operation is convenient. It replaces each bloc of neurons with fixed size, by one neuron containing value extracted from the others. Applied inside a convolutional layer, it reduces output parameters number by a scale factor of four, at least. The first deep architecture proposed was AlexNet [Kri12]: an eight-layer neural network alternating convolution, pooling, and fully connected layers. AlexNet was used to label over than one million marked images and reach better performances of the time. Many improvements of accuracy were proposed within ZFNet [Zei14], VGG [Sim14], GoogLeNet [Sze15], etc.

Using CNN to classify 2D forms datasets, such as MPEG-7, Animal or Leaf, focusing on boundary information, and ignoring color and texture information, was first introduced in [Ata16]. This work uses binary images to learn BSCNN, composed of three convolutional layers and one fully connected layer, leading to a Softmax layer for final classification. Many data augmentation ways were used to improve its accuracy rate, which achieve on MPEG-7 dataset, 98.99 on TOP-1 metric, and 99.76 on TOP-5 metric.

In [Zha21], another architecture of CNN was introduced and called SCN, aiming to be more general by classifying forms from different datasets. SCN is composed of four convolutional layers, and a fully connected layer. After the third convolutional layer, one de-convolution layer was inserted. To overcome changes of data distribution within network, and speed up learning, a batch normalization layers were added to SCN. To increase data amount and improve learning process, data augmentation was used. Accuracy rate achieved by SCN, according to TOP-1 metric is 90.99 % on MPEG-7 dataset.

The first CNN used to classify boundary data shapes [Dro20] is called ContourCNN. Authors consider both Cartesian and polar representations for training their system. They combined circular convolution layers and priority pooling layers with two dimension space representation. Circular convolution help system to highlight circularity of contour points regardless of abstract representation of these points. Priority pooling layers do not remove points in a regular manner, but iterates on them until having a fixed size. ContourCNN is composed of three circular convolution layers, three priority pooling layers, and one global average pooling layers, connected to two fully connected layers used for classification. Because of poorness of datasets like MPEG-7 and Animal, authors have tested ContourCNN on EMNIST dataset, and achieve an accuracy rate less than 97%. However, considerations made by authors does not seem to be contribution in contour classification for these reasons: 1) Circular convolution layer can simply be replaced by padding, 2) priority pooling do not make improvements with contour data since quantity of data is weak, 3) benchmark used to test robustness is EMNIST dataset, and we do not know performances on famous datasets, such as MPEG-7, Kimia, etc.

3. DATASET, COORDINATE SYSTEM AND CNN

In this paper, we aim to design a CNN to classify MPEG-7 CE-SHAPE-1 part B objects. This dataset is one of the most used benchmarks to compare performances of classification techniques. It contains images of seventy class of shapes, each class has twenty different objects, resulting in 1400 images. An overview of all these classes, and a sample of objects of some classes are illustrated in figures Fig.1 and Fig.2.

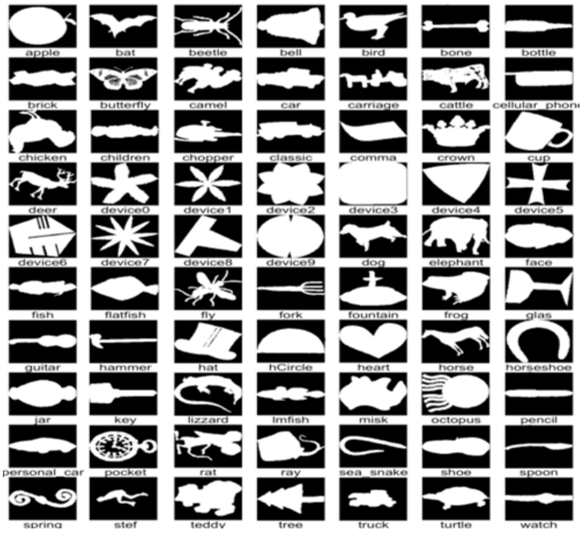


Fig. 1 : MPEG-7 class objects.

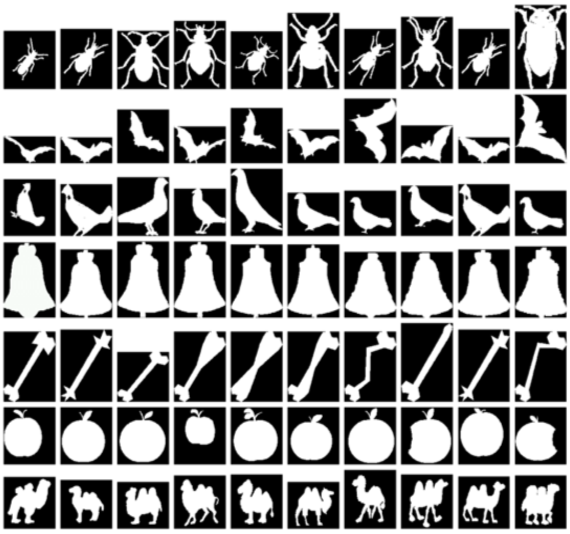


Fig. 2 : MPEG-7: some forms of some classes.

As we are concerned with system coordinates, we extract boundary from image object, and apply a natural normalized arc-length parameterization to get coordinates of N points forming contour. To preserve details of all shapes, we use a value of N=100. Then, each contour is represented by N*3 matrix: first column is x-coordinate, second column is y-coordinate, and third column is curvature estimated on this point. On one point of the curve, curvature describes how much curve direction varies over a small distance. It was used to construct invariant descriptors such as CSS [Mok03], CED [Seb03], BAS [Ari03], etc. So we add curvature information to improve discriminating power of our recognition system.

To see how CNN can perform a classification task on MPEG-7 dataset, we begin with the simplest way considerations. We design a CNN with only one convolutional layer, and three fully-connected layers. Experiments show that 256 filters in the first layer lead to better accuracy rates, compared to other filter numbers.

To overcome the small number of objects used for training, we use data augmentation. It is a label-preserving technique, allowing generation of new objects, obtained from initial ones, by applying rotations using angles of π/4, π/2, 3π/4, π, 5π/4, 3π/2 and 7π/4. This technique improve robustness of CNN and let it learn objects in different positions and orientation. Finally, our dataset is composed of 11200 contours of seventy class object, each class have 160 contours. We split then dataset to train set and test set using proportions of 70% and 30%.

Training CNN with various learning rates until 30 epochs show that optimal learning rate is 5 10^{-3}, according to Fig. 3.

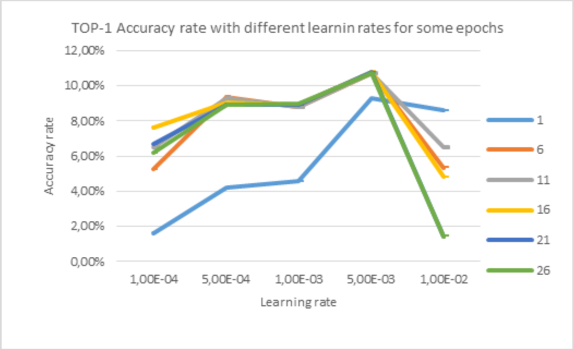


Fig. 3 : Choosing appropriate learning rate according to TOP-1 metric.

Recognition rate obtained with this CNN according to TOP-1 (resp. TOP-5) is very low: 10.71% (resp. 26.79%) after six epochs of training, and remain the same even with 30 epochs. To perform this rates, we added a batch normalization layer, after the convolutional layer of our CNN. Theoretically, this layer attenuates effects of gradient instability, by standardizing and normalizing output of the previous convolutional layer. By adding batch normalization, Top-1 accuracy rate increases by 3.28%, Top-5 accuracy rate increases by 5.56%, both on test set.

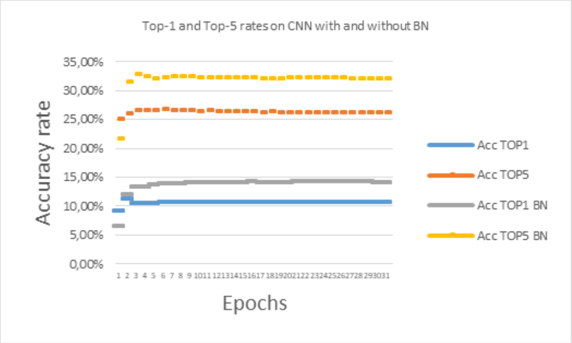


Fig. 4 : Impact of BN use on our CNN.

Figure Fig. 4 compares accuracy rates Top-1 and Top-5 for CNN with and without BN layer.

Despite slightly improving Top-1 and Top-5 accuracy rates, batch normalization negatively affects CNN stability. In fact, comparing accuracy rates of our CNN without and with BN, on noisy data generated from initial data by adding a Gaussian noise, show that CNN without BN learn better than CNN with BN.

With obtained results, it seems that CNN with one convolutional layer are very weak to classify contours of MPEG-7 images, and it is recommended to use more deep architectures. The problem arising with this idea is the shallowness of data representation. In fact, with convolution filters of size 3*3 in the first layer, its output is a one-dimensional vector, on which we cannot apply more convolutions. To defeat this drawback, we use curve evolution [6, 18].

4. CONTOURNET

Curve Evolution & Data Representation

Curve evolution is a technique allowing generation of several new curves, obtained from original one, by smoothing it with Gaussian function. Evolving a given curve with Gaussian function with σ parameter, try to smooth it by eliminating some salient point. Applying successive evolutions to a curve lead finally to a convex curve without salient points. Mathematically, it consists of convolving initial curve with a Gaussian function. Evolution process example is shown in Fig. 5.

The problem of choosing appropriate σ values is far from being resolved. At the beginning, Mokhtarian and Bober [Mok03] chose a regular range to evolve a curve and stop when curve become completely convex. Ben Khlifa and Ghorbel [7,8] introduce different discretization of σ value space.

By experiments, we find that using a regular range from 1 to σ_{max} = 60, leads to convex curves for all shapes we study. Using curve evolution process, a contour from MPEG-7 dataset will be represented by initial contour, concatenated to evolved versions of it. So, input to our CNN will be a 100*(3*61) matrix. Preprocessing process in mentioned in Fig.6. Such representation is very more dense and appropriate for applying convolution, and allowing us to go deeper with CNN.

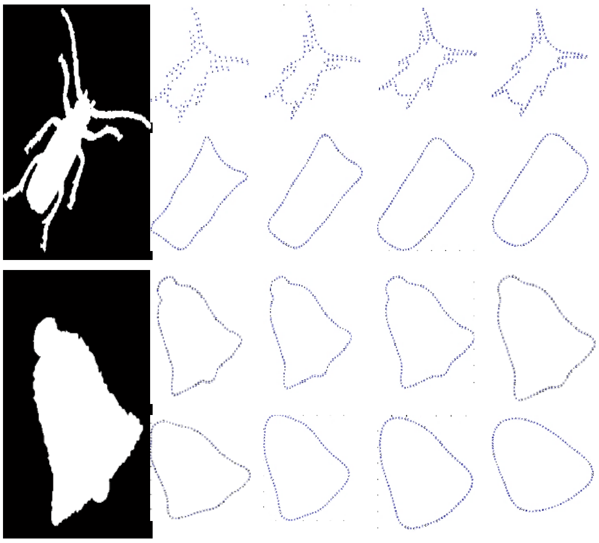


Fig. 5 : Evolution process with σ = 1, 2, 3, 4, 10, 20, 30, 40.

ContourNet

To design a CNN architecture, we need to decide about all of its hyper parameters: number of layers, number of filters in each layer, learning rate, etc. We used cross validation to define optimal number of layers of our CNN. Since CNN with one convolutional layer don't lead to good performances, we studied architectures with two convolutional layers. The table in Fig. 7 illustrates mean validation error with different number of filters. This table shows that architecture with 256 filters in the first layer and 96 filters in the second layer has the best qualification, regarding all studied architectures, to learn features from our dataset.

ContourNet Architecture

ContourNet is composed of two convolutional layers with respectively 256 and 96 filters, and three fully-connected layers. Besides convolution, the two first layers are using a ReLU function to introduce a non-linearity on data, and followed by a Max-Pooling operation, to keep important information while reducing data volume.

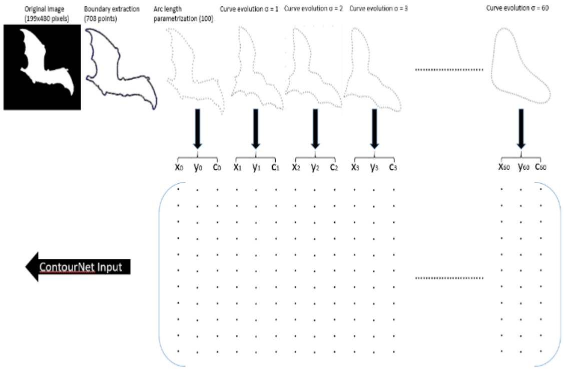

Original Image (199x480 pixels)
Boundary extraction (708 points)
Arc length parametrization (100)
Curve evolution σ = 1
Curve evolution σ = 2
Curve evolution σ = 3
Curve evolution σ = 60
x0 y0 c0 x1 y1 c1 x2 y2 c2 x3 y3 c3
x60 y60 c60
ContourNet Input

Fig. 6 : Preprocessing data for training.

	64	96	128	160	192	224	256
64	25.24	28.24	31.25	43.18	50.83	78.78	74.49
96	22.42	24.75	26.53	29.92	37.79	71.88	67.90
128	21.12	21.89	22.64	26.96	32.17	38.15	48.61
160	18.90	19.43	21.30	22.26	24.25	28.11	29.11
192	19.30	20.14	19.22	21.12	22.05	28.20	30.48
224	19.12	20.78	20.94	19.31	20.80	22.42	25.87
256	18.66	18.13	19.32	19.72	19.77	21.06	27.58

Fig. 7 : Mean validation error Top-1 on two convolutional layers architectures.

Our goal is to find discriminant features in initial contour or in one of its evolved version, that's why we set a filter size of 3*3. In the first layer of ContourNet, filters have size of 3*3, with a stride of 3. According to this configuration, each filter tries to learn discriminant features in the initial contour or in smoothed versions of it. On the convolution result, we apply a ReLU function and a max pooling using a bloc size of 2*2. Output of the layer is introduced to the second layer. In the second layer, filters have a size of 3*3, with unitary stride. This layer attempt to learn nonlinear combinations of features learned in the first layer. Output is then reshaped and delivered to the classification part of our network. Classification part of ContourNet contain three fully connected layers with respective size of 192, 128 and 70. Figure Fig. 8 shows ContourNet architecture. To guarantee circularity of contour data, we use padding by adding the last point to the beginning, and the first point to the end, that's why number of lines of an input is 102.

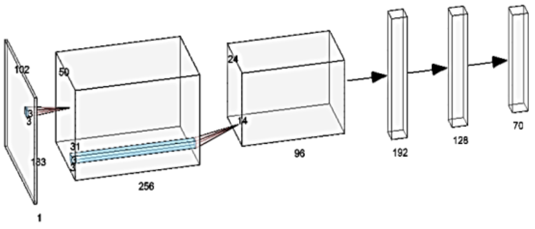


Fig. 8 : ContourNet architecture.

Learning Details

To train ContourNet, we use stochastic gradient descent with various learning rate within several epochs. Weights in all layers were initialized from a normal distribution. Training and tests were carried out on an i7-7700 processor machine, with an NVIDIA GeForce GT 730.

5. EXPERIMENTS

To evaluate ContourNet performances, we use a test set containing 3360 contours, extracted randomly from MPEG-7 dataset. To evaluate generalization of ContourNet, we use Kimia216 shapes, as all of them exist in MPEG-7 dataset. To evaluate stability of ContourNet, we also use a set of 1400 contours, obtained by adding a normal Gaussian noise to original dataset. We use Top-1 and Top-5 metrics to measure error rate.

Among learning rate values tested, a learning rate of 10^{-3} has best effect on learning process. It is shown in figure Fig. 9 that with this rate, ContourNet needs only two epochs to achieve 100% of recognition.

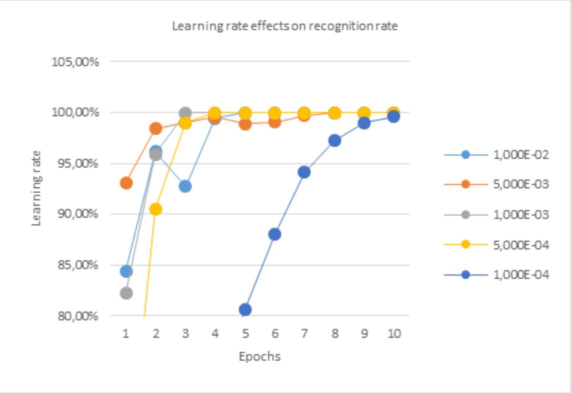


Fig. 9 : Top-1 recognition rate with different learning rates.

Using $5*10^{-4}$ as learning rate, performances achieved are not far from those obtained with 10^{-3}. It took four epochs before achieving 100% of recognition. Except for these two rates mentioned above, learning process is not monotone and show fluctuations when epochs increases. With learning rate of 10^{-2}, recognition rate of ContourNet exceed 96%, decrease to 92% and then reach 100% of recognition after four epochs. With learning rate of $5*10^{-3}$, recognition rate of ContourNet exceed 99.5%, decrease to 98% and then reach 100% of recognition after seven epochs. As we can see, performances gap between learning rates is not large enough, that's why we need to study generalization of ContourNet.

Kimia216 is a dataset containing eighteen object classes, each class has twelve images. All classes in Kimia216 are also part of MPEG-7 dataset. With data augmentation, we create a dataset containing common classes in MPEG-7 and Kimia, composed of 1728 shapes, each class is represented by 96 shapes. Using labels of MPEG-7 classes, we tested these shapes on ContourNet. Results shows that training ContourNet with 10^{-2} learning rate, for six epochs, only through 7840 inputs, lead to an architecture with 100 % recognition rate on MPEG-7 test set, and 92.53 % recognition rate on Kimia classes object (129 misclassified shapes among 1728). Using 5 10^{-3} (resp. 10^{-3} and 5 10^{-4}) as learning rate, the best recognition rate reached is 83.56 % (resp 59.60 % and 55.28 %). Fig. 10 illustrates these results.

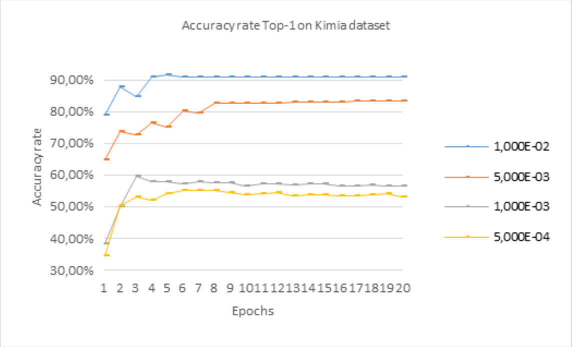


Fig. 10 : ContourNet Tests on Kimia dataset.

On noised dataset we use, we notice that using 10^{-2} as learning rate, give the most stable architecture, compared with others ones. Fig. 11 shows that this architecture achieves a recognition rate of 97.5% on noisy data, while a learning rate of $5*10^{-3}$ (resp. 10^{-3} and $5*10^{-4}$) cannot exceed 86% (resp. 65% and 60%).

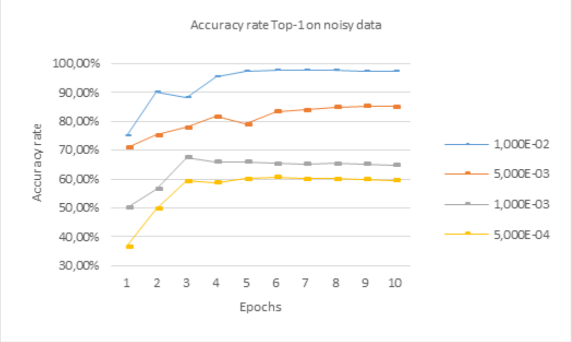


Fig. 11 : ContourNet stability.

Experiments above show that with a CNN with two convolutional layers, containing 256 and 96 filters, followed by three full-connected layers, and using a learning rate of 10^{-2} for five epochs, is an architecture who outperforms the stat-of-the-art on MPEG-7 CE-SHAPE-1 part B object classification. In Fig. 12 we dress a comparative results of ContourNet, with other works from state of the art on MPEG-7 dataset classification.

6. DISCUSSIONS

To understand more what ContourNet was learning, we study shapes from Kimia with which our architecture fails. We notice that all 96 shapes from bone, glass, heart, misk, camel, car, children, face, fountain and ray classes are wholly recognized. Bird class has 28 misclassified shapes, classic car class has 8 misclassified shapes, elephant class has 5 misclassified shapes, fork class has 40 misclassified shapes, hammer class has 8 misclassified shapes, key class has 32 misclassified shapes and turtle class has 8 misclassified shapes. The eight misclassified classic cars were all predicted as jar, corresponding objects have some similarity in their shapes. The five misclassified elephants were all predicted as turtle, corresponding objects have also some similarity in their shapes. The thirty-two misclassified keys were all predicted as guitar, and corresponding shapes are also similar. The eight misclassified turtles were predicted as beetles but corresponding shapes are not similar (see Fig. 16). For the remaining classes, prediction was various for each class. For the twenty-eight misclassified birds, eight were predicted as frog, eight were predicted as chicken, eight were predicted as fork, and four shapes were predicted as elephant (see Fig.13). For the eight misclassified hammers, three shapes were predicted as carriage, and five shapes were predicted as rat (see Fig 14). The worst class in prediction was the fork class, with forty misclassified shapes. One shape was seen as a bone, one other shape was seen as an lm fish, three shapes were predicted as misk, four shapes were seen as cup, five shapes were seen as hammers, seven shapes were predicted as shoppers, eight shapes were seen as spring and eleven shapes were considered as a lizard (see Fig.15) . We illustrate in Fig. 13, Fig. 14, Fig. 15 and Fig. 16 some misclassified shapes from Kimia216, and their corresponding prediction.

Approach	Retrieval Accuracy Rate (%)
WARP [Bar05]	58.50
VP [Lat00]	76.45
CED [Seb03]	78.17
CSS [Mok03]	81.12
BAS [Ari03]	82.37
MCC [Ada04]	84.93
IDSC [Lin07]	85.40
SCN [Zha21]	90.99
FBcC [Tho09]	95.50
ContourNet	100.00

Fig. 12 : ContourNet Performances comparison.

We observe also that in most cases, the number of misclassified shapes for one class, represent the same shape with multiple rotations applied. For the bird class for example, the misclassified shapes were composed of three original shapes and their rotated objects for seven rotation angles, which is equal to 24 shapes. The remaining four shapes were predicted as belonging to elephant class.

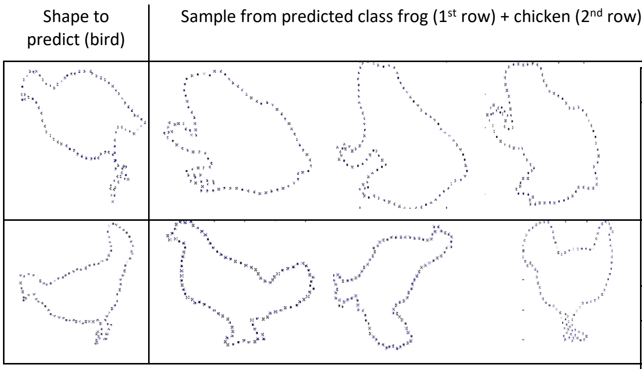

Shape to predict (bird)
Sample from predicted class frog (1st row) + chicken (2nd row)

Fig. 13 : ContourNet misclassified prediction on bird objects.

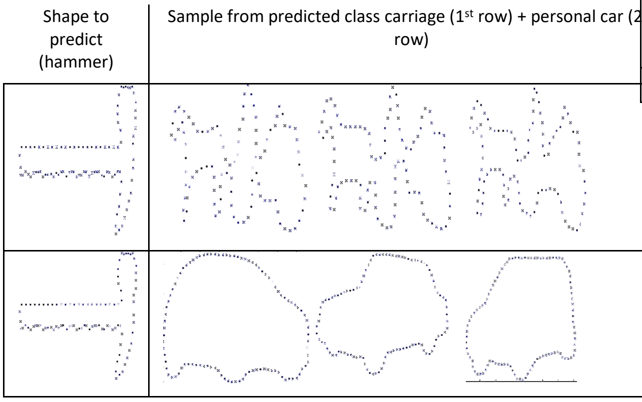

Shape to predict (hammer)
Sample from predicted class carriage (1st row) + personal car (2nd row)

Fig. 14 : ContourNet misclassified prediction on hammer objects.

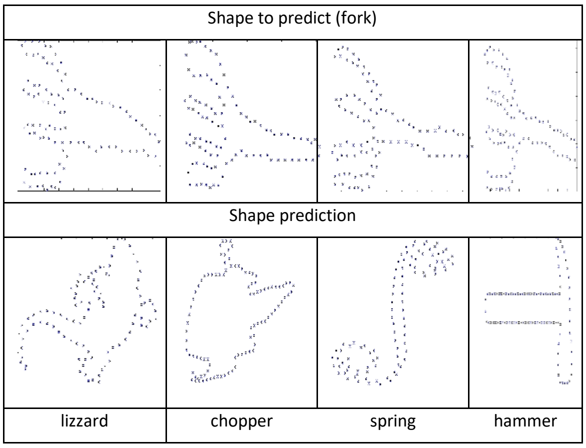


Fig 15 : ContourNet misclassified prediction on fork objects.

7. CONCLUSION & FUTURE WORKS

In this paper, we presented a new convolutional neural network, called ContourNet, to label contours of MPEG-7 dataset. To be convenient with convolution, we apply a curve evolution on initial contours to generate other versions of same objects. A data augmentation technique was used to enrich dataset, based on height angle rotations. A validation step was used to specify optimal hyper parameters to conceive ContourNet. We tested our architecture on both MPEG-7 test set and a noisy version dataset and Kimia dataset, and shows that it performs models used in the state of the art.

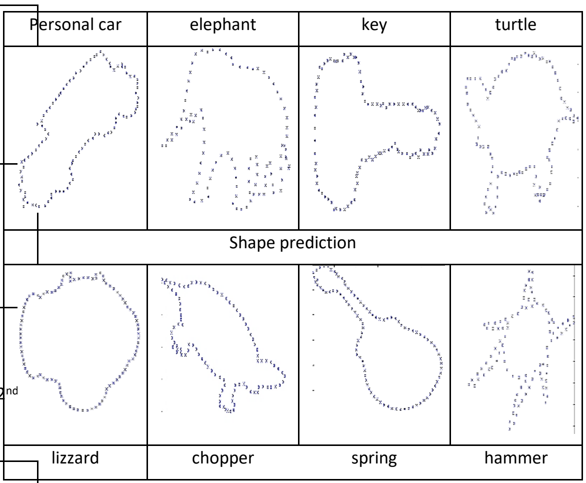


Fig. 16 : ContourNet misclassified prediction with only one class objects.

As a future work, we are thinking about extending this approach to 3D shapes, and how to overcome the fact that R3 is unordered space. Using auto-encoders to label contours is another track to explore, by avoiding supervisor need. More attention will be paid on how to choose scales for curve evolution.

8. REFERENCES

[Lat00] Latecki L.J. and Lakamper R., Shape Similarity Measure Based on Correspondence of Visual Parts. IEEE Trans. On Pattern Analysis and Machine Intelligence, vol 22, No 10, October 2000, pp 1185-1190.

[Bel02] Belongie S, Malik J. and Puzicha J., Shape Matching and Object Recognition using Shape Context, IEEE Trans. On Pattern Analysis and Machine Intelligence, vol 24, No 4, April 2002, pp 509-522.

[Seb03] Sebastian T., Klein P. and Kimia B., On aligning Curves, IEEE Trans. On Pattern Analysis and Machine Intelligence, vol 25, No 1, January 2003, pp 116-125.

[Bar05] Bartolini I., Ciaccia P., and Patella M., WARP: Accurate Retrieval of Shapes using Phase of Fourier Descriptor and Time Warping Distance, IEEE Trans. On Pattern Analysis and Machine Intelligence, vol 27, No 1, January 2005, pp 142-147.

[Lin07] Ling H. and Jacobs D., Shape Classification using Inner-Distance, IEEE Trans. On Pattern Analysis and Machine Intelligence, vol 29, No 2, February 2007, pp 286-299.

[Mok03] Mokhtarian F. and Bober M., Curvature Scale Space Representation : Theory, Applications, and MPEG-7 Standardization, Kluwer Academic Publishers, 2003.

[Ari03] Arica N. and Vural F., BAS: A perceptual Shape Descriptor based on the Beam Angle Statistics, Pattern Recognition Letters, 2003, pp. 1627-1639.

[Ada04] Adamek T. and O’Connor N. E. A Multiscale Representation Method for Nonrigid Shapes with a Single Closed Contour, IEEE Trans. On Circuits System and Video Technology, 2004, pp 742-753.

[Ala07] Alajlan N., Rube E. I., Kamel M. S. and Freeman G., Shape retrieval using triangle-area representation and dynamic space warping, Pattern Recognition 40, 2007, pp 1911-1920.

[Tho09] Thourn K., Kitjaidure Y. and Kondo S., Eigen and Fisher Barycenter contour for 2d shape classification, International Conference on Computing and Communication Technologies, 2009.

[Kri12] Krizhevsky A. Sutskever I. and Hinton G., Imagenet Classification with deep convolutional neural networks, NIPS, 2012.

[Zei14] Zeiler M. D. and Fergus R., Visualizing and understanding convolutional networks, ECCV, 2014.

[Sim14] Simonyan K. and Zisserman A, Very deep convolutional networks for large scale image recognition, arXiv preprint arXiv: 1409.1556.2014.

[Sze15] Szegedy C. and al, Going deeper with convolutions, CVPR, 2015.

[Dro20] Droby A. and El-sana J., ContourCNN: convolutional neural network for contour data classification, arXiv preprint arXiv: 2009.09412v2.

[Ata16] Atabay H. A., Binary shape classification using convolutional neural networks, IIOABJ, 7(5), 332-336, 2016.

[Zha21] Zhang C. and al, SCN: A Novel Shape Classification Algorithm based on Convolutional Neural Network, Symmetry 2021, 13, 499.

Embracing Raycasting for Virtual Reality

Andre Waschk
Universität Duisburg-Essen
Lotharstraße 65
Germany, 47057 Duisburg
andre.waschk@uni-due.de

Jens Krüger
Universität Duisburg-Essen
Lotharstraße 65
Germany, 47057 Duisburg
jens.krüger@uni-due.de

ABSTRACT

This paper proposes an acceleration method for direct volume rendering (DVR). Our approach works like a wrapper or braces surrounding raycasting implementations and requires very few changes to the existing code. Visualization systems can significantly improve their rendering performance in virtual reality setups and make DVR feasible in these environments. The first step—the opening brace—modifies the initial ray construction by adaptively reducing the ray density, hence feeding fewer rays to the raycaster. The second brace step is a composting computation after the ray traversal that re-samples the raycasting results across the screen to reconstruct the final image. The rendering resolution is adapted during run-time to the specifications of the VR hardware and the performance of the renderer to guarantee stable and high refresh rates necessary to avoid severe cyber-sickness symptoms. The presented method utilizes gaze-dependent resolution levels tailored towards the human visual system (HVS) and hardware characteristics found in state-of-the-art head-mounted displays (HMDs). The resolution, and therefore the number of processed fragments during ray traversal, is reduced in the peripheral vision, delivering unnoticeable losses in image quality while providing a significant gain in rendering performance.

Keywords

Technique, Virtual Reality, Acceleration, Volume Rendering

1 INTRODUCTION

Due to the availability of cost-effective virtual reality hardware systems, Virtual Reality (VR) has, over the past years, found its way in many research fields of visualization [O'Leary et al., 2017, El Beheiry et al., 2019]. In particular, domains like astrophysics [Vogt and Shingles, 2013, Baracaglia and Vogt, 2020, Davelaar et al., 2018], engineering [Abulrub et al., 2011, Wolfartsberger, 2019, Wang et al., 2018], archeology [Novotny et al., 2019, Bruno et al., 2010] and also medical science [Huang et al., 2018, Chan et al., 2013, Chheang et al., 2021] have been of greater interest. Most of the publications mentioned suggest, that VR has the potential to improve the discovery and decision-making process in these domains as they benefit from the immersive, interactive, stereoscopic reproduction of complex spatial structures.

Many of the datasets in the aforementioned domains are not just surface structures but rather complex 3D volumes. A well-established approach to visualize volumetric data is direct volume rendering (DVR) [Drebin et al., 1988], which provides deep insight into the volume's data and structure. In the last few decades, significant advancements have been made to bring high-performance, high-quality direct volume rendering to commodity desktop systems [Krüger and Westermann, 2003, Fogal and Krüger, 2010, Meyer-Spradow et al., 2009]. Today, ray-guided volume raycasting systems have been widely adopted on traditional desktop systems with one or multiple monoscopic monitors. For virtual reality setups, however, raycasting is generally considered to be too computationally expensive.

Most affordable VR systems use head-mounted displays (HMD) and spatial tracking hardware to provide an immersive experience and offer spatial motion to the user. This freedom in motion and the usage of HMDs with the close proximity of the displays to the users' eyes come with significant constraints to the VR software. The users' near-continuous motion demands high refresh rates of the display and low latencies of the system. Even relatively short delays, which would be hard to notice on desktop systems, quickly increase the risk of fatigue and nausea, also known as cybersickness symptoms [LaViola, 2000]. Therefore, currently available VR systems utilize HMDs with update rates of at least 90 Hz with a tendency to 120 Hz or more.

https://www.doi.org/10.24132/CSRN.3201.16

In addition to high refresh rates, the close proximity of the display to the user's eyes also demands high resolutions. Present HMDs feature resolutions around 2880 x 1600 pixels with a trend to increase the resolution further to avoid the screendoor-effect [Anthes et al., 2016] and provide an ever-sharper image to the users. For algorithms that are performance bound by the number of fragments processed, the increased number of pixels poses a challenge. A DVR raycaster is such an approach.

In the last few decades, multiple different techniques [Levoy, 1990, Boada et al., 2001, LaMar et al., 2000, Zimmermann et al., 2000] have been developed to accelerate the computation of every individual ray. Early ray-termination, the intelligent choice of the correct transfer function, and progressive rendering [Fogal et al., 2013] have made it possible to visualize large datasets with refresh rates, which can be described as interactive but not high enough for VR usage. A well-established approach to scale rendering speeds is progressive rendering. During the interaction, the rendering quality is reduced by limiting costly computations, such as the number of rays or the sampling rate. When the viewport is fixed, e.g., mouse interaction seizes, the image is progressively refined. In a VR system, however, such a steady state is never really achieved as the camera is constantly adjusted to the tracking system's output.

The method presented in this paper is most closely related to the FAVR approach by Waschk and Krüger [Waschk and Krüger, 2020], i.e., that many of the pixels of the frame-buffer that are sent to the VR systems do not contribute to the perception of the scene. Although the nonuniform resolution distribution of the human-visual-system is well known [Resnikoff, 1989, Valois, 2000], and the use of this phenomenon in computer graphics and visualization is known as foveated imaging [Reder, 1973, Duchowski and Çöltekin, 2007, Murphy et al., 2009], FAVR considers not only human perception but also the nonuniform lens distortion of the HMDs (see Fig. 1). This research, in turn, builds on several works on gaze-dependent rendering [Bektaş et al., 2019, Reingold et al., 2003, Duchowski et al., 2005]. Even modern VR systems, equipped with retrofitted eye-tracking hardware, were considered by Vincent et al. [Vincent and Brannan, 2017] and Albert et al. [Albert et al., 2017]. With the recent developments in consumer VR systems, scientific visualization in VR environments has become of broader interest and has attracted many researchers' attention [Scholl et al., 2018, Usher et al., 2018, Chheang et al., 2021].

The improvements of this work over the FAVR-system can be summarized as follows:

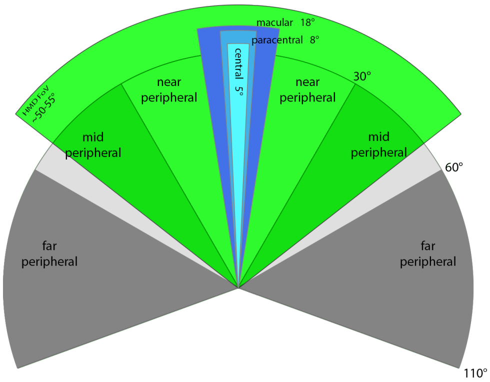


Figure 1: The human visual system is subdivided into numerous fields. Starting from the central vision, the perceived sharpness and the trichromatic perception are reduced the further the angle increases. Notice that modern HMDs cover only the central half of the visual field.

- simpler and more flexible ray-generation implementation
- load-based dynamic layer adjustment
- system scalability to lower end hardware

2 METHOD

To keep our acceleration method independent from the actual raycaster implementation, our method is split into two steps that are injected into existing pipelines (see Figure 2). Conceptually, our approach generates ray-entry and ray-direction information and stores those in texture maps (see Figure 2 (top-left)) to be used as input to the subsequent ray traversal (see Figure 2 (center)). The result of the ray traversal is handed back to our algorithm for final compositing (see Figure 2 (bottom)). Hence, the approach can be considered a bracket around existing ray traversal methods, which allows the combination of practically any ray traversal mechanism with our method, including recent ray-guided volume rendering systems.

We keep the visualization of the bounding geometry as our first step but add a subsampling stage before the ray evaluation. The subsampling-stage takes the entry and exit buffers as input and generates new buffers based on specific hardware characteristics of HMDs. This stage's resulting buffers still represent the bounding geometry but use multiple resolution levels to cover fewer fragments in total (see Fig. 3). The actual implementation and features of this stage are covered in Section 2. The images received from the ray traversal stage are now based on the subsampled input buffers and need to undergo a reconstruction method to generate the final output. To recreate a representation of the original image layout, we added a reconstruction stage to the pipeline, taking the resulting volume visualization and creating

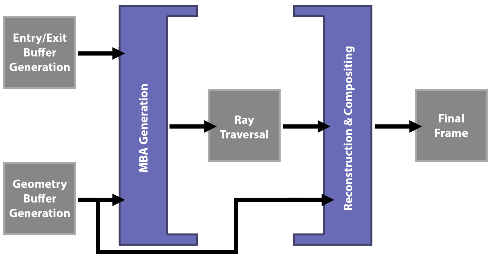

Entry/Exit Buffer Generation
Geometry Buffer Generation
MBA Generation
Ray Traversal
Reconstruction & Compositing
Final Frame

Figure 2: The processing pipeline with the addition of the multiresolution atlas buffer (MAB) generation as preprocess and the reconstruction of the full-resolution image after the ray traversal.

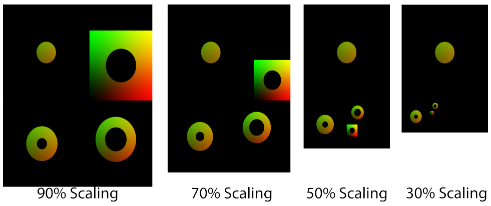


Figure 3: The multiresolution atlas buffer supports arbitrary resolution scaling values between subsequent levels. The resolution levels are ordered to minimize the overall framebuffer size.

a full-resolution image based on our pipeline parameters. Finally, in each frame, we evaluate the system's frame-timings and adjust rendering parameters for the next subsampling stage to maintain a consistent refresh rate (see Section 3).

2.1 Mutliresolution Atlas Buffer Construction

To reduce the number of primary rays that have to be traversed by the raycaster, the system adapts the spatial ray frequency to the decreased spatial perception in the peripheral region of the user's field of view. While foveated rendering systems also follow this approach, by utilizing high-speed eye-tracking hardware to locate the current gaze point, in the case of HMD setups, we do not need to consider the rapid eye movement of the users. Lenses built into HMDs are fixed and provide only a slight angle with the highest projected resolution.

The system uses discrete levels with progressively lower resolutions for specific portions of the image to reduce the number of rays. This layered approach is conceptually similar to MIP-Maps (see Figure 3). Compared to Mip-Maps, however, the system should support varying ratios between levels that are not necessarily a power of two, allowing us to further reduce the noticeable difference between levels.

To realize such a pipeline, our system has to construct the set of levels, forward the nonuniformly generated rays to the traversal, collect the results from the raycaster, and finally generate a consistent output image for display in the HMD.

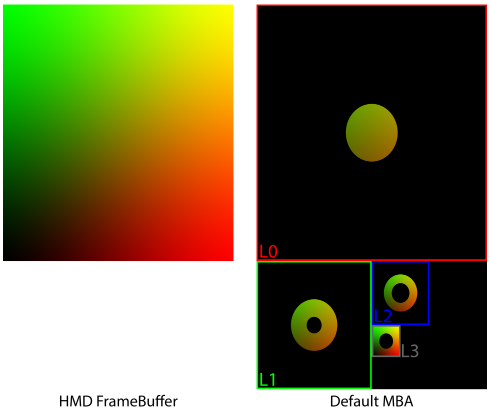


Figure 4: On the left: Screen-space framebuffer as it is used for the HMD rendering. Right: The default MAB setup of the system. The different resolution levels are color coded. The black areas in each level are discarded before the raytraversal.

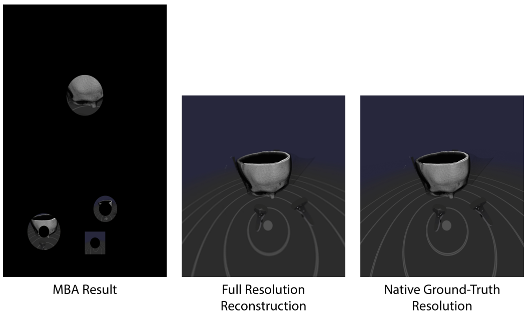

MBA Result
Full Resolution Reconstruction
Native Ground-Truth Resolution

Figure 5: From the left to the right: The result of the raycaster rendered into the MAB. The full-resolution reconstruction that is submitted to the HMD. A native-resolution result as a comparison.

3 IMPLEMENTATION

To follow our approach, it is crucial to first understand the fundamental mapping between the HMD's frame buffer (s. Figure 4 left) and our multiresolution atlas buffer (MAB) (s. Figure 4 right).

The MAB is partitioned into multiple regions, where each region corresponds to the entire frame buffer at a specific resolution. This concept is similar to the levels of a MIP-map texture; however, in our approach, the difference between the various resolutions is not necessarily a fixed 4:1 ratio, which allows us to vary the resolution of each level independently.

A bidirectional mapping between the two spaces is mandatory to perform the two bracket steps of our approach, i.e., the computation of multiresolution ray coordinates for the raycaster and the compositing of the multiple resolutions into a single consistent framebuffer image. This function assigns to every MAB fragment a unique position in the frame buffer and a blending factor. This factor determines what resolutions will be mapped to this frame buffer in the compositing stage. The inverse of the function assigns to each framebuffer-fragment several MAB fragments and blending factors for final compositing.

To explain the process, we look at the necessary steps in reverse order. After the ray traversal is complete, the compositing is initiated by rendering a full-screen-primitive that covers the entire output framebuffer. For each generated fragment, the system has to perform a mapping from the frame buffer to the MAB. This mapping is computed using two functions. The first function (f_a) takes as input the distance to the center of gaze, based on the screen-space coordinates of the fragment, and outputs a set of blending factors for the MAB. The second function (f_b) utilizes the screen-space coordinates of the fragment and outputs a set of corresponding coordinates in the MAB. Both functions have to be evaluated to perform the final compositing. For each nonzero-entry in the blending-factor set (f_a), a lookup into the MAB is carried out (f_b) to obtain the stored color values. These values are blended together into a single color for display.

Before this compositing step, the raycasting is initiated by rendering another full-screen primitive. This primitive, however, covers the entire MAB. Hence, the rasterizer generates all fragments for all resolutions of the frame buffer. To select only those MAB fragments that contribute to the compositing, we compute the inverse of the function mentioned above and determine the blend factor for a given MAB fragment. If the value is zero, the fragment is discarded.

To perform the inverse mapping, we use a third function (f_c) that takes as input the MAB coordinates and outputs two quantities, first, the resolution level of this fragment, and second the corresponding position in the frame buffer. The framebuffer position is used to determine the distance value for function f_a, where the resolution level is needed to select the correct blending factor returned by f_a.

In our current implementation, the function f_a is realized as a 1D lookup texture similar to a transfer functions used in direct volume rendering (Figure 6). The individual resolution levels of the MAB are mapped to the four color channels of the texture. The individual value of each channel describes the blending factor for the corresponding resolution level. To deliver unnoticeable blending, a smooth-step function is added between adjacent resolution levels (Figure 6).

4 FRAME RATE ADAPTION

The frame rate of a DVR system depends on a large variety of parameters. During run-time, even small parametric changes can significantly impact the performance, i.e., changing the transfer function used to sample the data or modifying the view frustum. A frame rate below a certain threshold is not desirable for virtual environments, and even small but abrupt dips in the rendering performance can lead to nausea and other cybersickness symptoms.

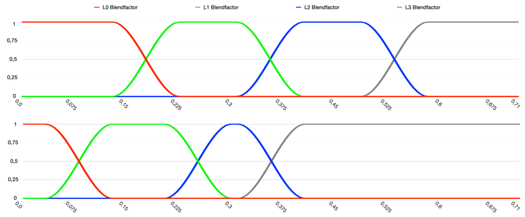


Figure 6: Two different resolution-distribution functions that are integrated into the system. Each line represents a specific resolution level of the MAB (Red = L0, Green = L1, Blue = L3, gray = L4). The Y-Value gives the opacity value for a fragment at the computed distance from the gaze. Values equal to 0 get discarded.

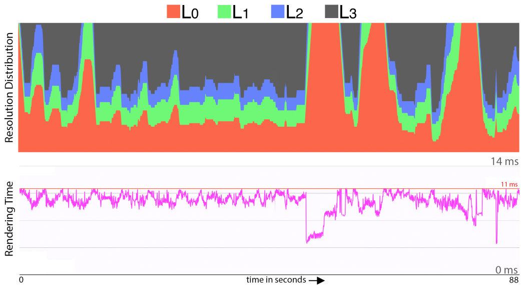


Figure 7: Adjustment of the distribution function is made during run-time. The sequence displayed is the result of our benchmark. The top displays the sizes of the different resolution levels and anchor points. The bottom displays the corresponding rendering time.

Parameters such as the camera position and viewing direction are especially variable when in a VR setup. The user's real-time tracking that enables immersive interaction is one of the major benefits of VR but it can lead to severe fluctuations in overall system performance.

To counter these effects, it is necessary to adjust the rendering system on the fly to maintain a consistently high update rate. The rendering method presented here provides a straightforward adjustment system to guarantee refresh rates above a set threshold. The system tracks the rendering performance and adjusts the resolution levels' distribution on the fly.

The function f_a (Section 2.1), implemented as a 1D-texture, is used in the opening brace as a fast lookup function to determine the corresponding resolution level of a texel and whether or not it has to be traversed by the raycaster. By modifying the texture, the different sizes of each resolution level can be adjusted.

To decide how the resolution levels should be distributed, the system tracks a sliding window of past n frames. First, the previous frame's rendering time is compared against two thresholds, a lower threshold of a minimum acceptable refresh rate and an upper threshold of a sufficiently high refresh rate.

If the performance is below the threshold, we shift the resolution distribution to cause fewer rays to be emitted. If the refresh rate is above the higher threshold, the system increases the number of rays. To compute the increment or decrements in rays, the system considers the gradient of the window of n-frames.

By evaluating the change in rendering time over the last n-frames, the system determines which anchor point between two resolution levels should be moved and how far it should be shifted in the corresponding direction. Minor variations lead to adjustments in the higher resolution levels, only slightly altering the resolution distribution, where high deviations primarily shift the lower resolutions to compensate in favor of the rendering performance (see Figure 7).

5 PERFORMANCE ANALYSIS

To test the scalability of our method, we evaluated our system on two hardware setups. We equipped a desktop computer with a Nvidia RTX 2070 as our graphics processing unit. As a second setup, we tested a laptop with a Nvidia RTX 2060 with a power limit of 80 watts to verify that our system is also scalable to a rendering setup. For our tests, we used the HTC Vive Pro as a VR system with a total display resolution of 2560 x 1440 pixels and a display refresh rate of 90 Hz.

We recorded a fixed set of interactions in the form of rotations, translations, scaling, and modifications in the transfer function to benchmark the system. To further extend our testing, we decided to test three rendering modes. The first mode, the baseline, is a full-resolution raycaster implementation utilizing empty-space skipping and early-ray termination. This mode represents a rendering setup without the opening and closing braces presented in this paper. The next mode tested is the default resolution level distribution (Figure 6) without any automatic adjustments and presents a uniform distribution of the resolutions across all levels. Finally, we benchmarked our method with the addition of automatic resolution adoption. We evaluated the minimum and average refresh rate on both hardware setups for all testes modes.

We selected two datasets for our benchmark common for state-of-the-art scanners, a CT scan with a resolution of 512 x 512 x 404 voxels and an MRI scan with a resolution of 512 x 512 x 392 voxels. To avoid undersampling the volume, we selected a sampling rate of 512 samples.

The results of our benchmarks, presented in table 1 show how our approach can increase DVR performance on HMD setups. The baseline tests on our desktop and mobile/laptop rendering setup could not maintain refresh rates even close to 90 Hz with an average frame rate of 65 fps on the RTX 2070 and 50 fps on the mobile RTX 2060 across both datasets. Using the default MAB setup without the addition of the dynamic resolution adjustments, the system could maintain an average refresh rates above 90 Hz for both datasets on the RTX 2070, but only on one of them for the RTX 2060. With the additional dynamic resolution distribution, the system could maintain the overall refresh rate above the threshold of 90 Hz on all devices

		RTX	2070	RTX	2060m
		1%	avg	1%	avg
MRI	baseline	23.7	51.1	18.5	37.3
	(1) MBA	78.8	105.2	65.1	84.9
	(2) MBA	**90.3**	**108.2**	**90.1**	**91.2**
CT	baseline	37.6	79.0	26.8	62.6
	(1) MBA	80.8	132.9	77.4	97.1
	(2) MBA	**92.9**	**141.6**	**90.9**	**98.8**

Table 1: The table displays the rendering system's average performance on two datasets and two hardware setups. In addition to the average frame rate, we also display the 1% lowest frame rates. We compare a baseline ray-caster with our default (1) MBA setup and our dynamic (2) MBA system.

6 PRELIMINARY TESTS

The previous chapter 4 focused on the performance increment achieved by our acceleration approach. To validate our results, we performed preliminary tests. The study took 20 minutes on average and was conducted in our $16m^2$ lab using the same hardware setup as mentioned above. We recruited 17 participants, all of whom had previous experience with the exploration of volumetric datasets. In our test setup, participants could explore various datasets support by interaction methods such as translation, scaling, rotation, or transfer-function editing.

Our preliminary tests address two aspects of our system, first, is the loss in image quality noticeable to the users, and second, are dips below the rendering threshold more noticeable to the user than changes in the image quality.

To test these two aspects of our acceleration approach, we first let users explore datasets in a controlled environment in which we could verify that the default MAB could maintain 90 Hz most of the time. During exploration, we altered the resolution distribution, further reducing the image quality in peripheral vision. None of the participants noticed any change in overall image quality during exploration, supporting our assumption that the rendering resolution can be reduced in the peripheral vision in HMD scenarios.

To test the second aspect of our approach, we put more strain on the rendering system. We increased the size of the volumetric data and the sampling rate to stress the ray traversal further. Although the default MAB without additional resolution adaption could maintain an average refresh rate above 90 Hz, on some occasions and transfer-function setups, the refresh rate could dip to $\tilde{6}0$ Hz. With the dynamic resolution adaption added to the rendering pipeline, the refresh rate could be caped above 90 Hz at all times. At this point, we randomized the starting condition for each participant. Either the participants started with a fixed-resolution distribution including dips into low refresh rates, or they started with an adaptive resolution distribution but consistent refresh rate. After 1 minute of free exploration, the participant switched to the other mode. After collecting the feedback from each participant, none of them noticed that our system was changing the overall image quality on the fly, but all of them noticed that during the fixed MAB setup, the system was feeling less responsive, and two of our participants experienced nausea during the test.

7 CONCLUSION AND FUTURE WORK

In this paper, we presented a fast and applicable acceleration method for direct volume rendering in the VR context. Our system can provide an interactive exploration of volumetric datasets at 90 Hz or more by reducing the overall number of fragments processed during the computationally expensive ray traversal. Our system's scalability also provides an excellent opportunity to reduce the hardware requirement for volume visualization on hmd setups.

We conducted preliminary tests to assess the assumption that the loss in image quality in peripheral vision is not noticeable in the VR context. In our tests, we focused on the difference between consistent image quality and constant refresh rate. Our results showed that users did not notice changes in the image quality due to changing resolution zones, whereas rapid drops in the refresh rate could lead to nausea.

Thanks to our current implementation, the system can visualize the most common data-set sizes in real time with high refresh rates on a large scale of hardware setups. For the future, we plan to combine our approach with modern state-of-the-art ray-guided rendering systems to visualize even larger datasets interactively in VR.

8 REFERENCES

[Abulrub et al., 2011] Abulrub, A. G., Attridge, A. N., and Williams, M. A. (2011). Virtual reality in engineering education: The future of creative learning. In *2011 IEEE Global Engineering Education Conference (EDUCON)*, pages 751–757.

[Albert et al., 2017] Albert, R., Patney, A., Luebke, D., and Kim, J. (2017). Latency requirements for foveated rendering in virtual reality. *ACM Trans. Appl. Percept.*, 14(4).

[Anthes et al., 2016] Anthes, C., GarcÃa-HernÃ¡ndez, R. J., Wiedemann, M., and KranzlmÃ¼ller, D. (2016). State of the art of virtual reality technology. In *2016 IEEE Aerospace Conference*, pages 1–19.

[Baracaglia and Vogt, 2020] Baracaglia, E. and Vogt, F. (2020). E0102-vr: Exploring the scientific potential of virtual reality for observational astrophysics. *Astronomy and Computing*, 30:100352.

[Bektaş et al., 2019] Bektaş, K., Çöltekin, A., Krüger, J., Duchowski, A. T., and Fabrikant, S. I. (2019). Geogcd: Improved visual search via gaze-contingent display. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, New York, NY, USA. Association for Computing Machinery.

[Boada et al., 2001] Boada, I., Navazo, I., and Scopigno, R. (2001). Multiresolution volume visualization with a texture-based octree. *The Visual Computer*, 17(3):185–197.

[Bruno et al., 2010] Bruno, F., Bruno, S., De Sensi, G., Luchi, M.-L., Mancuso, S., and Muzzupappa, M. (2010). From 3d reconstruction to virtual reality: A complete methodology for digital archaeological exhibition. *Journal of Cultural Heritage*, 11(1):42–49.

[Chan et al., 2013] Chan, S., Conti, F., Salisbury, K., and Blevins, N. H. (2013). Virtual Reality Simulation in Neurosurgery: Technologies and Evolution. *Neurosurgery*, 72(suppl1):A154–A164.

[Chheang et al., 2021] Chheang, V., Apilla, V., Saalfeld, P., Boedecker, C., Huber, T., Huettl, F., Lang, H., Preim, B., and Hansen, C. (2021). Collaborative VR for Liver Surgery Planning using Wearable Data Gloves: An Interactive Demonstration. In *Proc. of IEEE Conference on Virtual Reality (IEEE VR)*, Lisbon, Portugal.

[Davelaar et al., 2018] Davelaar, J., Bronzwaer, T., Kok, D., Younsi, Z., Mościbrodzka, M., and Falcke, H. (2018). Observing supermassive black holes in virtual reality. *Computational Astrophysics and Cosmology*, 5(1):1.

[Drebin et al., 1988] Drebin, R., Carpenter, L., and Hanrahan, P. (1988). Volume rendering. volume 22, pages 65–74.

[Duchowski et al., 2005] Duchowski, A., Cournia, N., and Murphy, H. (2005). Gaze-contingent displays: A review. *Cyberpsychology & behavior : the impact of the Internet, multimedia and virtual reality on behavior and society*, 7:621–34.

[Duchowski and Çöltekin, 2007] Duchowski, A. T. and Çöltekin, A. (2007). Foveated gaze-contingent displays for peripheral lod management, 3d visualization, and stereo imaging. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(4).

[El Beheiry et al., 2019] El Beheiry, M., Doutreligne, S., Caporal, C., Ostertag, C., Dahan, M., and Masson, J.-B. (2019). Virtual reality: Beyond visualization. *Journal of Molecular Biology*, 431(7):1315–1321.

[Fogal and Krüger, 2010] Fogal, T. and Krüger, J. (2010). Tuvok, an Architecture for Large Scale Volume Rendering. In Koch, R., Kolb, A., and Rezk-Salama, C., editors, *Vision, Modeling, and Visualization (2010)*. The Eurographics Association.

[Fogal et al., 2013] Fogal, T., Schiewe, A., and Krüger, J. (2013). An analysis of scalable gpu-based ray-guided volume rendering. *Proceedings. IEEE Symposium on Large-Scale Data Analysis and Visualization*, 2013:43–51.

[Huang et al., 2018] Huang, T.-K., Yang, C.-H., Hsieh, Y.-H., Wang, J.-C., and Hung, C.-C. (2018). Augmented reality (ar) and virtual reality (vr) applied in dentistry. *The Kaohsiung Journal of Medical Sciences*, 34(4):243–248. Special Issue on Dental Research to celebrate KMUD 60th Anniversary.

[Krüger and Westermann, 2003] Krüger, J. and Westermann, R. (2003). Acceleration techniques for gpu-based volume rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, page 38, USA. IEEE Computer Society.

[LaMar et al., 2000] LaMar, E., Duchaineau, M. A., Hamann, B., and Joy, K. I. (2000). Multiresolution techniques for interactive texture-based rendering of arbitrarily oriented cutting planes. In de Leeuw, W. C. and van Liere, R., editors, *Data Visualization 2000*, pages 105–114, Vienna. Springer Vienna.

[LaViola, 2000] LaViola, Jr., J. J. (2000). A discussion of cybersickness in virtual environments. *SIGCHI Bull.*, 32(1):47–56.

[Levoy, 1990] Levoy, M. (1990). Efficient ray tracing of volume data. *ACM Trans. Graph.*, 9(3):245–261.

[Meyer-Spradow et al., 2009] Meyer-Spradow, J., Ropinski, T., Mensmann, J., and Hinrichs, K. (2009). Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations. *IEEE Computer Graphics and Applications*, 29:6–13.

[Murphy et al., 2009] Murphy, H. A., Duchowski, A. T., and Tyrrell, R. A. (2009). Hybrid image/model-based gaze-contingent rendering. *ACM Trans. Appl. Percept.*, 5(4).

[Novotny et al., 2019] Novotny, J., Tveite, J., Turner, M. L., Gatesy, S., Drury, F., Falkingham, P., and Laidlaw, D. H. (2019). Developing virtual reality visualizations for unsteady flow analysis of dinosaur track formation using scientific sketching. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):2145–2154.

[O'Leary et al., 2017] O'Leary, P., Jhaveri, S., Chaudhary, A., Sherman, W., Martin, K., Lonie, D., Whiting, E., Money, J., and McKenzie, S. (2017). Enhancements to vtk enabling scientific visualization in immersive environments. In *2017 IEEE Virtual Reality (VR)*, pages 186–194.

[Reder, 1973] Reder, S. M. (1973). On-line monitoring of eye-position signals in contingent and non-contingent paradigms. *Behavior Research Methods & Instrumentation*, 5(2):218–228.

[Reingold et al., 2003] Reingold, E. M., Loschky, L. C., McConkie, G. W., and Stampe, D. M. (2003). Gaze-contingent multiresolutional displays: An integrative review. *Human Factors*, 45(2):307–328. PMID: 14529201.

[Resnikoff, 1989] Resnikoff, H. L. (1989). *The Illusion of Reality*. Springer US, New York, NY.

[Scholl et al., 2018] Scholl, I., Suder, S., and Schiffer, S. (2018). *Direct Volume Rendering in Virtual Reality*, pages 297–302.

[Usher et al., 2018] Usher, W., Klacansky, P., Federer, F., Bremer, P., Knoll, A., Yarch, J., Angelucci, A., and Pascucci, V. (2018). A virtual reality visualization tool for neuron tracing. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):994–1003.

[Valois, 2000] Valois, K. D. (2000). *Seeing*. Elsevier.

[Vincent and Brannan, 2017] Vincent, P. and Brannan, R. (2017). Tobii eye tracked foveated rendering for vr and desktop.

[Vogt and Shingles, 2013] Vogt, F. P. A. and Shingles, L. J. (2013). Augmented reality in astrophysics. *Astrophysics and Space Science*, 347(1):47–60.

[Wang et al., 2018] Wang, P., Wu, P., Wang, J., Chi, H.-L., and Wang, X. (2018). A critical review of the use of virtual reality in construction engineering education and training. *International Journal of Environmental Research and Public Health*, 15(6).

[Waschk and Krüger, 2020] Waschk, A. and Krüger, J. (2020). Favr - accelerating direct volume rendering for virtual reality systems. In *2020 IEEE Visualization Conference (VIS)*, pages 106–110.

[Wolfartsberger, 2019] Wolfartsberger, J. (2019). Analyzing the potential of virtual reality for engineering design review. *Automation in Construction*, 104:27–37.

[Zimmermann et al., 2000] Zimmermann, K., Westermann, R., Ertl, T., Hansen, C., and Weiler, M. (2000). Level-of-detail volume rendering via 3d textures. In *2000 IEEE Symposium on Volume Visualization (VV 2000)*, pages 7–13.

Depth Completion for Close-Range Specular Objects

S. Pourmand
XLIM, UMR CNRS 7252
Université de Limoges
F-87000 Limoges, France
shahrzad.pourmand@unilim.fr

N. Merillou
XLIM, UMR CNRS 7252
Université de Limoges
F-87000 Limoges, France
nicolas.merillou@unilim.fr

S. Merillou
XLIM, UMR CNRS 7252
Université de Limoges
F-87000 Limoges, France
stephane.merillou@unilim.fr

ABSTRACT

Many objects in the real world exhibit specular reflections. Due to the limitations of the basic RGB-D cameras, it is particularly challenging to accurately capture their 3D shapes. In this work, we present an approach to correct the depth of close-range specular objects using convolutional neural networks. We first generate a synthetic dataset containing such close-range objects. We then train a deep convolutional network to estimate normal and boundary maps from a single image.With these results, we propose an algorithm to detect the incorrect area of the raw depth map. After removing the erroneous zone, we complete the depth channel.

Keywords

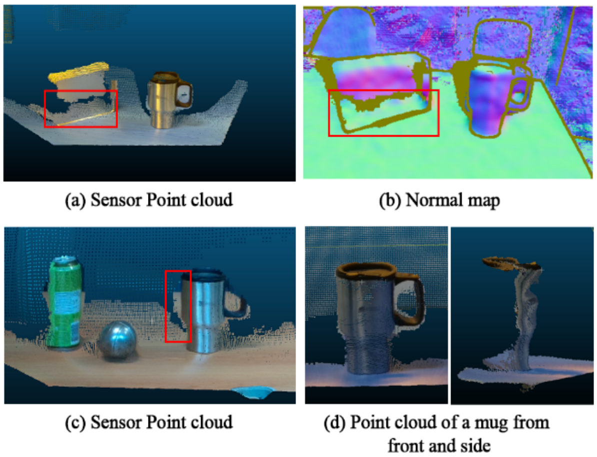
Depth completion, RGB-D images, synthetic dataset, specular reflections.

1. INTRODUCTION

With the availability of affordable RGB-D cameras, there have been a lot of advances in 3D computer vision to improve their capabilities for consumer use. This includes a variety of applications on 3D reconstruction, robot manipulation, virtual and augmented reality. However, these RGB-D cameras have some limitations; their depth estimation frequently suffers from missing or incorrect values, especially on shiny and transparent objects.

The Intel Realsense D435 [IR21] is a good and affordable RGB-D camera that produces a color image along with its corresponding depth map. It uses stereo vision technology with an infrared projector/sensor to accurately measure depth. However, we can observe some limitations by testing the camera, illustrated on figure 1:

- Missing data in the raw depth-map;
- Pixels with invalid values on the shiny or transparent surfaces;
- Confusion of the depth near edges of the foreground object and background or two close foreground objects;
- Noisy estimation of shiny object's curvature.


(a) Sensor Point cloud

(b) Normal map

(c) Sensor Point cloud

(d) Point cloud of a mug from front and side

Figure 1. Examples of Intel Realsense D435 camera limitations; (a) shows invalid values on a shiny box, (b) is the corresponding normal map. (c) shows the confusion of the depth near edges and (d) shows the

https://www.doi.org/10.24132/CSRN.3201.17

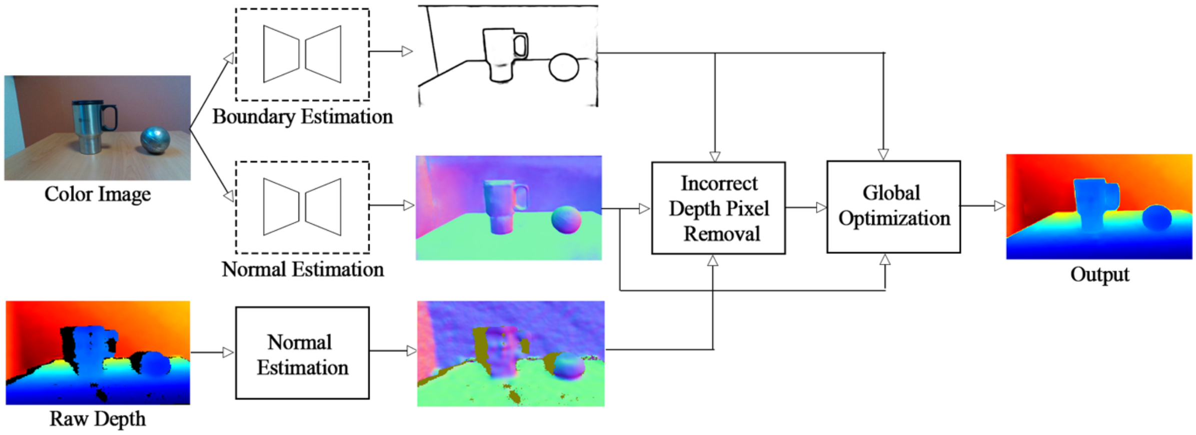


Figure 2: Proposed pipeline: surface normals and boundaries are estimated from the color image. They are then used to remove the unreliable regions in the depth map. The depth is finally completed using a global optimization step.

In this work, we propose a pipeline to refine captured depth values of glossy objects for close-range applications. Our approach is illustrated in figure 2. Given a single color image, we first use convolutional neural networks (CNN) to predict both the surface normals and objects boundaries. The CNN is trained with our own synthetic datasets (section 3.1). At this point, we can remove incorrect depth values in three steps. First, we remove the boundary of the objects using an estimated boundary mask. Second, we compute the normals directly from the sensor raw depth map and compare them to the normals estimated from the color image in order to remove areas where they differ significantly. Third, we use morphological transformations to remove the persisting noise. Eventually, we use a global optimization approach to fill the holes in the depth map.

2. RELATED WORK

2.1. Depth completion algorithm

Since the appearance of low cost RGB-D cameras, many methods have been proposed to overcome their limitations, which appear as holes and incorrect areas in the depth map. Most of these methods utilize a color image as guidance to correct the depth map.

Huang et al [HHC14] use the edges of the color image to detect the unreliable zone in depth. Then they removed the existing depth in the unreliable zone to prevent erroneous depth propagation. Next, they used the fast marching method (FMM) [T04] which was first introduced for image inpainting and then revised for depth inpainting [LGL12] to complete the depth.

In recent years, with the latest advances in deep learning, a lot of data-driven approaches have emerged. Zhang and Funkhouser [ZF18] completed the depth channel of an RGB-D image using global optimization. They first predict the normals from the color image and then solve for completed depth. Their work is mainly focused on large indoor environments with large holes. However, the lack of depth denoising prevents their method to obtain correct results in depth completion.

Shreeyak et al. [SMPN20] proposed a modification to the previous method to estimate the depth of close-range transparent objects for manipulation. Their method consists of detecting transparent objects in the color image, removing their depth values, and using global optimization to complete the depth of all missing areas. However, their method is not suitable in our case because the depth captured by depth cameras is not as inaccurate for specular objects as it is for transparent objects. As a result, removing all the depth values of the objects would be excessive.

Xian et al [XQLL20] have developped a method to eliminate incorrect depth based on a segmentation network trained with labelled RGB-D images of large indoor scenes.

2.2. 3D understanding from the single color image

It has been shown that a single color image can be used to directly infer depth [EPF14, LKY17], or other geometric features [SR12, EF15, WFG15]. One of the geometric features that is tightly coupled with depth is normal-map. Normal-map is a pixel-map that contains the orientation of the surface at each point. Wang et al. [WFG15] proposed a method to predict surface normals at local and global scales, then they used a fusion network to combine both predictions into a final result. Eigen and Fergus [EF15] designed a single multiscale convolutional network to predict depth, surface normals, and semantic labels.

Recently, Yinda Zhang et al. [ZS17] utilized a U-net architecture with VGG-16 as backbone for normal estimation and achieved state-of-the-art results.

In this work, we use U-Net architecture with inceptionv4 [SI16] as our model.

2.3. Dataset for training neural network

There are several real-world datasets available for training neural networks including NYU depth v2 [SHKF12], ScanNet [DCSH17], and Matterport3D [CDFH17] which are all highly used datasets.

However, real-world datasets generally suffer from noise and missing data in shiny, transparent, or distant areas as they are limited by the quality of the capturing device.

To overcome these limitations, researchers have been increasingly using synthetic data for training networks for vision and robotic tasks [KMHV17, SMPN20]. The advantage of using synthetic data for training is that it is easy to obtain pixel-perfect ground truth data. Furthermore, it is affordable to create large-scale datasets. In this work, we will generate our own dataset containing a mix of close-range shiny (with specular reflections) and rough objects (with diffuse reflections) to meet our needs.

3. PROPOSED METHOD

3.1. Generating data

There are several RGB-D datasets available to train neural networks consisting of synthetic or real-world data. Yet, none of these are especially focused on close-range glossy objects. Therefore, we choose to generate our own customized dataset. We have created a scene generator in Blender. The generated scene would consist of a plane with different objects on it. The ground (plane) is randomly textured from 82 textures [URL1]. The camera is randomly placed in the scene, always focused on the ground. The environment lighting is randomly chosen from 120 indoor HDRI environments [URL2]. Next, some objects are chosen from primitive shapes presented in figure 3. These shapes represent a large variety of objects: smooth sphere, mesh with smoothed or hard edges, objects with or without holes. 3D objects are added to the scene as in Shreeyak et al. [SMPN20], by using Blender physics simulation to drop them on the ground: objects will collide with each other and the plane insuring a random final positioning. The textures of the objects are chosen from 47 textures [URL1] (including Blender's checker pattern) or a solid color. Their roughness and metallic values are also chosen randomly.

Each of the created scenes is rendered with Blender Cycles at the resolution of 256x256 pixels. Corresponding normal map and boundary map are also generated for each image. Before generating the normal map, a bounding-sphere is created in the scene. This sphere contains the plane and all the objects to ensure that if the HDRI background is visible in the rendered scene, the normals of those areas do not stay empty. This does not create an issue when training our network because the background information is irrelevant in our context. The generated normals are then oriented from world space to camera space and normalized in the range of -1 and 1. The resulting normal map is saved in OpenEXR format. This method permits us to create thousands of images for the next step (illustrated in figure 4).

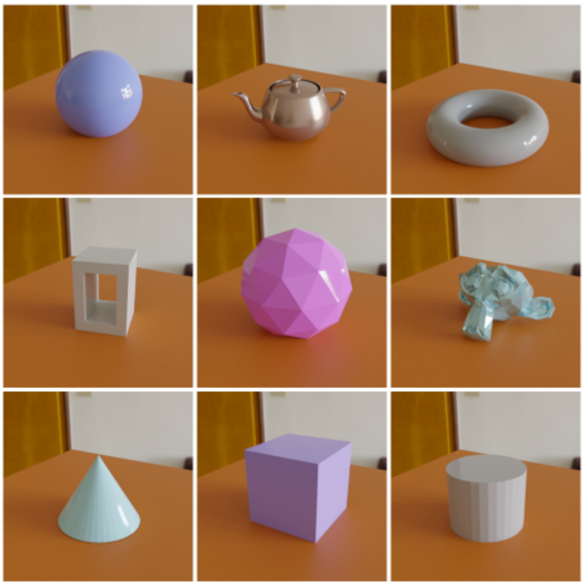


Figure 3: Blender primitive shapes used for dataset generation

3.2. Network Architecture and Training

We experimented with U-Net [RFB15] and DeepLabV3 [CP17] as our encoder-decoder model architecture. We trained DeeplabV3 with resnet101 [HZ16] and Unet with resnet101, inceptionv4 [SI16] and vgg16 [SZ14] as backbones [PY20]. All models have been pre-trained on ImageNet [RDSK15].

Our experiments led us to choose U-Net with inception4 as our model architecture for both normal estimation and boundary detection, as it generates better results with our dataset. U-net is (a U-shape) encoder-decoder architecture that uses skip-connections between corresponding layers of encoder and decoder. We train both networks on the data that we generated using Blender.

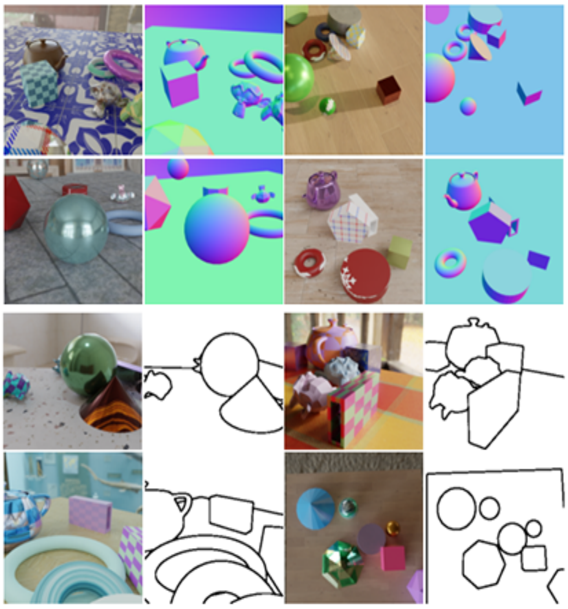


Figure 4: Some examples of our synthetic training datasets. First and second rows illustrate sample data used for normal estimation training, while the two bottom rows illustrate data used for boundary detection training

Normal estimation: We modify the last layer of the network to generate a three-channel output. Therefore, we will have the three components of the normal vector per pixel: x, y, and z. We then normalize the output to L2-norm. To calculate the loss, we use an element-wise dot product between the network's output and the generated synthetic ground-truth.

We train the network in two steps; first, we train the network on 10,000 synthetic images, which are then augmented using Gaussian blur. In the second step, we reduce the learning rate and re-train the network on 5000 synthetic images of very close-range objects. We have again augmented the data using Gaussian blur. We empirically found that re-training the network on very close-range objects with a reduced learning rate gives better results than training the network on all data in a single step.

Boundary estimation: For this task, we modify the last layer of the network to generate a single channel output. The activation function of the last layer is then set to sigmoid to output the value of each pixel between 0 and 1. Note that the boundary pixels of the ground-truth data are zero, while the non-boundary pixels are one. We dilate the boundary pixels before training to be able to use the result of the network directly for boundary removal, as described in section 3.3.

We use the binary cross-entropy loss as the loss function. Since the number of non-boundary labels is significantly higher than the number of boundary pixels, we use a loss weight that is ten times greater for boundary pixels than for non-boundary ones, as suggested by Yang et al [YPCLY16].

We train the network on 5000 synthetic images and we augment them using Gaussian blur and random changes of brightness, contrast, saturation, and hue.

3.3. Incorrect Depth Pixel Removal

RGB-D cameras generally do not measure edge depth correctly, causing confusion between the depths of the foreground object and the background or the depths of two close foreground objects. Thus, we begin by removing the depth of the object's boundaries. First we create a binary mask from the estimated boundary map, and then we apply this mask to the depth map to remove the border of objects.

Next, we compute the normal map of the depth map by considering adjacent 3D points [ZPK18]. We compare each of these normals to its corresponding one in the normal map estimated from the color image using a simple dot product. We remove the areas where the difference between the angles of the normals is greater than 30 degrees: we experimentally found that the normal estimation network gives nearly perfect results from this value, thus the normal map generated from color image would be reliable, see section 4. This step is performed to remove erroneous zones that appear on our objects as a result of specular reflections.

At last, we remove any small noise persisting in the two previous steps. This is done by generating a mask from the depth map by setting every zero value to zero (which are the holes in the depth map) and every non-zero value to one. We then use a morphological opening on the mask with a 5x5 circular structuring element as our kernel. We apply the mask to the depth map to remove the noise.

3.4. Global optimization

After removing the unreliable depth from the depth map, we use the optimization algorithm proposed by Zhang and Funkhouser [ZF18] to complete the depth. This approach takes the estimated normal map and estimated boundary map from the color image as input and uses them as a guide to complete the depth map. The optimization algorithm has the objective function as follows:

$$E = \lambda_D E_D + \lambda_S E_S + \lambda_N E_N B \,,$$

$$E_D = \sum_{p \in T_{obs}} \| D\,(p) - D_0(p) \|^2,$$

$$E_N = \sum_{p,q \in N} \| < v(p,q), N(p) > \|^2,$$

$$E_S = \sum_{p,q \in N} \| D\,(p) - D(q) \|^2,$$

where E_D is the distance between the estimated depth and the raw depth, E_S ensures that the neighbor pixels have similar depth values and E_N uses the dot product to ensure that the normal estimated from the image and the estimated depth are consistent. B has a value between [0,1] and down-weights the E_N based on the predicted probability that a pixel is on the boundary.

4. EXPERIMENTAL RESULTS

We evaluate the normal estimation network on a synthetic test set; the test set consists of 100 synthetic unseen images that we generated using our data generation method, but composed of new (unseen) objects. For all the images, we compute the mean and the median of the angular error measured between the estimated normals and ground truth ones. We also compute the percentage of the pixels with errors smaller than 30, 22.5, and 11.5 degrees (common metric also used for example in [SMPN20]). The results are presented in Table 1 and show a perfect matching above 30°.

	Mean	Median	11.25°	22.5°	30°
Our model	24,2	18,3	38,7	54,3	99,8

Table 1. Evaluation metrics on the synthetic test set for normal estimation network.

We have tested our approach on real data captured by the Intel Realsense D435 camera. The testing environment consists of a table with a few objects on it. We use a variety of specular objects with different shapes and materials; Some of them have similar shapes to the synthetic objects while others have previously unseen shapes. The images and depth maps are taken under ambient light and at the resolution of 424x240 pixels.

We train and test our proposed pipeline on a desktop computer equipped with an Intel Core(TM) i9 2.80GHz CPU with 16 GB RAM and NVIDIA GeForce RTX 2080 GPU. For each test of RGB-D data with a resolution of 424x240 pixels, estimating normals and boundaries of the color image and removing unreliable region from the depth map takes about 0.3 seconds while optimization on depth map takes about 1.8 seconds. We compare our proposed method with the method proposed by Zhang and Funkhouser [ZF18]. We have used the code they provide on their github page without making any modification. For the optimization used in both Zhang and Funkhouser's and ours, we use the following values: $\lambda_D = 1000, \lambda_S = 0.001$ and $\lambda_N = 1$.

As illustrated in figure 5, we can see that our proposed pipeline greatly improves the depth completion results in close-range tasks. Note that the object boundaries got sharper and depth inside the objects is more consistent, especially for the box in rows three and four.

5. CONCLUSION

In this paper, we propose a depth correction and completion pipeline for close-range shiny objects. Our contribution is twofold; first, we prepare a synthetic dataset consisting of close-range specular and non-specular objects. This permits to train a CNN with pixel-perfect values of normals and boundaries. Second, we propose an algorithm to detect and remove unreliable regions in depth map, based on the normals and boundaries predicted from a single color image. Eventually, we complete the depth using the optimization method proposed by Zhang and Funkhouser [ZF18]. A theoretically possible drawback could arise if the depth of a specular object is incorrect, while its normal map is accurate. We will investigate this as future work. We will also estimate the influence of noise differential (synthetic images vs real captures) in our kind of pipeline. Despite theses possible drawbacks, in most cases, our method produces excellent results for close-range objects.

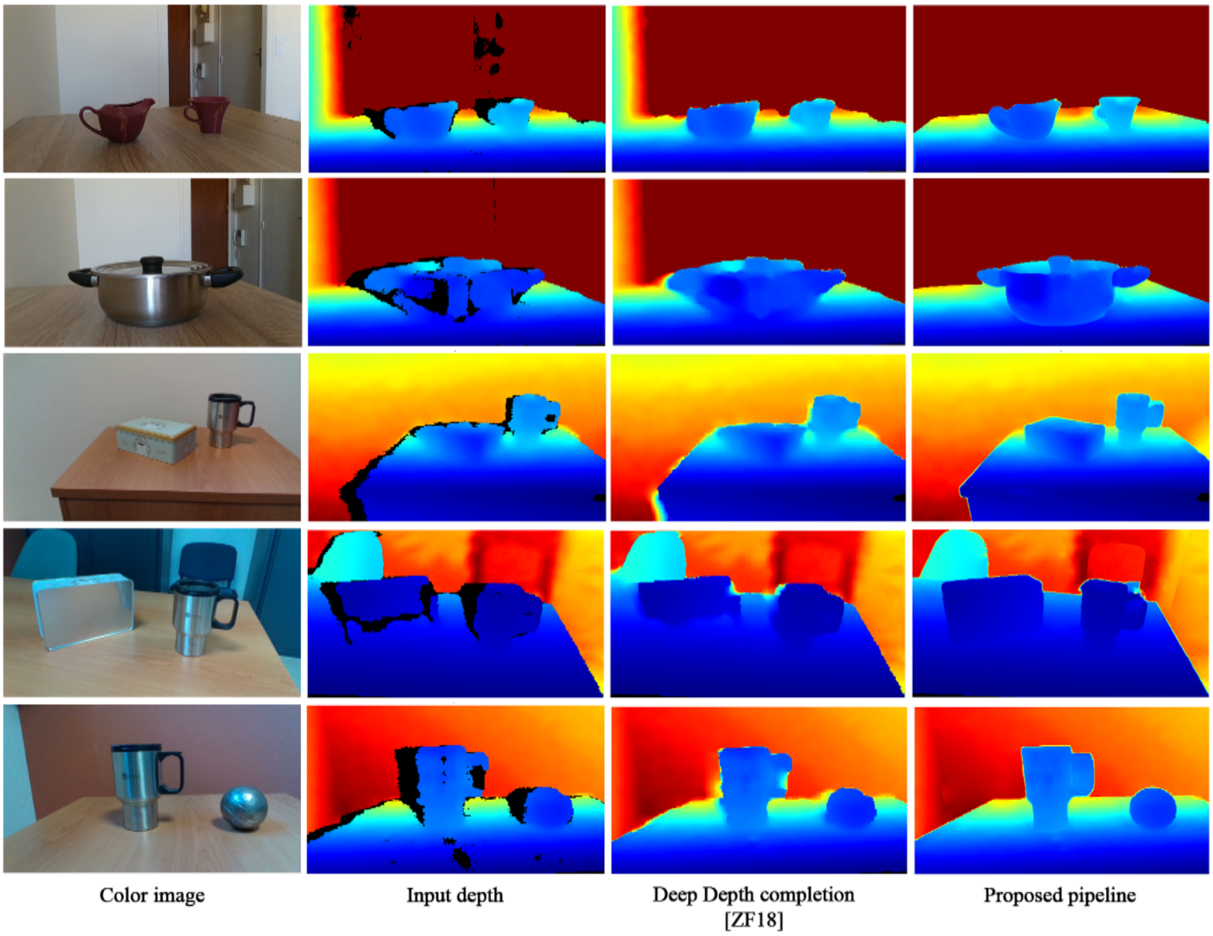


Figure 5: Although we used the same depth completion approach as Zhang and Funkhouser [ZF18], this comparison demonstrates that our proposed pipeline considerably improves the depth completion results for close-range specular objects. (In the first two rows, the depth beyond one meter is clipped for better visualization)

REFERENCES

[CDFH17] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In International Conference on 3D Vision (3DV), 2017

[CP17] Liang-Chieh Chen, George Papandreou, Florian Schroff, Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. 2017

[DCSH17] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[EF15] David Eigen, Rob Fergus. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. ICCV, 2015.

[EPF14] David Eigen, Christian Puhrsch and Rob Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. CoRR, 2014.

[HHC14] Yung-Lin Huang, Tang-Wei Hsu, Shao-Yi Chien. Edge-aware depth completion for point-cloud 3D scene visualization on an RGB-D camera. In IEEE Visual Communications and Image Processing Conference, VCIP 2014

[HZ16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016

[IR21] Intel RealSense D400 series Product Family Datasheet. URL: https://dev.intelrealsense.com /docs/intel-realsense-d400-series-product-family-datasheet

[KMHV17] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An Interactive 3D Environment for Visual Ai, 2017.

[LGL12] Junyi Liu, Xiaojin Gong and Jilin Liu. Guided inpainting and filtering for Kinect depth maps. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR), 2012

[LKY17] Jun Li, Reinhard Klein and Angela Yao. A Two-Streamed Network for Estimating Fine-Scaled Depth Maps from Single RGB Images. In Proceedings of the IEEE International Conference on Computer vision, 2017.

[PY20] Pavel Yakubovskiy. Segmentation Models Pytorch, 2020, URL: https://github.com /qubvel/segmentation_models.pytorch

[RDSK15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Anderj Karpathy, Adity Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.

[RFB15] Olaf Ronneberger, Philipp Fischer and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234-241. Springer, 2015.

[SHKF12] Nathan Silberman, Derek Hoiem, Pushmeet Kohli and Rob Fergus. Indoor segmentation and support inference from rgbd images. In European Conference on Computer Vision, pages 746-760. 2012

[SI16] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proc. AAAI Conference on Artificial Intelligence, 2016.

[SMPN20] Shreeyak S. Sajjan, Matthew Moore, Mike Pan, Ganesh Nagaraja, Johnny Lee, Andy Zeng, Shuran Song. ClearGrasp: 3D Shape Estimation of Transparent Objects for Manipulation - IEEE International Conference on Robotics and Automation (ICRA), 2020

[SR12] Alexander G. Schwing and Raquel Urtasun. Efficient Exact Inference for 3D Indoor Scene Understanding. In ECCV, 2012.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014

[T04] Alexandru Telea. An image inpainting technique based on the fast marching method. In Journal of Graphics Tools, vol.9, 2004

[URL1] https://resources.blogscopia.com/category /textures/, and https://3dtextures.me/

[URL2] https://polyhaven.com/hdris/

[WFG15] Xiaolong Wang, David F. Fouhey and Abhinav Gupta. Designing Deep Networks for Surface Normal Estimation. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[XQLL20] Chuhua Xian, Kun Qian, Guoliang Luo, Guiqing Li and Jianming Lv. Elimination of incorrect Depth Points for Depth Completion. In proceedings of CGI 2020, pp. 245-255

[YPCLY16] Jimei Yang, Brian Price, Scott Cohen, Honglak Lee and Ming-Hsuan Yang. Object Contour Detection with a Fully Convolutional Encoder-Decoder Network. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 193-202

[ZF18] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single RGB-D Image – The IEEE Conference on computer vision and pattern recognition, 2018

[ZPK18] Qian-Yi Zhou, Jaesik Park and Vladlen Koltun. Open3D: A Modern Library for 3D Data Processing, 2018.

[ZS17] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

DEEP LEARNING FOR THE DETECTION OF CAR FLAP STATES

Benoît Guérand
Karlsruher Institut für Technologie
Daimler Protics GmbH
benoit.guerand@mercedes-benz.com

Fabian Scheer
Daimler Protics GmbH
fabian.scheer@mercedes-benz.com

Mustafa Demetgül
Karlsruher Institut für Technologie
mustafa.demetguel@kit.edu

Jürgen Fleischer
Karlsruher Institut für Technologie
Juergen.Fleischer@kit.edu

ABSTRACT

In recent years, deep learning and object detection has continuously attracted more attention. Especially in the automotive world where many car manufacturers are currently investigating its possible applications. On production lines, even if processes are more and more automatized mistakes can happen and hinder the performance of an industrial plant. In this study, a method and application of object detection-based deep learning algorithm to detect open flaps on cars, like doors, trunk, hood etc. is examined. With this approach, the advantages of gap detection in cars on production lines, specifically the application of Resnet50 Convolutional Neural Networks (CNNs) and transfer learning in an industrial use case, are demonstrated. We show how the problem of detecting open flaps on cars is modeled in a way that a CNN can be applied to this new kind of application and present a detailed evaluation of the results and challenges. Finally, many suggestions are given for future applications of similar algorithms.

Keywords

deep learning, Resnet50, RetinaNet, door gaps, object detection of open car flaps, Convolutional Neural Network (CNN), industrial use case, production line

1. INTRODUCTION

Generally, assembly is done with the help of robots in automotive production lines. Misalignment of assembled parts or open parts of the vehicle body (e.g. doors, trunks, etc.) during production can cause collisions between the robots and car components. To avoid such problems computer vision techniques can be used to improve process productivity and production flexibility, provide position information to the robot controller and set or correct the robot's path [1, 2]. For this, a camera must be placed at different positions on the production line. Typical challenges of machine vision techniques applied to automotive production lines remain, like the influences of lighting, light reflections and the changing of image backgrounds. Object detection may be an important option for solving these problems [3].

Object detection is gaining a lot of attention recently as its applications cover a very large field of studies [4]. Object detection has many different application domains such as pedestrian detection, behavioral analysis, autonomous driving, face recognition, pattern recognition, car vision, and so on [5]. When the literature is examined, there are studies on the detection of car body problems like scratches with deep learning, but there is no study on the detection of open flaps or gaps on cars on the production line. Open car flaps are the doors, the trunk, the hood or the tank cap.

The purpose of this article is to contribute to the literature on the detection of open flaps or gaps on cars. This problem is of particular interest to the automotive industry as it causes potential damage to vehicles during production, and this problem has not yet been solved using object detection algorithms. A method based on Resnet50 is presented and it is

https://www.doi.org/10.24132/CSRN.3201.18

shown how to model the problem of detecting open flaps on cars as an object detection problem. An industrial use case in a real production line is used for a detailed evaluation and discussion of the presented method.

2. RELATED WORK

Looking at the literature studies on this subject, Kang [3] tried to reduce the wrong decision-making processes caused by ambient lighting and light reflection problems during the detection of problems by monitoring the automobile production line with a camera. In their studies, the distance between the car body and the door part and the door was obtained with the measuring device combining the laser slit light source and the LED patterned light source [3].

Kosmopoulos and Varvarigou in [6] introduced a system for automatic gap inspection using computer vision. It can measure the lateral and gap size of the gap. The measurement setup consists of two calibrated stereo cameras and two infrared LED lamps, which are used to highlight the edges of the range through specular reflection [6]. Considering these studies, it is applied for a single door and many tools are needed.

Mazzetto in [7] have implemented deep learning-based object detection, semantic segmentation, and anomaly detection to assist in finding automotive assembly errors. They worked on the brake, disc, and motor assembly [7].

A study has also been carried out on the determination of the outer body and interior parts of the cars with object detection. For this, Resnet 50 and Darknet are used. Although this study is close to our study, only parts of it were detected. There is no abnormal detection [8]. This is the closest study to the study we have done. However, object detection has been used in the problem detection of different vehicle parts. Rahimi in [9] used the YOLO deep learning algorithm, which distinguishes between vehicles and people at an automotive manufacturing plant using object detection [9].

Apart from these studies, detection of vehicle damages without using object detection is done with deep learning methods [10, 11]. In addition, there are studies carried out for the diagnosis of problems in different production lines [12-13]. However, this technique is not suitable for our application.

When doing quality control with images, the main issue is the effect of ambient and light reflection, the background and the doors will be in approximately the same place as the cars will be located in the same place each time. Object detection avoids using many complex filters to remove the effect of such parameters [14]. This domain doesn´t have a lot of literature so this paper will help increase resources regarding this topic. The goal of our paper was to add more literature on this specific topic as deep learning is an always growing field and such implementation was relevant for our case in the production plant.

This paper demonstrates a new approach to detect the state of vehicle flaps in production lines using the object detection-based Resnet 50 deep learning method.

3. THEORETICAL BACKGROUND

There are many algorithm architectures that are able to solve this problem [15-16]. But according to the literature [17-18] and what is the most efficient at this time; the ResNet 50 method is chosen to solve the object detection problem. ResNet50 is a special type of Convolutional Neural Network (CNN) [17-18]. CNNs are typically used the most to compute image data, as the architecture is well suited to detect patterns such as curves, lines, etc. A CNN typically consists of three layers. The first is a convolutional layer which is like a filter. It has a matrix as input and a kernel (filter). This is shown in Figure 1.

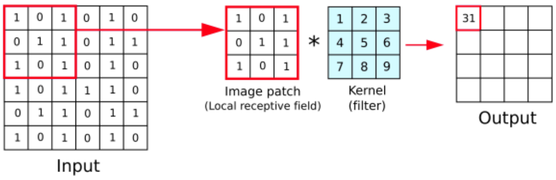


Figure 1: Convolutional layer, *similar to [19]*

Then it has the pooling layer which reduces the parameters of the input matrix such as a max-pooling layer where the maximum of each quadrant is taken (see Figure 2).

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \rightarrow \begin{bmatrix} 6 & 8 \\ 14 & 16 \end{bmatrix}$$

Figure 2: Pooling

The last layer is a fully connected layer which is a feed-forward neural network.

The algorithm used in this paper is an ssd_resnet50_v1_fpn_coco (also called Retinanet

[20]). This is an assembly of various architecture and this combination was firstly introduced to have one stage proposal-driven mechanism while maintaining performances. The main component of this architecture is a ResNet50. It is a type of CNN where a convolutional layer, a pooling layer, 50 convolutional layers, an average pooling layer and a fully connected layer at the end is used. It is our backbone network.

The specificity of a ResNet is that the group of 50 convolutional layers is using “identity shortcut connections”(Figure 3). This was introduced at a time when the main method for building a network was to add more layers. But it was shown [20] that is wasn’t the most effective method, as the accuracy saturated when the network was converging: this is called degradation. The ResNet architecture Figure 4 solved this problem. The novel solution introduced residual networks (shown in Figure 3). They allow to skip layers. This will permit the network to avoid the layers that are nuisances for the results during the regularization phase. This results in a very deep network without the burden of the large amount of layers.

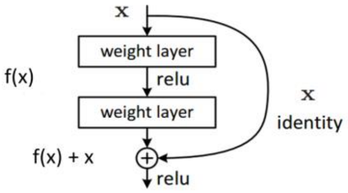


Figure 3: Resnet Structure, *similar to [21]*

The main advantage of this architecture according to its authors [21] is that you have better results than the actual network with a lower amount of parameters. It is using fewer parameters than its former counterparts, such as VGG algorithms [17]. The VGG-16 uses 134,7M parameters whereas the ResNet50 only uses 23,9M.

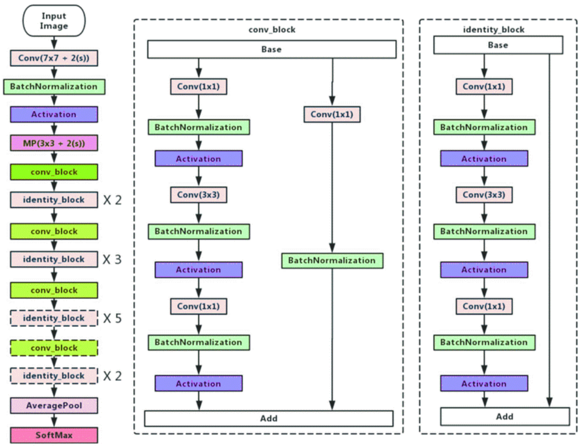


Figure 4: Resnet50 architecture, *similar to [22]*

To measure accuracy in object detection, the metric IoU is introduced, which means Intersection over Union. The formula of the IoU is the following:

$$IoU = \frac{|Area(Ground\ truth\ box) \cap Area(Anchor\ box)|}{|Area(Ground\ truth\ box) \cup Area(Anchor\ box)|}$$

An IoU bigger than 0.5 is considered as a “good” metric [23].

We have manually defined our bounding box of the original object: it is the ground-truth bounding box. Then the algorithm will generate multiple random bounding boxes with different scales and different forms: these are called anchor boxes.

In Figure 5 we have the ground truth box marked in green which shows the truth and an anchor box that is marked in orange. The intersection of both boxes is marked in light orange (Figure 5).]

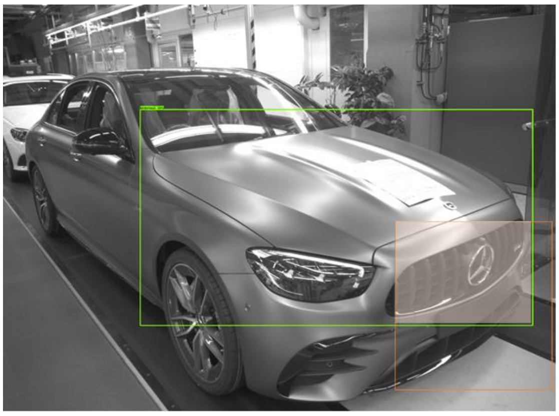


Figure 5: IoU illustration

For object detection, the algorithm is fed with data consisting of pictures where the position of the object is manually annotated. These are the ground truth boxes. The coordinates of the object form a box. To make predictions, the algorithm will guess where the boxes are. But to help find the boxes, we add some layers to the algorithm: convolutional layers. These layers are forming a so called Single-Shot Detector (SSD). Although SSD have a lower accuracy by 10% in average [20] of a two-stage method, they are designed for speed and efficiency. SSD will divide the image using a grid and will try to detect the objects in each grid. Then the SSD calculates the probability that the object is present by comparing it with predefined anchor boxes (Figure 6).

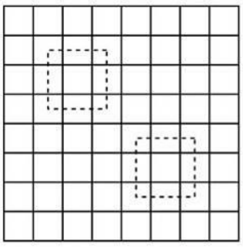


Figure 6: Object detection, *source [24]*

We adopted the feature pyramid network from [26] (Figure 7). This helps finding objects from different scales on the same image. This is very useful for our situation, where we have very small gaps for the doors and bigger gaps when the trunk is open. The main principle is to take an image and subsample it using convolutional layers to transform it into lower resolution and lower image size, hence forming a pyramidal structure while keeping the strongest features using lateral connections [20] (Figure 7).

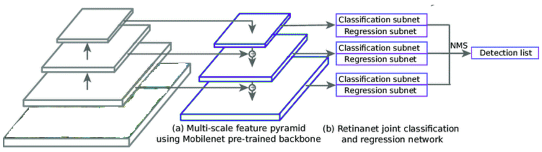


Figure 7: Feature Pyramid Network, *similar to [20]*

To achieve better results with a small amount of data, it is advisable to use a technique called transfer learning [25]. Thereby, a model is used that has already been trained on a different dataset. This allows to reach far better results with the own dataset. In our case the algorithm was trained on a dataset called COCO17 (Common Objects in Context) [29] and we trained the algorithm from this starting point.

4. EXPERIMENTAL SETUP

Our use case of detecting open flaps on cars normally is a problem in the domain of measurement technology. We reformulate this problem as an object detection problem and define separate classes for the state of each flap. To evaluate the presented method, a setup in a real automotive production line is used. We drew our process in Figure 10 where we use a classical deep learning algorithm training, optimizing process.

Three cameras mounted on a steal structure are used for the system. The cameras cover three different viewing directions onto a car (see figure 8): the front view for the hood; the left view for doors on the left side and the trunk; the right view for doors on the right side, the tank cap and the trunk again.

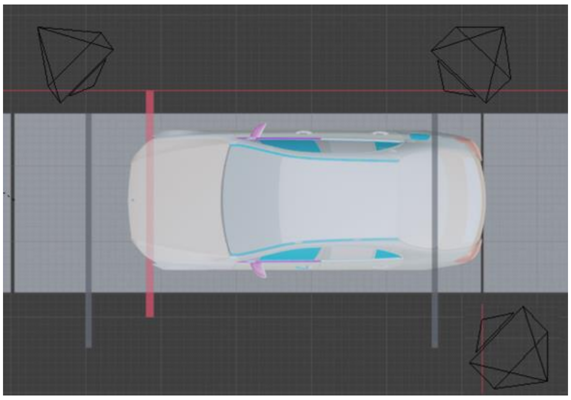


Figure 8: Camera layout

For the system, color cameras of the type UEye UI-3000SE-C-HQ with global shutter, 12 mm focal lense and pixel dimensions of 4110x3006 were used. The cars run on a conveyor belt and are thereby passing the viewing range of the cameras (Figure 9). A light trigger signals that a new car enters the station. Approximately 5 meters after the light barrier the car is in full view of the cameras and pictures for each camera are taken. This is triggered by a fix increment value of the conveyor belt that is measured by a rotary encoder. The signal of the light barrier resets this increment value to zero for each new car.

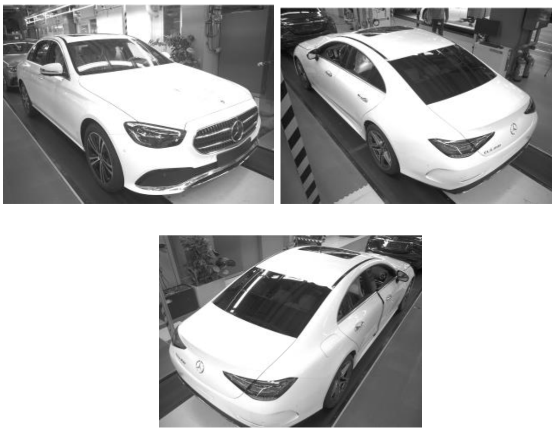


Figure 9: Different camera angles

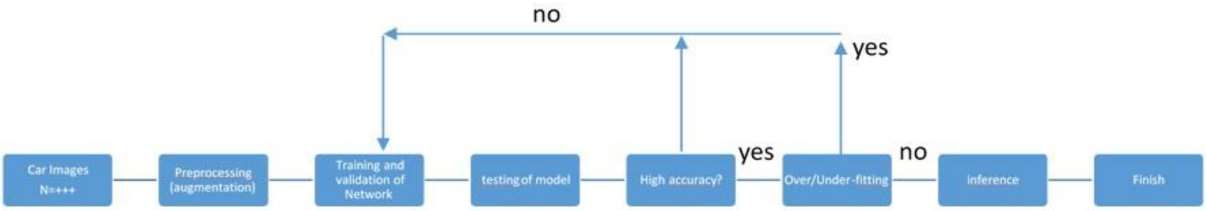


Figure 10: Flow chart of process

It is important to note that we trained three different models: one for each camera angle. The tables and data that follow are valid for one model.
For example, we trained our networks with 265 pictures of cars, this mean that we used 265 pictures of cars per network: i.e.: 265*3 pictures. As you can see in the following tables.

Model	*Number of cars*
Type 1	3*37
Type 2	3*35
Type 3	3*183

Table 1: Number of cars for training per model

State *Door*	*OPEN*	*CLOSED*
door_front_left	3*67	3*188
door_front_right	3*87	3*168
door_rear_left	3*83	3*172
door_rear_right	3*80	3*175
hood	3*103	3*152
tank	3*39	3*216
trunk	3*39	3*216

Table 2: Training sample of cars

In Table 1and in Table 2 we show the training sample to improve reproductability of our results. If you compare with Table 4 you can see that we have more closed state pictures for training and more opened state pictures for testing. It is because for training we wanted to emphasize the default and correct state to be sure to have this state very well learned by the network. Then for the opened states in the test sample, it is because first of all, there are many different openend states (very small gap, small gap, open) and furthermore we wanted to check all of the different cases, when some opened flaps were hiding others for examples.

To train our three networks pictures of 265 cars per network are used. We then tested the three models using 3*944 pictures of cars in the same proportion as in the 3*265 samples for training. The training/testing set repartition was generated according to the production flow on the days we were on site. The proportion of the types were as follows for the test sample:

Model	*Number of cars*
Type 1	3*179
Type 2	3*121
Type 3	3*644

Table 3: Number of cars for testing per model

To formulate our problem of finding open flaps on a car in a way that a CNN can handle it, we defined 14 different classes for the object detection. In concrete, this means two classes representing the state of open or close for every flap to detect. In Table 4 all classes are shown. To have a sufficient amount of recorded car picture per class, the cars were modified manually during production. Table 4 also shows how many car pictures for each class were considered.
After the acquisition of the pictures, they were labeled by using the software labelme [27]. The classes we used were one class per object and per status: for example, the door_front_left_open and door_front_left_closed are two different objects. These classes are the objects that are searched for in the images.

State *Door*	*OPEN*	*CLOSED*
door_front_left	3*767	3*177
door_front_right	3*693	3*251
door_rear_left	3*771	3*173
door_rear_right	3*760	3*184
hood	3*373	3*571
tank	3*854	3*90
trunk	3*856	3*88

Table 4: Test sample of cars

When creating the bounding boxes with labelme, a jsonfile format is obtained that is consecutively transformed into PASCALVOCXML and then CSV using python script in order to finally obtain the desired format TFrecord. That is a format specifically for tensorflow [28]. This format is used for all training and validation data (see Figure 11).

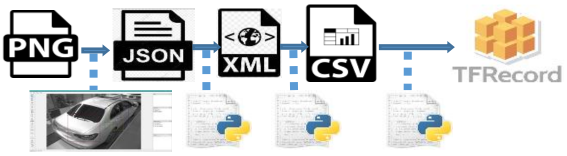


Figure 11: Turning images into input data with Object Label

To test our images we ran the algorithm and made tests according to the flow chart in Figure 10. Changes were made if the results were unsatisfying.

An accuracy of over 95% was aspired, because that is approximately the accuracy a human person can achieve on this specific task, i.e.: the overview of the gaps of a car on the conveyor belt or rather to check if a flap is open or not. This is comprehendible, since a very small gap of an open car door is visually hard to identify, even for humans. This precision was obtained after personally spending days on production lines and seeing how many flaps we were able to detect without external assistance. One of the most important things for the training was that the bounding boxes for the labeled ground truth was big enough to detect the gaps of open flaps. As the object detection algorithm is trying to find the best IoU for its anchor boxes, it can lead to bad detection results if the boxes are too small or thin. To keep detections where the algorithm achieves high confidence rates, we used an IoU threshold of 0.6. This corresponds to a confidence rate of 60% for the boxes and can be seen exemplarily in Figure 11. According to [20] the bounding box threshold and parameters for their detection should be high. In the literature an IoU value of 0.2 [20] or 0.5[23] is recommended, but in our evaluations we found many cases where the algorithm was confident with 55% or 58%. Therefore, an IoU of 60% is feasible for us to achieve high quality results.

However, this can also lead to bad results, as smaller gaps were undetected. To avoid this, we took 0.2 as the minimum aspect ratio and 0.2 also as the minimum area for the minimum object covered. In Figure 12 you can see our output.

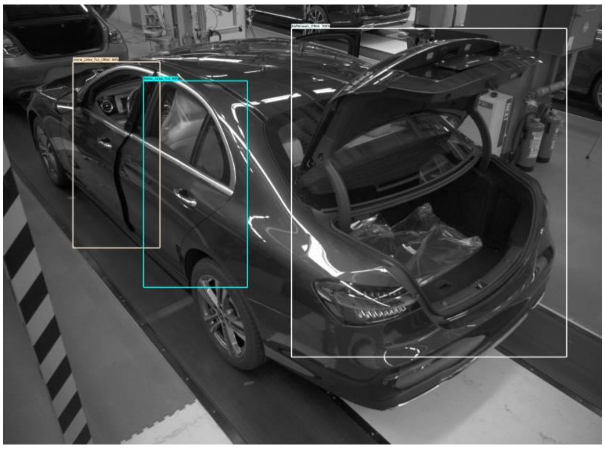


Figure 12: Output of our system. With confidence values of 99%, 99% and 100% from left to right

5. RESULTS

For the CNN training and detection a test system with the computing power of an i7-10750H-2,60GHZ CPU, 32GB RAM and an NVIDIA quatro T2000 graphics card with 4GB Ram was used.

The training time of 265 images for one of the three cameras was 4 hours and 30 minutes with a batch size of two. In Figure 13 the learning rate for 25.000 steps is shown. The networks were parametrized with a decaying learning rate for the training as it is more interesting to slow down the training as the model is converging.

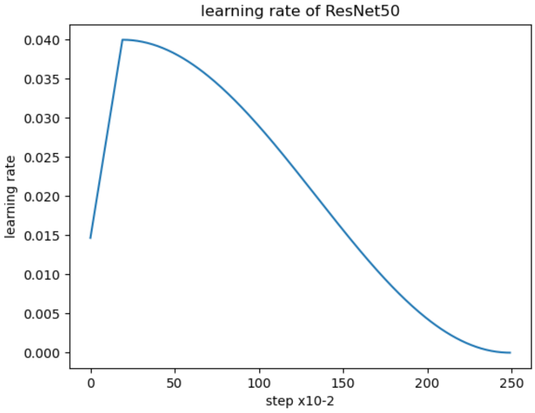


Figure 13: Learning Rate

The losses and training time for each of our three models are similar. That is why we only displayed curves for one model (right side camera). When training the algorithm we recorded the localization and classification losses. These represent the inaccuracy of the predictions and as seen in the sequence (see Figure 14), they continuously diminish in amplitude and value.

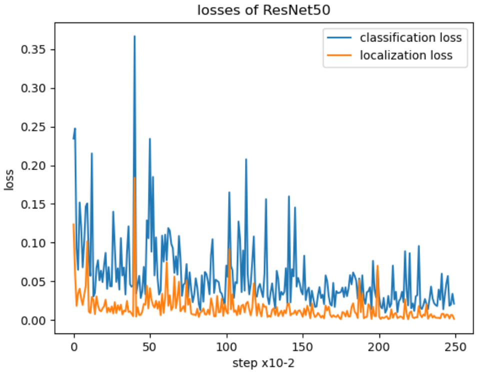


Figure 14: Classification and Localization Loss

The classification loss is the ability of the network to find the appropriate class and the localization loss is the ability to find the class at the right place. In Figure 14 it can be seen that both of these losses are converging towards zero. Even if there is still some noise along this decay, this is a proof for the convergence and performance of our networks. The losses are continuously striving towards zero, meaning that the network is finding more and more the right objects at the right place.

To avoid overfitting and help the networks be less likely prone to make highly nonlinear decisions, an optimization function was introduced that minimizes the global loss with a regularization term. This aims at having weights as close to zero as possible. Having a look at the regularization losses (see Figure 15), it can be clearly seen that the convergence of the losses towards zero shows the diminishing necessity of regularization.

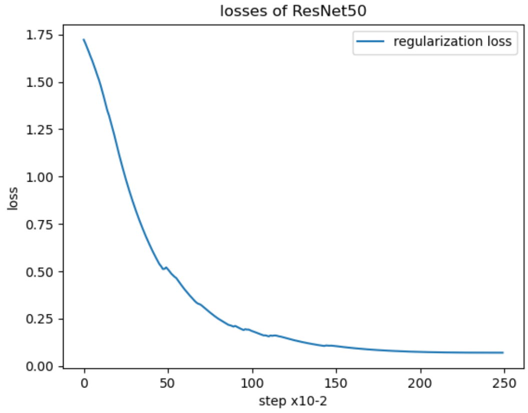


Figure 15: Regularization loss

To measure the performance of our algorithm, the results are presented in a confusion matrix (see Table 5). In object detection a true positive is if the right object is detected at the right place, a false positive is a false detection, a false negative is when the ground truth object is present but the algorithm didn't detect it. A true negative is every part of the picture where no object was predicted. As it is irrelevant in object detection, it can be ignored in the confusion matrix.

After training, 3*944 new car images per camera were used to make predictions and measure the performance of the algorithm. The results are shown in Table 5 with a separate matrix for each camera.

confusion matrix

True Positive	False Positive
False Negative	True Negative

Front camera

943	2
1	0

Left camera

941	4
1	0

Right camera

939	6
3	0

Table 5: Confusion Matrix

It is important to note that the total of each matrix can be above 944 (number of image used for testing), because if we detect more doors than there are in the test images, it is a false positive.

In Table 6 the results of the three cameras are put together by having a look on the results per car and not per camera.

935	12
5	0

Table 6: Confusion matrix per car in regards to the detections of the 3 cameras together

During setup, we had several complications. During the normal production flow, it is hard to control every factor and it leads to abnormalities.

For example, some people may walk in front of the cameras precisely when the picture is taken. There could also be missing pieces on cars such as bumpers. Finally, there also were papers or additional tape on some cars to point out faults that had to be corrected later on. Even if those faults or errors are very infrequent they still have an impact on the results of the predictions for those cases, where the gaps of the car are occluded. If a paper is in the middle of the hood for example and nothing is occluded the prediction works properly. During normal production flow this shouldn't happen since our setup is a defined field test.

To overcome these errors in the future new classes specifically for these errors can be defined, so that they become properly identified during production flow.

With the results of Table 5 and Table 6 the precision can be calculated, which is the ability to find only relevant objects and also the recall, which is the ability to find the truth. Precision can be seen as the evaluation of *quality* and recall as the evaluation of *quantity*. The results for the precision and recall can be found in Table 7. They were computed by using the results of Table 6.

	Formula	Value
Precision	TP/(TP+FP)	0.987328405
Recall	TP/(TP+FN)	0.994680851

Table 7: Precision and recall from Table4

With our method a recall rate of 99.5% was achieved, which means that in the majority of the cases the right objects were found. This is very important for our use case of finding open flaps, since our method is able to find the right gaps for 99.5% of the cases. A recall of 100% may be technically reachable under perfect production situations, but in reality if a station is not fenced it may happen that someone walks through the camera image and occludes the car.

The precision is 98.7% and shows the ability of our method to determine only the relevant gaps. In many of the false positive cases the algorithm would have detected a door open with a confidence of 80% and the same door closed with a confidence of 65%. As we take all results above 60% we have a true positive and a false positive at the same time. Even if the door is really open and the algorithm is having more trust in detecting the door open, this leads to a smaller precision. This is the most common error we found in our examinations. To solve this in the future further post processing rules or conditions can be introduced, e.g. that only one object in the class front door (open/close) has to be detected and that always the one with the highest confidence should be taken as the result.

When a human is watching over the production flow a success rate of approximately 95% is achievable because small gaps can be missed because of boredom, tiredness or lacking attention. Looking at the entire vehicle with our method, a sensitivity of 98.7% was achieved (see Table 7), which is superior to the sensitivity of a human by 3.7%.

Finally, we empirically found in the evaluation of our method that these good results are only achievable if the IoU is above 60%.

6. CONCLUSION AND FUTURE WORK

In this paper the new application of deep learning and transfer learning for the industrial use case of detecting open flaps on cars on a conveyor belt was shown. The modeling and break down of the problem into a classification problem was presented, as well as a detailed evaluation of the method and discussion of the challenges. By merely using picture details for the training, the presented method is nearly independent to changes in the factory in the image background. This kind of application and its evaluation is one of the first in the industry and contributes to the literature in this domain of applied research. Very high precision and recall rates over 98 % have been achieved, whereas the errors arised from lacking car parts, occlusions by passing peoples or never seen objects, e.g. paper with checklists, that are added in the picture details used for detection. This is only a problem if the gap between car parts is mostly occluded. The presented method was trained with three different car models and due to the usage of transfer learning, it will be easily adaptable for other models in the future. This is part of our future work. Moreover, this approach can be very useful in other production lines and for the detection of other objects or gaps.

7. REFERENCES

[1] Scheer,F., Neumann,M., Wirth,K., Ginader,M., Oezkurt,Y., Mueller,S.: Evaluation of model-based tracking and its application in a robotic production line. Journal of WSCG, 2020

[2] Scheer,F., Loos, M., Neumann,M.: Model-Based Tracking on Conveyor Belts: Evaluation and Practical Results in the Automotive Industry, WSCG, 2021.

[3] Kang, D. S., Lee, J. W., Ko, K. H., Kim, T. M., Park, K. B., Park, J. R., ... & Lim, D. W., "A study on measurement and compensation of automobile door gap using optical triangulation algorithm." Journal of the Korea Society of Die & Mold Engineering 14.1, 8-14, 2020.

[4] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, "Deep learning for generic object detection: A survey." International journal of computer vision 128.2, pp.261-318, 2020.

[5] Derman, E., & Salah, A. A. , "Continuous real-time vehicle driver authentication using convolutional neural network-based face

recognition." 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). IEEE, 2018.

[6] Kosmopoulos, D., & Varvarigou, T., "Automated inspection of gaps on the automobile production line through stereo vision and specular reflection." Computers in Industry 46.1, 49-63, 2001.

[7] Mazzetto, M., Teixeira, M., Rodrigues, É. O., & Casanova, D., "Deep learning models for visual inspection on automotive assembling line." arXiv preprint arXiv:2007.01857, 2020.

[8] Stappen, L., Du, X., Karas, V., Müller, S., & Schuller, B. W., "Go-CaRD–Generic, Optical Car Part Recognition and Detection: Collection, Insights, and Applications." arXiv preprint arXiv:2006.08521 (2020).

[9] Rahimi, A., Anvaripour M., and Hayat K. "Object Detection using Deep Learning in a Manufacturing Plant to Improve Manual Inspection." 2021 IEEE International Conference on Prognostics and Health Management (ICPHM). IEEE, 2021.

[10] Malik, H. S., Dwivedi, M., Omakar, S. N., Samal, S. R., Rathi, A., Monis, E. B., ... & Tiwari, A., "Deep Learning Based Car Damage Classification and Detection." EasyChair Preprint 3008,2020.

[11] Kyu, P. M., & Woraratpanya, K.. "Car damage detection and classification." Proceedings of the 11th international conference on advances in information technology, 2020.

[12] Vedang C., Surgenor B. "Fault detection and classification in automated assembly machines using machine vision." The International Journal of Advanced Manufacturing Technology 90.9, 2491-2512, 2017.

[13] Popescu, D., Ichim, L., "Image based fault detection algorithm for flexible industrial assembly line." 22nd International Conference on Control Systems and Computer Science (CSCS). IEEE, 2019.

[14] Behrendt, K., Witt, J., "Deep learning lane marker segmentation from automatically generated labels." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017.

[15] Mingxing T.,Le Q.V., "Efficientnet: Improving accuracy and efficiency through AutoML and model scaling." arXiv preprint arXiv:1905.11946, 2019.

[16] Haisong H., Wei Z.,Yao L., "A novel approach to component assembly inspection based on mask R-CNN and support vector machines." Information 10.9, 282, 2019.

[17] deep-learning - Why is resnet faster than vgg - Cross Validated. https://stats.stackexchange.com/questions/280179/why-is-resnet-faster-than-vgg/280338. Accessed 8 november 2021.

[18] Fung, V. "An overview of resnet and its variants." Towards data science, 2017

[19] Reynolds, Anh H. « Anh H. Reynolds ». Anh H. Reynolds, https://anhreynolds.com/. visited 17 mars 2022.

[20] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, Piotr Doll: Focal Loss for Dense Object Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, https://arxiv.org/pdf/1708.02002.pdf

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, https://arxiv.org/pdf/1512.03385.pdf

[22] Ji, Qingge & Huang, Jie & He, Wenjie & Sun, Yankui. (2019). Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images. Algorithms. 12. 51. 10.3390/a12030051.

[23] « Intersection over Union (IoU) for Object Detection ». PyImageSearch, 7 november 2016, https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

[24] « Détection d'objets SSD: Détecteur MultiBox Single Shot pour un traitement en temps réel ». ICHI.PRO, https://ichi.pro/fr/detection-d-objets-ssd-detecteur-multibox-single-shot-pour-un-traitement-en-temps-reel-171355751058950. Seen 15 march 2022.

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie, Feature Pyramid Networks for Object Detection, https://arxiv.org/abs/1612.03144

[26] Bozinovski, S., Fulgosi, A.: The influence of pattern similarity and transfer learning upon the training of a base perceptron B2." Proceedings of Symposium Informatica 1976.

[27] Wada, K. Labelme: Image Polygonal Annotation with Python [Computer software]. https://doi.org/10.5281/zenodo.5711226

[28] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Schuster, M., Monga, R., Moore, S., Murray, D., Olah, C., Shlens, J., Steiner, B.,

Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2015). TensorFlow, Large-scale machine learning on heterogeneous systems [Computer software]. https://doi.org/10.5281/zenodo.4724125

[29] Coco data set (Common Objects in Context), https://cocodataset.org, visited March 2020.

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg: SSD: Single Shot MultiBox Detector, ECCV 2016, https://arxiv.org/pdf/1512.02325.pdf

.

Method for Dysgraphia Disorder Detection using Convolutional Neural Network

Juraj Skunda
Faculty of Electrical Engineering
and Information Technology,
Slovak University of Technology
in Bratislava
Ilkovičova 3
812 19, Bratislava, Slovakia
juraj.skunda@stuba.sk

Boris Nerusil
Faculty of Electrical Engineering
and Information Technology,
Slovak University of Technology
in Bratislava
Ilkovičova 3
812 19, Bratislava, Slovakia
boris.nerusil@stuba.sk

Jaroslav Polec
Faculty of Electrical Engineering
and Information Technology,
Slovak University of Technology
in Bratislava
Ilkovičova 3
812 19, Bratislava, Slovakia
jaroslav.polec@stuba.sk

ABSTRACT

This paper describes a method for dysgraphia disorder detection based on the classification of handwritten text. In the experiment we have verified proposed approach based on the conventional signal theory. Input data consists of the handwritten text by dysgraphia diagnosed children. Techniques for early dysgraphia detection could be applied in the schools to detect children with a possible diagnosis of dysgraphia and early intervention could improve their lives.

The main goal of research is to develop a tool based on a machine learning for schools to diagnose dyslexia and dysgraphia. An experiment was performed on the dataset of 120 children in the school age (63 normally developing and 57 dysgraphia diagnosed). The main advantage is the simple algorithm for preprocessing of the raw data. Then was designed simple 3-layers convolutional neural network for classification of data. On the test data, our model reached accuracy 79.7%.

Keywords

Dysgraphia, Convolutional Neural Network, Machine Learning, Spectrum

1. INTRODUCTION

Dysgraphia is a learning disability characterized by problems with writing. It is estimated that 4 – 20 percent of the population has issues connected to writing. It is a neurological disorder that can affect children or adults. It can manifest at different ages with different symptoms. However, it usually occurs when children learn to write. In addition to writing hard-to-read words, people with dysgraphia tend to use the wrong word for what they are trying to communicate. Individuals who can potentially be diagnosed with dysgraphia will exhibit certain symptoms, such as irregular or slow writing; difficulty moving their hands across the writing surface; manipulation of the writing instrument is often incorrect; inadequate body positions when writing; or excessive leaning over the written text [DBC20, MFM16]. Moreover, typical symptoms may include omitting words, watching the hands when writing, inappropriate letter spacing and sizing, difficulties to take notes at work or school, avoiding tasks including drawing or writing etc.

The cause of dysgraphia is not always known. One of the most common causes is the presence of neurological disorders in the frontal lobe, which is associated with reading and writing. In adult individuals, it can sometimes manifest, for example, as a result of a traumatic event or in association with another cognitive disorder (e.g. Parkinson's disease) [KSG17, NiF11]. If dysgraphia is left untreated, it can cause poor performance at school or work, mental issues, low self-esteem and social contacts.

Dysgraphia is often associated with dyslexia - disorder characterized by reading below the expected level for their age, but these disorders are usually associated and does not always occur together. However, dyslexia often co-occurs with other problems, whether it is other specific learning disabilities such as dysgraphia (writing disorder), dysorthography (spelling disorder), dyscalculia (mathematical learning disorder) or attention deficit disorder (ADD/ADHD) [HV20]. In some case problems associated with dysgraphia are beginning affect older children. They are writing letters that are out of place, or that do not have the correct ratio of letter sizes. In the final form, the written text looks chaotic, they do not follow each other, do not stick line, have poor spacing within words or spacing between words. They do not follow the boundaries between words in the

https://www.doi.org/10.24132/CSRN.3201.19

writing, join words together or divide them illogically. Also, as in dyslexia, confuse similar letters for example 'm' and 'n', 'b' and 'd' or numerals 7 and 4 [CC16].

The research on methods for dysgraphia detection based on the machine learning started developing in the last decade, since quality graphic tablets (WACOM, XP-Pen) have become more available and widely used. They allowed to obtain more data and features for algorithms.

Application of machine learning algorithms shown promising results in the field of diagnostics tools. For example there are experiments with for diagnosis of: mild cognitive impairment (MCI) [GML21], autism spectrum disorder (ASD) [YHE18], schizophrenia [KBA21], bipolar disorder [SCC21] or obsessive-compulsive disorder [HMD13]. Methods are using various input data from functional magnetic resonance imaging (fMRI), electroencephalograph (EEG), eye-trackers and the others.

In the first part of the research paper we deal with the recent trends in methods for dysgraphia detection. Then is described our proposed approach used for data classification and the experiment, where data was classified into two groups – normally developing and diagnosed. Finally, we discuss achieved results and define future work.

2. RELATED WORK

This section mentions some important methods for dysgraphia detection based on handwritten data. In [DD20] authors verified classification methods on various features, like velocity, acceleration, rate of change of acceleration, total writing time and so on. Dataset is same as in our experiment. For classification of the data were used AdaBoost classifier, support vector machine (SVM) and random forest algorithm. AdaBoost algorithm outperformed the others and the reached accuracy 79.5% (STD ±3).

Paper [DDG21] describes experiment, where authors tested different classifiers to find the most suitable for this sort of data. A dataset consists of 580 children, where 122 are dysgraphia diagnosed. The conditions during handwritten text were the same for all children and graphic tablets by WACOM were used. Authors analyzed features and chosen 10 optimal according to Fisher criterion, for example record time, total writing time or number of stops. To classification were used different models, like - Long short-term memory (LTSM), Decision Tree, Random Forest or SVM with different kernels. The highest balanced accuracy achieved SVM with linear kernel and SVM with RBF (Radial Basis Function) – 77%.

Authors in [RS20] compared differences between dyslexic and dysgraphic children and in experiment analysis of pictures of a handwritten manuscript and audio files, to create a classification model. Authors applied machine learning algorithms (Naïve Bayes, Logistic Regression and Random Forest) on each disorder – dyslexia and dysgraphia. Dataset of handwritten text is composed of 1481 pictures (198 of them are diagnosed with dysgraphia). With 10-fold cross validation, the best accuracy achieved Random Forest as 96.2% (STD ±2.7).

Experiment described in [KNH19] was aimed on the design of the tool for dyslexia, dysgraphia and dyscalculia (math learning disability characterized by issues with solving tasks and perform other basic math skills). Authors used tool that implements gamified environment to interact with children. There was used convolutional neural network and average precision for dysgraphia detection is 88%.

Dysgraphia is also associated with the other cognitive disorders. Authors in [MMG18] analyzed handwritten text and compared groups of 36 healthy and 33 Parkinson’s disease diagnosed individuals. Data are based on the 9 handwritten tasks for each participant and during acquisition were used digitizing tablets (like WACOM Intous 4M). For classification authors used binary XGBoost (decision-tree-based) model. Authors analyzed different features and the best approach was conventional, where they selected horizontal velocity (median) of the sentence, with the highest classification accuracy 97.14% (STD ±5.71).

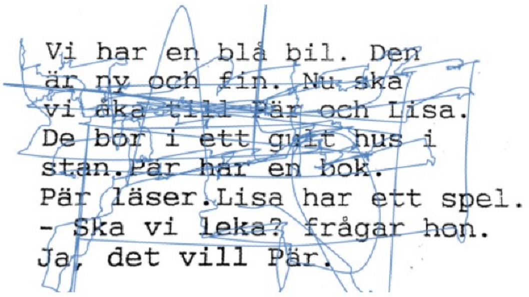
3. PROPOSED APPROACH

This section describes our method for classification of handwriting data of tested subjects. We previously proposed method used for dyslexia diagnosis based on the classification of eye-tracking data, described in [NPS21] and [NP21].

Most methods of classifying dyslexia are based on the assumption that subjects with confirmed dyslexia read significantly slower than subjects without dyslexia. However, in the case of dysgraphia, the assumption of slow writing is not so much used. The [NPS21] method eliminates the effect of time on the classification of subjects with dyslexia, so it could also be used as a basis for the classification of subjects with dysgraphia.

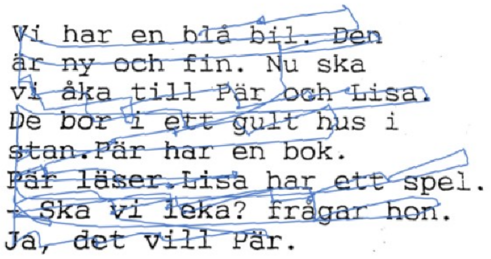
In the case of dyslexia, the eye movements of the subjects were recorded. Figures 1 and 2 shows a typical result of such eye movement in a subject with dyslexia compared to a subject without a disorder.

In the case of the examination of dysgraphia, the coordinates of the pen movement on the tablet were taken. Figures 3 and 4 shows a typical result of such pen movement in a subject with dysgraphia compared to a subject without a disorder.



Vi har en blå bil. Den
är ny och fin. Nu ska
vi åka till Pär och Lisa.
De bor i ett gult hus i
stan.Pär har en bok.
Pär läser.Lisa har ett spel.
- Ska vi leka? frågar hon.
Ja, det vill Pär.

Figure 1. Eye movements of high risk (dyslectic) individual during reading.



Vi har en blå bil. Den
är ny och fin. Nu ska
vi åka till Pär och Lisa.
De bor i ett gult hus i
stan.Pär har en bok.
Pär läser.Lisa har ett spel.
- Ska vi leka? frågar hon.
Ja, det vill Pär.

Figure 2. Eye movements of low risk (healthy) individual during reading.

The results shown in figure 1, 2 and 3, 4 have significant common features. They differ in that the writers do not have the same (or significantly similar) initial coordinates. They do not have to have a similar spatial resolution. This is due to the unequal font size. However, it does not affect the classification of dysgraphia. The use of DFT properties eliminates both of these problems, which is also solved by the [NPS21] method. It is important that all sequences of coordinates are the same length. The [NPS21] interpolation method solves this.

The method [NPS21] originally designed to classify dyslexia was applied to the unprocessed coordinates obtained by scanning the position of the pen during writing on the tablet of selected phrases by subjects [DD20].

The signals were first interpolated to the maximum length according to the slowest writing subject. This was used to remove irrelevant time information from the signals. The signal was interpolated using DCT3. The sequences thus obtained were used to calculate the magnitude spectrum DFT. The magnitude spectrum eliminates the spatial shift of the beginning of writing, to remove redundant information is used decorrelation, which very well and concentrates the most important shape properties of the written text into a smaller number of energetically important spectral components, similarly to text classification [PBV16].

Such preprocessing was done one-dimensionally in the x-direction and one-dimensionally in the y-direction. The preprocessed signals then enter the classifier. A convolutional neural network was used as a classifier in the experiment. Entry into it were pairs of preprocessed vectors x and y.

$$DCT3_{U_{k,n}} = \sqrt{\frac{2}{N}}\, c_n \cos\left(\frac{\pi((2k+1)n)}{2N}\right) \tag{1}$$

k, *n* = 0,1 , …, *N*-1;

$$DFT_{U_{k,n}} = \sqrt{\frac{1}{N}}\, exp\left(-j\frac{2\pi kn}{N}\right) \tag{2}$$

k, *n* = 0,1 , …, *N*-1;

where *k* represents the order in the spectrum and the *n* order in time.

4. EXPERIMENT

This section describes dataset we have used for design of a method for dysgraphia detection based on handwritten data. The next part of the section presents classification algorithm.

Participants

Dataset consists of 120 children (40 female and 80 male) in the school age between 8 and 15 years. There is 63 normally developing (healthy) individuals and 57 dysgraphia diagnosed individuals. In the figures 3 and 4 the examples of the handwritten text are shown.

For data acquisition was used WACOM Intuos Pro Large tablet. During collection of the data were subjects requested to write: letter "l" at normal and fast speeds, syllable "le" at normal and fast speeds, simple word "leto" (summer), pseudoword "lamoken", difficult word "hračkárstvo" (toy-shop), sentence "V lete bude teplo a sucho" (The weather in summer is hot and dry) [DD20].

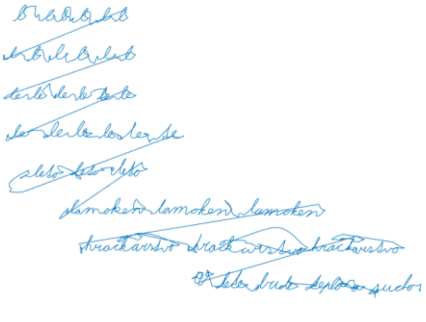


Figure 3. Handwritten text by subject 39 (normally developing).

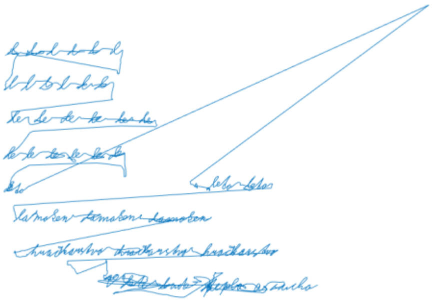


Figure 4. Handwritten text by subject 114 (dysgraphia diagnosed).

Data are represented as x- and y- coordinates. Then were preprocessed as it is described in the previous section.

Classification

To classify the subjects into two groups we have used convolutional neutral network (ConvNet). ConvNets belong to a class of artificial neural network in they have become dominant in computer vision tasks.

Convolutional neural networks are feedforward neural networks - data input to individual nodes of the network is in one direction only. The architecture is exemplified by the visual cortex in the brain (the part that processes visual information), which consists of alternating layers of simple and complex cells. There are currently several well-known network architectures, such as AlexNet, GoogleNet, LeNet, VGG-16, ResNet-50, Xception, Inception, Inception-v4, Inception-ResNet-V2, or ResNeXt-50. ConvNets generally consist of convolutional and pooling layers that are grouped into modules. The individual modules are then stacked in sequence and can thus form a deep neural network (DNN) [KRS18, RW17].

ConvNets uses relatively small preprocessing compared to other image classification algorithms. This means that the network learns the filters that were created manually in traditional algorithms. This independence from prior knowledge and human effort in designing functions is a major advantage [KRS18].

For purpose of our experiment we have used tools in MATLAB, to create a simple neural network for classification, which especially suited for image recognition. Our steps were as follows:

- loaded dataset of images;
- defined the network architecture;
- specification of training options;
- training of the network;
- prediction of the labels of new data and calculate the classification accuracy.

We have used 3-layers network. Model with this number of layers was also used in [NPS21], but we have tried to use various parameters of network, to find optimal network. The first layer of a model has filter size 3x3 with number of 8. The next convolutional layer has 16 filters, also with size of 3x3. The third convolutional neural layer uses 32 filters with size as previous layers. Stochastic Gradient Descent with Momentum (SGDM) is used to learn the convolutional neural network. The initial learning speed for SGDM is 0.01. After each iteration, the training data is rearranged. The maximum number of training epochs is 12. The used hardware is laptop with a configuration: CPU Intel Core i5-1135G7, GPU NVIDIA GeForce MX450 (2 GB) and 16 GB RAM and training time was approximately 6.5 minutes.

Net Name	Structures [kernel sizes;No. of kernels]
CNN3	[3x3, 3x3, 3x3; 8, 16, 32],

Table 1. Network Structures

10-fold cross-validation was used during training. Data was split into ratio 80-10-10, training, validation and testing, respectively. A problem is also small dataset. Optimal machine learning algorithm should be trained and tested on much more larger amount of data.

5. RESULTS

We have achieved average accuracy 79.7% (STD ±2.9), compared to method [DD20] where authors achieved best accuracy 79.5% (STD ±3).

True positive rate (TPR) and true negative rate (TNR) are calculated as follows:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{4}$$

Our model achieved TPR is 76% (STD ±4.9) and TNR is 80.6% (STD ±3). The results show that model is stable and standard deviation does not have high values. In the next section we will discuss the results and next step in research.

	3-layer CNN	Ada-Boost	SVM	RF
TPR [%]	76% ±4.9%	**79.7% ±5%**	74.5% ±4%	71.4% ±3%
TNR [%]	80.6% ±3%	76.7% ±2%	82.4% ±4%	**83.3% ±2%**
ACC [%]	**79.7% ±2.9%**	79.5%± 3%	78.8% ±2%	77.6% ±1%

Table 2. Result for the tested CNN structures and the reference methods

6. CONCLUSIONS AND THE FUTURE WORK

We have presented in this paper an approach for dysgraphia disorder detection. Method achieved comparable results to the other papers.

The next steps in our research will be improvement of the accuracy, for example by other approach to preprocessing of data or improvement of convolutional neural network model. The other experiment could verify our approach on the dataset of the written text in the different language.

Our goal is to create a dataset with collected data of handwritten text and recorded eye movements of the subjects to develop a method for detection of writing or reading disorders. This tool could be implemented in the schools and provide improvement in an early diagnosis of these disorders, with an impact on quality of children's live.

7. ACKNOWLEDGMENTS

We would like to thank the authors for providing the dataset and information regarding to the previous research Peter Drotár & Marek Dobeš from Department of Computers and Informatics, Technical University of Košice.

This study was supported by the following funds: Mladý výskumník - WRITE2021 research grant and KEGA 015STU- 4/2021.

8. REFERENCES

[CC16] Cardoso, M. H., & Capellini, S. A. (2016). Identification and characterization of dysgraphia in students with learning difficulties and learning disorders. Distúrb Comun, 28(1), 27-37.

[CIS19] Crespo Y, Ibañez A, Soriano MF, Iglesias S, Aznarte JI (2019) Handwriting movements for assessment of motor symptoms in schizophrenia spectrum disorders and bipolar disorder. PLoS ONE 14(3): e0213657. https://doi.org/10.1371/journal.pone.0213657

[DBC20] Dimauro, G., Bevilacqua, V., Colizzi, L., & Di Pierro, D. (2020). TestGraphia, a Software System for the Early Diagnosis of Dysgraphia. IEEE Access, 8, 19564-19575.

[DD20] Drotár, P., Dobeš, M. Dysgraphia detection through machine learning. Sci Rep 10, 21541 (2020). https://doi.org/10.1038/s41598-020-78611-9

[DDG21] Deschamps, L., Devillaine, L., Gaffet, C., Lambert, R., Aloui, S., Boutet, J., Brault, V., & Labyt, E., Jolly, C. (2021). Development of a Pre-Diagnosis Tool Based on Machine Learning Algorithms on the BHK Test to Improve the Diagnosis of Dysgraphia. Advances in Artificial Intelligence and Machine Learning. 01. 10.54364/AAIML.2021.1108.

[DH16] Döhla D, Heim S. Developmental Dyslexia and Dysgraphia: What can We Learn from the One About the Other?. Front Psychol. 2016;6:2045. Published 2016 Jan 26. doi:10.3389/fpsyg.2015.02045

[GML21] Groznik, V., Možina, M., Lazar, T., Georgiev, D., Sadikov, A. "Gaze Behaviour During Reading as a Predictor of Mild Cognitive Impairment," 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), 2021, pp. 1-4, doi: 10.1109/BHI50953.2021.9508586.

[HMD13] Hoexter, M. Q., Miguel, E. C., Diniz, J. B., Shavitt, R. G., Busatto, G. F., & Sato, J. R. (2013). Predicting obsessive-compulsive disorder severity combining neuroimaging and machine learning methods. Journal of affective disorders, 150(3), 1213–1216. https://doi.org/10.1016/j.jad.2013.05.041

[HV20] Hamdioui, S., Vaivre-Douret, L. "Clinical Markers of Dysgraphia According to Intellectual Quotient in Children with Developmental Coordination Disorder". Journal of Psychiatry and Psychiatric Disorders 4 (2020): 366-382.

[KBA21] Khare, S. K., Bajaj, V., & Acharya, U. R. (2021). SPWVD-CNN for Automated Detection of Schizophrenia Patients Using EEG Signals. IEEE Transactions on Instrumentation and Measurement, 70, 1–9. doi:10.1109/tim.2021.3070608

[KNH19] Kariyawasam, R., Nadeeshani, M., Hamid, T., Subasinghe, I., Samarasinghe P., Ratnayake, P. "Pubudu: Deep Learning Based Screening And Intervention of Dyslexia, Dysgraphia And Dyscalculia," 2019 14th Conference on Industrial and Information Systems (ICIIS), 2019, pp. 476-481, doi: 10.1109/ICIIS47346.2019.9063301.

[KSG17] Kurniawan, D.A., Sihwi, S.W., & Gunarhadi (2017). An expert system for diagnosing dysgraphia. 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 468-472.

[KRS18] Khan S., Rahmani H., Ali Shah S.A, Bennamoun M., Medioni G., Dickinson S. A Guide to Convolutional Neural Networks for Computer Vision , Morgan & Claypool, 2018, doi: 10.2200/S00822ED1V01Y201712COV015.

[MFM16] Mekyska, J., Faundez-Zanuy, M., Mzourek, Z., Galáž, Z., Smekal, Z., Rosenblum, S. (2016). Identification and Rating of Developmental Dysgraphia by Handwriting Analysis. IEEE Transactions on Human-Machine Systems. 47. 10.1109/THMS.2016.2586605.

[MMG18] Mucha, J.; Mekyska, J.; Galaz, Z.; Faundez-Zanuy, M.; Lopez-de-Ipina, K.; Zvoncak, V.; Kiska, T.; Smekal, Z.; Brabenec, L.; Rektorova, I. Identification and Monitoring of Parkinson's Disease Dysgraphia Based on Fractional-Order Derivatives of Online Handwriting. Appl. Sci. 2018, 8, 2566. https://doi.org/10.3390/app8122566

[NiF11] Nicolson, R. I., & Fawcett, A. J. (2011). Dyslexia, dysgraphia, procedural learning and the cerebellum. Cortex; a journal devoted to the study of the nervous system and behavior, 47(1), 117–127. https://doi.org/10.1016/j.cortex.2009.08.016

[NPS21] Nerušil, B., Polec, J., Škunda, J. et al. Eye tracking based dyslexia detection using a holistic approach. Sci Rep 11, 15687 (2021). https://doi.org/10.1038/s41598-021-95275-1

[NP21] Nerusil, B., Polec, J., Skunda, J. (2021). Detection of Dyslexia Using Eye Tracking. 202-205. 10.1109/SPSympo51155.2020.9593313.

[RW17] Rawat W. and Wang Z., Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review, Neural Computation 2017 29:9, 2352-2449, doi: 10.1162/neco_a_00990

[PBV16] Polec, J., Benesova, W., Vargic, R., Ilčíková, I., & Csóka, T. (2016). Texture feature extraction using an orthogonal transform of arbitrarily shaped image regions. Journal of Electronic Imaging. 25. 061413. 10.1117/1.JEI.25.6.061413.

[RS20] Richard, G., & Serrurier, M. (2020). Dyslexia and Dysgraphia prediction: A new machine learning approach. ArXiv, abs/2005.06401.

[SCC21] Sawalha, J., Cao, L., Chen, J., Selvitella, A., Liu, Y., Yang, C., Li, X., Zhang, X., Sun, J., Zhang, Y., Zhao, L., Cui, L., Zhang, Y., Sui, J., Greiner, R., Li, X. M., Greenshaw, A., Li, T., & Cao, B. (2021). Individualized identification of first-episode bipolar disorder using machine learning and cognitive tests. Journal of affective disorders, 282, 662–668. https://doi.org/10.1016/j.jad.2020.12.046

[YHE18] Yaneva V., Ha L. A., Eraslan S., Yesilada Y., and Mitkov R. 2018. Detecting Autism Based on Eye-Tracking Data from Web Searching Tasks. In Proceedings of the Internet of Accessible Things (W4A '18). Association for Computing Machinery, New York, NY, USA, Article 16, 1–10. :https://doi.org/10.1145/3192714.3192819

Empirical verification of the suggested hyperparameters for data augmentation using the fast.ai library

Wojciech Oronowicz-Jaśkowiak
Faculty of Computer Science of the Polish-Japanese Academy of Information Technology
Koszykowa 86
Poland, Warsaw 02-008
oronowiczjaskowiak@pjwstk.edu.pl

Piotr Wasilewski
Faculty of Mathematics, Informatics and Mechanics of the University of Warsaw
Banacha 2
Poland, Warsaw 02-008
piotr@mimuw.edu.pl

Mirosław Kowaluk
Faculty of Mathematics, Informatics and Mechanics of the University of Warsaw
Banacha 2
Poland, Warsaw 02-008
kowaluk@mimuw.edu.pl

ABSTRACT

Data augmentation consists in adding slightly modified copies of the existing data to the training set, which increases the total amount of data and generally results in better results obtained by machine learning algorithms. The fast.ai library has some predefined values for data augmentation hyperparameters for visual data. It is claimed that these predefined parameters are to be the best for most data types, however, no empirical support for this statement has been provided. The aim of this research is to determine whether the suggested hyperparameter values for data augmentation in the fast.ai library are indeed optimal for the highest accuracy for image classification tasks. In order to answer this question, a detailed research was conducted, consisting of a series of experiments for subsequent data augmentation tools (rotation, magnification, contrast change, etc.). Three variables were modified for each tool: 1. maximal and minimal value of transformation (depending on the transformation type), 2. probability of the transformation, 3. padding behaviour. The results of the presented research lead to the conclusion that the suggested values of data augmentation implemented in the fast.ai library provides the good parameters of the model aimed at differentiating male and female faces, however in case of that classification slightly different parameters could be taken into consideration. The results are published in open-source repository (*Open Science Framework*, DOI:10.17605/OSF.IO/38UJG).

Keywords
Machine learning, supervised learning, computer vision

1. INTRODUCTION

Currently, machine learning uses many libraries to support the process of building advanced models. Libraries and their extensions enable the construction of known algorithms without the need to develop them from the very beginning and ensure flexibility of calculations. Fast.ai library used in this research will be characterized – fast.ai library, for which parameters the experiments were carried out.

The fast.ai library is an open source programming library that, like PyTorch, uses the Python [How00a] programming language. It is highly integrated with the PyTorch library. This library gained publicity at the end of 2018, when a group of students using fast.ai manager to defeat commercial teams from the Google and Intel teams in the CIFAR-10 classification competition. The results of this competition showed that even smaller development teams are still able to present better technological solutions than large corporations with an incomparably larger budget. The fast.ai library is mainly based on PyTorch library, but there are some significant differences compared to it. Fast.ai library allows to easily use very advanced techniques supporting machine learning, such as searching for the optimal *learning rate* range that can be used in the network training process. The documentation for fast.ai library, version 2.0, is extensive and provides a number of examples to illustrate the use of the built-in functions. The advantage of this library is therefore the possibility of quick use of functionalities supporting the process of training neural networks, which from the programmer's perspective are implemented as one line of code, but in fact are much more complex.

2. RELATED WORK

The fast.ai library has been used successively in many research. Recently, the author of the library, Jeremy Howard, proposed the ULMFiT (*Universal Language Model Fine-Tuning*) model based on the

https://www.doi.org/10.24132/CSRN.3201.20

fast.ai library, which allowed to improve the task of text classification on the example of several popular training data sets [How01a]. The technique of audio spectrogram language identification is also presented (LIFAS), using spectrograms generated from audio signals, and then transmitted as input data to the convolutional neural network, which in turn allows for language identification. The technique presented by the authors, thanks to the use of the fast.ai library, is transparent and seems to be easily available for replication [Rev00a]. An interesting article that has been published recently is the proposed classification of malware on the basis of image analysis [Bho00a]. Although the results presented by the authors showed that deep neural networks are not better at this task compared to simpler methods, such as the k-means method, the results related to neural networks may be characterized by a better generalization ability. In both cases (the use of neural networks as well as the k-means method), an analysis was performed using the fast.ai library.

The presented variety of applications shows the great possibilities and flexibility of the fast.ai library. The creators themselves encourage even users who are just starting their adventure with machine learning to experiment on data based on domain knowledge. The idea behind the creators was a kind of democratization of advanced machine learning methods, translating into the development of knowledge in many areas.

Data augmentation consists in introducing such modifications to photos that allow them to be enriched with features that they did not have before [Bho00a]. Thanks to this, important effects can be obtained from the point of view of the parameters of machine learning models. Increasing the amount of training data by introducing slightly modified copies of them helps to counteract rapid overtraining of neural networks, especially when one have a relatively small set of training data. It has been repeatedly indicated that the use of various data augmentation techniques may allow for significantly higher parameters of machine learning models [Zho00a, Won00a, Mas00a].

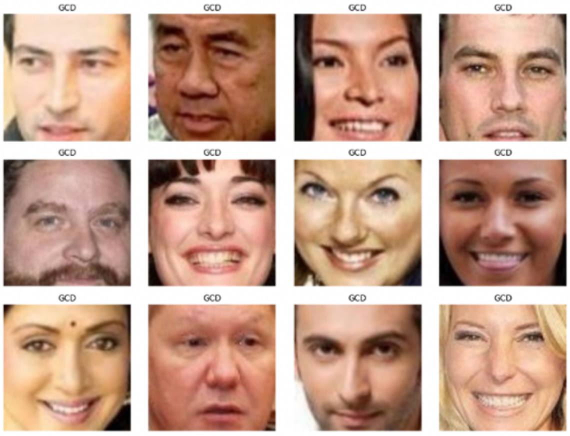
The fast.ai library allows to use several predefined transformations of training data, data augmentation techniques. Among the transformations that are made automatically by calling the *get_transforms* class, the following are distinguished: reflection of the photo in relation to the x and y axes, rotation of the photo by a random angle, zooming, brightening and *warp*.

The aim of the study was to verify the suggested values [How00a] of data augmentation transformations for the fast.ai library. In order to select the best hyperparameters, a total of 26 studies were conducted, which were related to individual techniques of data augmentation, and an initial study related to the selection of the appropriate architecture of the neural network.

3. METHOD

The *Gender Classification Dataset* [Cha00a] was used for the research. This database includes photos of human faces – women and men – of different ages and ethnic origins. These photos are a changed version of the material from another IMDB-WIKI [Cha00a] data set, in such a way that only the images of the face were separated from the entire human figure. The database created in this way contains photos in .jpg format with a resolution of 80x100 px., assigned to one of two categories – photos of female and male faces. In each case, the fast.ai library was used to train the neural networks. The training material consisted of 16,000 photos, 8,000 female faces and 8,000 male faces, respectively. The validation material consisted of 4,000 photos (with a similar division). Two classes were defined – defined as men and women. The training and validation material was selected each time at random according to the relevant research conditions. A total of 100 epochs (network training cycles) were run in ten stages. Each stage consisted of ten eras. If the network overtraining was observed at the end of a given stage, the stage that preceded the overtraining was selected. The Figure 1 shows the examples of training set [Cha00a].

Figure 1. Examples of images used in the training set.



To validate our study, we used four different datasets: CIFAR-10 [Kri00a], Intel Image Classification [Ban00a], sexACT 0.5 [Oro00a] and MNIST [LeC00a]. The first dataset consists of 60.000 color images categorized in 10 classes (e.g. bird, cat, dog, frog). The second dataset consists of 25.000 color images categorized in 6 classes (e.g. buildings, sea, street). The third dataset consists of

11.600 color images categorized in 11 classes presenting human sexual activity (e.g. BDSM). The last dataset consists of 60.000 images categorized in 10 classes (numbers ranged from 0 to 9).

The research that was performed included conducting a series of experiments in which three variables were modified that affect the nature of the applied data augmentation: maximum and minimum size of transformations, probability of making this type of transformations, padding.

Before conducting the appropriate series of experiments, a study was conducted to identify the architecture that would be most appropriate for the selected data type. It was decided to choose one of the few most common architectures in the scientific literature [Cha00a, Sim00a] – ResNet50, ResNet101, ResNet152, VGG16, VGG19. VGG (Visual Geometric Group) architecture is characterised by grouping convolution layers with small kernel sizes, however it is not resistant to explosion of gradients problem. ResNet is the residual neural network presented to solve problem with vanishing gradient. Using that architecture, it is possible to train network with large number of layers.

A total of 100 epochs (network training cycles) were performed for ten conditions – for five selected architectures (ResNet152, ResNet101, ResNet50, VGG19, VGG16) and for two conditions (no data augmentation and the use of suggested values for data augmentation from the fast.ai libraries). The criterion for the selection of the neural network architecture, which was used for later studies, was the maximum value of the classification accuracy on the validation material before the signs of network overtraining.

4. RESULTS

Table 1 and Table 2 show the maximum values of the classification accuracy for individual network architectures.

Table 1. Maximum values of classification accuracy without data augmentation for individual architectures before signs of neural network overtraining.

Research id	Architecture	Accuracy	F-score	Training loss	Validation loss
1.1.	ResNet152	96.40%	0,9674	0.2797	0.1174
1.2.	ResNet101	96.32%	0,9655	0.2854	0.1148
1.3.	ResNet50	96.87%	0,9899	0.2452	0.1275
1.4.	VGG16	96.42%	0,9644	0.2695	0.1246
1.5.	VGG19	96.50%	0,9670	0.2684	0.1455

Table 2. Maximum values of classification accuracy using data augmentation (fast.ai library suggested values) for individual architectures before overtraining.

id	Architecture	Accuracy	F-score	Training loss	Validation loss
2.1.	ResNet152	96.55%	0,9699	0.2721	0.1069
2.2.	ResNet101	96.95%	0,9720	0.2612	0.1183
2.3.	ResNet50	96.85%	0,9684	0.2705	0.1155
2.4.	VGG16	96.65%	0,9671	0.3129	0.1224
2.5.	VGG19	96.67%	0,9664	0.2918	0.1218

As a result of the research, the ResNet101 architecture was selected. Compared to other architecture, ResNet101 had the most stable learning process, the highest final classification accuracy was achieved, the validation loss was relatively low and the highest F-score. Six studies were conducted, where the hyperparameters appropriate for the right data transformations were modified – testing the values of transformations of the class flip_vert (true or false), flip (true or false), max_lighting (from 0.1 to 0.5), max_rotate (from 8.0 to 12.0), max_zoom (from 1.05 to 1.25). Tables 3 and 4 present the parameters of the neural network in the presence of the flip_vert and flip class transformations. Tables 5, 6 and 7 present the parameters of the neural network in the case of the best parameter from the range presented above.

Table 3. Neural network parameters in the case of the presence of the flip_vert class transformation with the method of filling missing zeroes pixels.

Research id	Data augmentation	Validation loss	Train loss	F-score	Accuracy
3.1.	No	0.1180	0.2977	0,9688	96.45%
3.2.	Yes	0.1178	0.3198	0,9649	96.45%

Table 4: Neural network parameters in the case of the presence of the *flip* class transformation with the method of filling in the missing *zeros* pixels.

Research id	Data augmentation	Validation loss	Training loss	F-score	Accuracy
4.1.	No	0.1180	0.2873	0,9677	96.75%
4.2.	Yes	0.1216	0.2858	0,9678	96.72%

Table 5: Neural network parameters in the case of the presence of the *max_ lighting = 0.30* class, with the method of filling the missing *border* pixels.

Research id	Probability	Validation loss	Training loss	F-score	Accuracy
5.1.	0.1	0.1118	0.3047	0,9653	96.62%
5.2	0.2	0.1123	0.2801	0,9655	96.67%
5.3	0.3	0.1034	0.2997	0,9669	96.67%
5.4	0.4	0.1077	0.2964	0,9688	96.65%
5.5.	0.5	0.1134	0.3038	0,9644	96.42%
5.6.	0.6	0.1121	0.2703	0,9710	96.80%
5.7.	0.7	0.1051	0.2704	0,9788	97.15%
5.8.	0.8	0.1094	0.2816	0,9710	96.67%
5.9.	0.9	0.1129	0.2906	0,9699	96.87%
5.10.	1.0	0.1117	0.2910	0,9661	96.60%

Table 6: Neural network parameters in the case of the presence of a *max_rotate = 9.0* class transformation, with the method of filling the missing *border* pixels.

Research id	Probability	Validation loss	Training loss	F-score	Accuracy
6.1.	0.1	0.1072	0.2798	0,9656	96.85%
6.2.	0.2	0.1067	0.2763	0,9721	97.00%
6.3.	0.3	0.1085	0.2951	0,9698	96.52%
6.4.	0.4	0.1092	0.2931	0,9681	96.87%
6.5.	0.5	0.1109	0.2940	0,9622	96.75%
6.6.	0.6	0.1074	0.3004	0,9678	96.72%
6.7.	0.7	0.1118	0.2884	0,9643	96.85%
6.8.	0.8	0.2671	0.0799	0,9821	97.92%
6.9.	0.9	0.3081	0.0916	0,9741	97.37%
6.10.	1.0	0.2929	0.095	0,9799	97.52%

Table 7: Neural network parameters in the case of the presence of the *max_ zoom = 1.25* class, with the method of filling the missing *border* pixels.

Research id	Probability	Validation loss	Training loss	F-score	Accuracy
7.1.	0.1	0.1111	0.2889	0,9677	96.85%
7.2.	0.2	0.1106	0.2617	0,9688	96.82%
7.3.	0.3	0.1021	0.2924	0,9689	97.17%
7.4.	0.4	0.1108	0.2979	0,9689	96.77%
7.5.	0.5	0.1050	0.2857	0,9699	97.00%
7.6.	0.6	0.1014	0.2884	0,9749	97.32%
7.7.	0.7	0.1071	0.2913	0,9613	96.87%
7.8.	0.8	0.1025	0.2880	0,9650	96.92%
7.9.	0.9	0.1050	0.2937	0,9676	97.05%
7.10.	1.0	0.1091	0.2829	0,9641	96.92%

After selecting the best parameters (flip_vert = False, do_flip = False, max_lighting = 0.3, p_ lighting = 0.70, max_rotate = 9.0, max_zoom = 1.25, p_ affine = 0.60) we conducted additional experiments aimed to compare the results using different datasets. Table 8 shows the maximum values of the classification accuracy for individual classification task.

Table 8. Neural network parameters using different datasets.

Research id	Dataset	Neural network model	Accuracy	F-score
8.1.	CIFAR-10 [Kri00a]	ResNet101, default data augmentation hyperparameters	83,68%	0,8730
8.2.		ResNet101, new data augmentation hyperparameters	83,89%	0,8392
8.3.	Intel Image Classification [Ban00a	ResNet101, default data augmentation hyperparameters	94,01%	0,9421
8.4.		ResNet101, new data augmentation hyperparameters	94,19%	0,9425
8.5.	sexACT 0.5 [Oro00a]	ResNet101, default data augmentation hyperparameters	95,45%	0,9549
8.6.		ResNet101, new data augmentation hyperparameters	95,98%	0,9621
8.7.	MNIST [LeC00a]	ResNet101, default data augmentation hyperparameters	96,02%	0,9612
8.8.		ResNet101, new data augmentation hyperparameters	95,98%	0,9610

5. DISCUSSION

In principle, each of the architectures selected would ultimately allow the differentiation between the data with high precision. There were slight differences between the architectures in terms of the amount of the loss of validation or the final accuracy of the classification. The lowest value of the validation loss in the case of non-augmented data was found in the ResNet101 architecture, while in the case of augmented data it was the ResNet152 network. The most stable course of the learning process, understood as the lack of clear differences in the validation loss and signs of network overtraining, occurred in the case of the VGG19 and VGG16 architectures.

It seems that random rotation of photos showing faces in the task of differentiating between faces of women and men does not translate into obtaining better model parameters. If so, these results would be in line with those proposed by the authors of the fast.ai library, as suggested. The random use of the “mirror image” in the task of differentiating between the faces of women and men does not translate into obtaining better parameters of the model. If so, these results would not be consistent with those proposed by the authors of the fast.ai library as suggested. The suggested value for this transformation is a logical value indicating that this transformation should be performed (*do_flip = True*), while the results of the conducted research indicate that this operation does not lead to better parameters, therefore there is no basis for its use in this set data. In the case of this class of transformations, consisting in random brightening of photos, it was found that the *max_lighting* technique significantly influenced the parameters of the model. The best results were achieved with the transformation class value set to 0.3, with the transformation probability of 0.7., the systematic use of the transformation consisting in changes in the angle of inclination of the photos in the task of differentiating between the faces of women and men translates into obtaining better model parameters. The observed results are not consistent with those proposed by the authors of the fast.ai library as suggested. In the case of this class of transformations, consisting in the random enlargement of images, it was found that the max_zoom technique significantly influenced the parameters of the model. The best results were obtained when the value of this transformation class was set to 1.25, with the transformation probability of 0.6. With such selected parameters, a low validation loss was obtained, amounting to 0.1014, and the highest accuracy of the test material classification, amounting to 97.32%.

It is worth highlighting the limitations of the study. Firstly, the presented study is limited to the images showing faces and only one dataset - *The Gender Classification Dataset* [Cha00a]. It is known that data transformation is different when it comes to train different data. We have tested new parameters using four different datasets, however the results were the best when there were faces in the photos (e.g. in sexACT dataset). This may prove that the new parameters are specific to the face classification task. Secondly, the presented differences in model parameters are slight, consequently they might be related with computational bias. Thirdly, we used only geometric transformations, but there were presented other data augmentation algorithms. Fourthly, to easier test our hypothesis we used only the basic metrics, however to draw more advanced results, different metrics such as NPV, FNR, FDR or ROC curve should be also compared.

6. CONCLUSIONS

The results of the presented research lead to the conclusion that the suggested values of data augmentation implemented in the fast.ai library provides the good parameters of the model aimed at differentiating male and female faces, however in case of that classification slightly different parameters could be taken into consideration. In the case of tests carried out on a selected set of training data:

- A flip_vert class transformation should not be applied (flip_vert = False) and the suggested value of this transformation (flip_vert = False) is consistent with what was observed.
- The flip class transformation should not be applied (do_flip = False) and the suggested value of this transformation (do_flip = True) does not match the observations.
- Transformation of the max_lighting class should be applied (max_lighting = 0.3, p_ lighting = 0.70) and the suggested value of this transformation (max_lighting = 0.2, p_ lighting = 0.75) does not match the observations.
- Transformation of the max_rotate class should be used (max_rotate = 9.0, p_ affine = 1.00) and the suggested value of this transformation (max_rotate = 10.0, p_ affine = 0.75) does not match the observations.
- Transformation of the max_zoom class should be used (max_zoom = 1.25, p_ affine = 0.60) and the suggested value of this transformation (max_zoom = 1.10, p_ affine = 0.75) does not match the observations.

7. ACKNOWLEDGMENTS

8. REFERENCES

[Bho00a] Bhodia, N., et al. Transfer learning for image-based malware classification. ArXiv preprint arXiv:1903.11551, 2019.

[Ban00a] Banslal, P. Intel Image Classification. Accessed 28.05.2022 from www.kaggle.com.

[Cha00a] Chauhan, A. Gender Classification Dataset. Accessed 22.12.2021 from www.kaggle.com/cashutosh/gender-classification-dataset.

[He00a] He, K., et al. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[How00a] Howard, J. Fastai V2 documentation. Accessed 22.12.2021 from http://docs.fast.ai.

https://www.doi.org/10.24132/CSRN.3201.20

[How01a] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. Accessed 22.12. 2021 from https://arxiv.org/abs/1801.06146; 16.04.2021.

[Kri00a] Krizhevsky, A, et. al. Learning multiple layers of features from tiny images. Accessed 28.05.2022 from: https://www.cs.toronto.edu.

[LeC00a] LeCun, Y. The MNIST database of handwritten digits. Accessed 28.05.2022 from www.yann.lecun. com/exdb/mnist.

[Mas00a] Masi, I., et al. Do we really need to collect millions of faces for effective face recognition?" European conference on computer vision, Springer, Cham, 2016.

[Oro00a] Oronowicz-Jaśkowiak, W, et. al. Binary classification of pornographic and non-pornographic materials using the sAI 0.4 model and the modified database. *Advances in Psychiatry and Neurology*, 29(2), pp.108-119, 2020.

[Rev00a] Revay, S., Teschke, M. Multiclass language identification using deep learning on spectral images of audio signals. ArXiv preprint arXiv:1905.04348, 2019.

[Sho00a] Shorten, C., Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *Journal of Big Data,* 6.1, pp. 1-48, 2019.

[Sim00a] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. ArXiv preprint arXiv:1409.1556, 2014.

[Won00a] Wong, S.C., et al. Understanding data augmentation for classification: when to warp? 2016 international conference on digital image computing: techniques and applications (DICTA), IEEE, 2016.

[Zho00a] Zhong, Z, et al. Random erasing data augmentation Proceedings of the AAAI Conference on Artificial Intelligence, 34, 2020.

Estimation of mitral valve hinge point coordinates - deep neural net for echocardiogram segmentation

Christian Schmidt
Westfälische Hochschule –
University of Applied Sciences
Medical Engineering Laboratory
Neidenburger Str. 43
45897 Gelsenkirchen, Germany
christian.schmidt@w-hs.de

Heinrich Martin Overhoff
Westfälische Hochschule –
University of Applied Sciences
Medical Engineering Laboratory
Neidenburger Str. 43
45897 Gelsenkirchen, Germany
heinrich-martin.overhoff@w-hs.de

ABSTRACT

Cardiac image segmentation is a powerful tool in regard to diagnostics and treatment of cardiovascular diseases. Purely feature-based detection of anatomical structures like the mitral valve is a laborious task due to specifically required feature engineering and is especially challenging in echocardiograms, because of their inherently low contrast and blurry boundaries between some anatomical structures. With the publication of further annotated medical datasets and the increase in GPU processing power, deep learning-based methods in medical image segmentation became more feasible in the past years. We propose a fully automatic detection method for mitral valve hinge points, which uses a U-Net based deep neural net to segment cardiac chambers in echocardiograms in a first step, and subsequently extracts the mitral valve hinge points from the resulting segmentations in a second step. Results measured with this automatic detection method were compared to reference coordinate values, which with median absolute hinge point coordinate errors of 1.35 mm for the *x*- (15-85 percentile range: [0.3 mm; 3.15 mm]) and 0.75 mm for the *y*- coordinate (15-85 percentile range: [0.15 mm; 1.88 mm]).

Keywords

Medical image segmentation, echocardiography, deep learning, U-Net, mitral valve

1. INTRODUCTION

According to the World Health Organization (WHO), 17.9 million people died from cardiovascular diseases (CVDs) in 2019, which is 32% of all global deaths. Advances in medical imaging significantly improved the process of diagnostics and treatment of CVDs over the past years, with cardiac image segmentation playing an important role.

Cardiac image segmentation is the process of partitioning an image by assigning a label to each pixel of the image in such a way, that pixels of a certain anatomical structure share the same label. Anatomical and functional parameters such as left ventricle (LV) volume, left atrium (LA) volume, ejection fraction and mitral valve (MV) dimensions can be determined using segmented images.

Direct segmentation of the two mitral valve leaflets (MVLs) with purely feature-based algorithms often fails, because of low contrast in echocardiograms or lack of visualization of both MVLs at the same time, which is common in clinical settings. Therefore, we propose to assess MV hinge point coordinates by using deep learning (DL) segmentations of the LV and LA. In 2019 Leclerc et al. [Lec19] published the Cardiac Acquisitions for Multi-structure Ultrasound Segmentation (CAMUS) dataset in conjunction with an image segmentation challenge https://www.creatis.insa-lyon.fr/Challenge/camus/. This is to our knowledge the first large-scale, publicly available transthoracic echocardiography (TTE) dataset, which includes ground truth segmentations of the LV and LA.

In this paper, we use the CAMUS dataset (Fig. 1, Fig. 2) to segment the LV and LA in apical four (a4c) and two-chamber views (a2c) in a first step, and in a second step extract the mitral valve diameter and hinge point coordinates from the resulting segmentation.

To the best of our knowledge, this is the first attempt of DL-based mitral valve measurement in

https://www.doi.org/10.24132/CSRN.3201.21

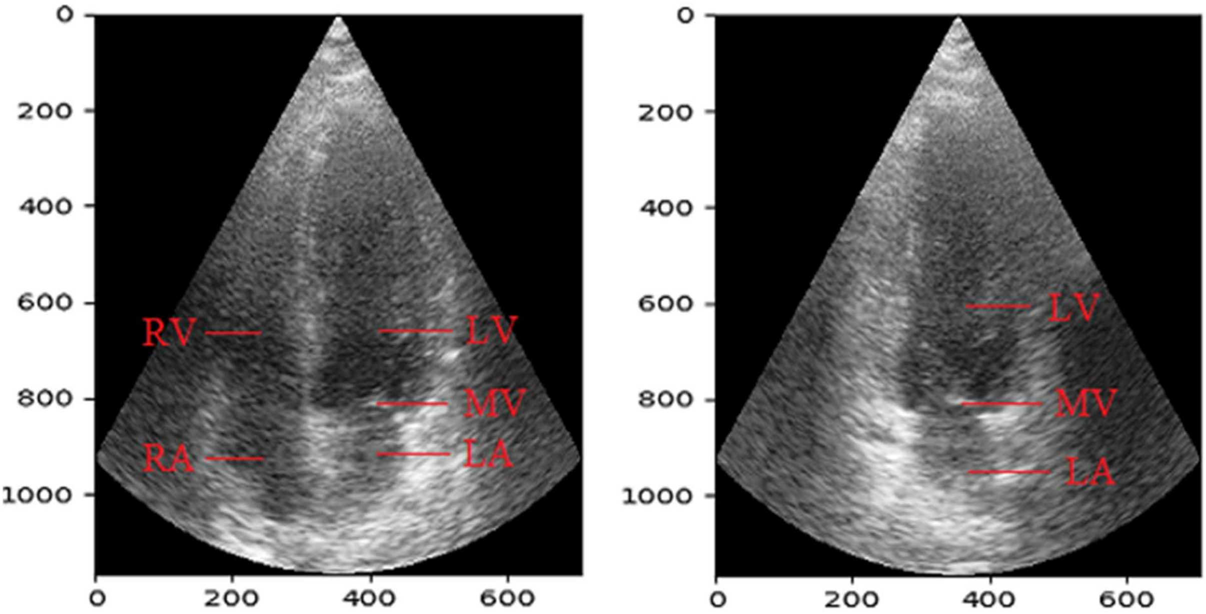


Figure 1. Exemplary labelled TTE of the CAMUS dataset in a4c (left) and a2c view (right), showing the ventricles (LV and RV), atria (LA and RA) and the mitral valve (MV).

transthoracic four- and two-chamber view echocardiograms.

2. RELATED WORKS

Conventional machine learning techniques, e.g., active shape and atlas-based models [Okt14; Tav13], showed good performance in cardiac image segmentation, but rely on user-based, manual feature extraction.

In recent years, with the increased availability of high-performance GPUs and more access to image training data, deep learning (DL) models, which automatically learn features from image data, have outperformed conventional machine learning techniques.

These DL segmentation approaches mainly consist of encoder-decoder convolutional neural networks (CNN), in particular fully convolutional networks [Lon15] and the U-Net architecture [Ron15], using ResNet [HeK16], Inception [Sze15] or VGG [Sim15] as popular encoder backbones, as these have performed best in the field of medical image segmentation.

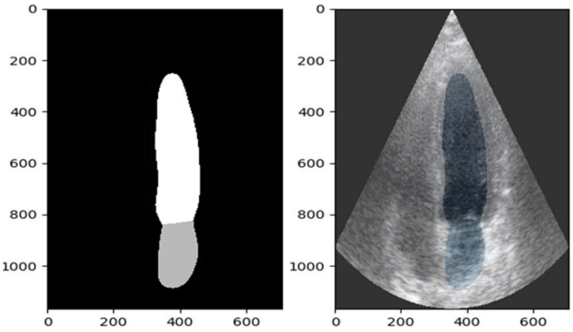


Figure 2. Ground truth annotation of the LV and LA in a4c image (left), overlay of the annotation on the original echocardiogram (right).

TTE is the most commonly performed imaging examination in cardiology, due to the fact that it is non-invasive, has low cost and high accessibility, yet up to 80% of annual publications between 2016 and 2019 [Che20] on DL-based cardiac image segmentation worked with magnetic resonance imaging (MRI) data [YuL17; Wol17], mainly because of larger dataset availability. Computer tomography (CT) [Zre15; Ton17] and echocardiography [Zha18; Smi17], despite their clinical importance, only played a subordinate role due to the lack of annotated datasets.

As for MV measurement in particular, in clinical practice, MV dimensions are either manually obtained by a user manually selecting points on frozen frames throughout the cardiac cycle [Gar15; Dwi14] or by semi-automatic segmentation. E.g., [Pou12] proposes an MV morphometry method, which requires user initialized selection of a region of interest and anatomical landmarks followed by feature-based contour segmentation.

Further feature-based (semi)-automatic methods for MV assessment often require vendor-specific software for analysis. In addition, they have high computational run times and have only been assessed in single-center studies with small patient numbers and little variety in MV conditions [Nol19].

3. PROPOSED SOLUTION

3.1 Dataset

The CAMUS dataset includes 450 annotated patient sub-datasets consisting of TTE apical four- and two-chamber views. Each patient sub-dataset consists of one cardiac cycle per view, but ground truth segmentations are only provided at the image frames at end-diastole (ED) and end-systole (ES). Image sizes vary, in a range between 400 x 700 pixels and 700 x 1000 pixels, with a spatial grid resolution of 0.3 mm along the *x*- and 0.15 mm along the *y*-axis. The dataset includes images of various acquisition settings, e.g., resolution, image contrast and transducer angle. Furthermore, some images are not properly centered, therefore certain anatomical structures (ventricles, atria) are potentially not fully visible on them. No further data selection or preprocessing has been performed. This results in a heterogeneous dataset, which is a realistic representation of data acquired in clinical practice.

Manual delineation of the LV, LA, and epicardium was performed by a cardiologist using a defined segmentation protocol. In particular, the LV delineation contour was to be terminated at the points where the MVLs are hinging. Epicardium annotation is not relevant to mitral valve measurement and is therefore not considered in this work.

3.2 Training the segmentation model

We introduce a two–step method for estimating MV hinge point coordinates. This method uses a deep learning algorithm for segmentation of the left ventricle and left atrium and subsequent feature-based image processing for the estimation of MV hinge point coordinates.

Step 1: deep learning segmentation algorithm

Since [Lec19] demonstrated that the U-Net architecture showed slightly better segmentation accuracy on the CAMUS dataset than more sophisticated encoder-decoder networks, a U-Net with the VGG16 backbone was used to train our model. Model training was implemented in Python version 3.7.7 using Tensorflow 2.0.0 in conjunction with the Keras API. The Adam optimizer [Kin14], a learning rate of 10^{-3} and the categorical cross-entropy loss function were used for training. No data augmentation was performed on the dataset.

450 patient sub-datasets were divided into three groups, 350 for training, 50 for validation, and 50 for testing, a roughly 80%/10%/10% split. Model training and validation were performed on an NVIDIA GeForce RTX 2060 and ran for 50 epochs, after which the validation accuracy stagnated or dropped, and training was terminated to avoid overfitting (Fig. 3). As a result of this first step, image pixels are assigned to the left ventricle (LV), left atrium (LA), and background.

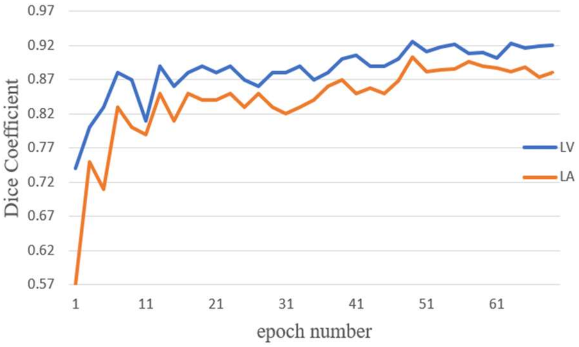


Figure 3. Dice coefficient of the left ventricle and left atrium for the validation dataset, training was stopped after 50 epochs.

3.3 Extraction of mitral valve hinge points

Step 2: feature-based hinge point extraction

A purely feature-based algorithm for mitral valve detection, which uses e.g., thresholding, edge-detection, or histogram methods, is likely to perform insufficiently in regions of low pixel grayscale gradients, and thus does not detect both MVLs reliably.

In numerous clinical cases, anatomical structures are not clearly visible on echocardiograms, due to low image contrast and blurry boundaries between anatomical structures.

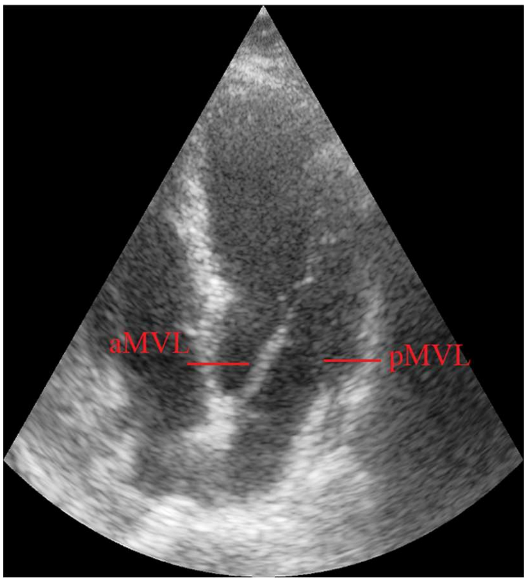
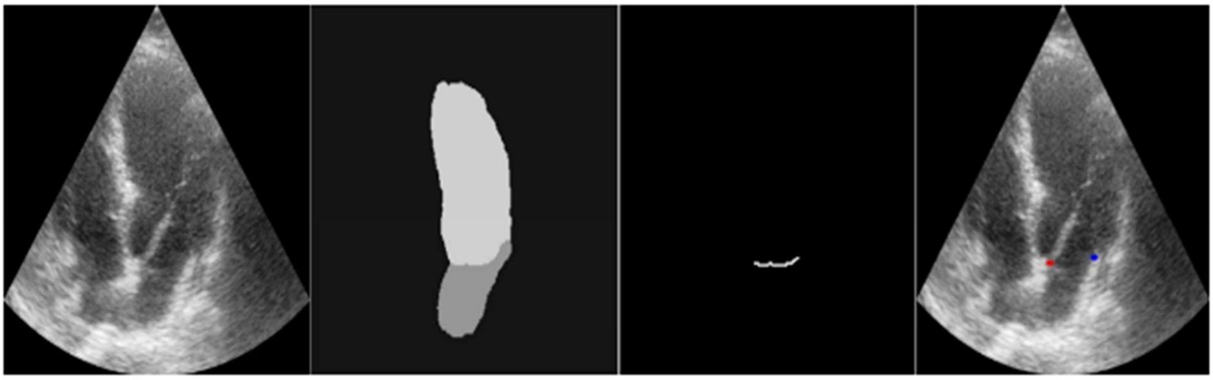


Figure 4. Apical four-chamber view (a4c) at end diastole (ED). The intersecting line between image plane and anterior mitral valve leaflet (aMVL) is clearly visible, whereas the posterior leaflet (pMVL) is barely visible.



a) original echocardiogram b) LA, LV segmentation c) extracted contact line d) hinge points overlay

Figure 5. Overview of the proposed method: The original echocardiogram (a) is first segmented, using the CNN. The resulting contact line (c) between the segments of left ventricle and left atrium (b) is then used to extract the mitral valve hinge points (d).

In particular, MVLs are hardly distinguishable from the background in many cases (Fig. 4).

Therefore, we use the DL-generated segmentations of a4c- and a2c echocardiograms (see section 3.2) to estimate MV hinge point coordinates. Figure 5 gives an overview of the individual steps in our proposed method.

According to the contouring protocol in [Lec19], the LV contour was to be terminated in the MV plane, at the MV hinge points. Using the resulting contact line between LV and LA segmentation (Fig. 5c), we define the anteriormost point of the contact line as the anterior mitral valve leaflet (aMVL) and the posteriormost point of the contact line as the posterior mitral valve leaflet (pMVL).

This second step results in *x*- and *y*-coordinates for the aMVL and pMVL.

3.4 Evaluation Metrics

The segmentation CNN in combination with the following MV hinge point extraction, is to be considered as a measurement tool for the pixel-coordinates of the MV hinge points. In this method, each measurement $\hat{z}$ is determined by

$$\hat{z} = z + \Delta\hat{z} = z + \left(\Delta\hat{z}^{(\mathrm{bias})} + e^{(\hat{z})}\right)$$

where z is the true value, $\Delta\hat{z}^{(\mathrm{bias})}$ the systematic error and $e^{(\hat{z})}$ the random error (DIN 1319-1/ISO 11843-1). Typically, a normal distribution of errors $\Delta\hat{z}$ is assumed and characterized by its mean μ and standard deviation σ. To assess the normality of our error distributions, we performed Shapiro-Wilk tests for $\Delta\hat{z}$ data series of each subgroup ($\Delta\hat{x}_{\mathrm{aMVL}}$, $\Delta\hat{y}_{\mathrm{aMVL}}$, $\Delta\hat{x}_{\mathrm{pMVL}}$, $\Delta\hat{y}_{\mathrm{pMVL}}$).

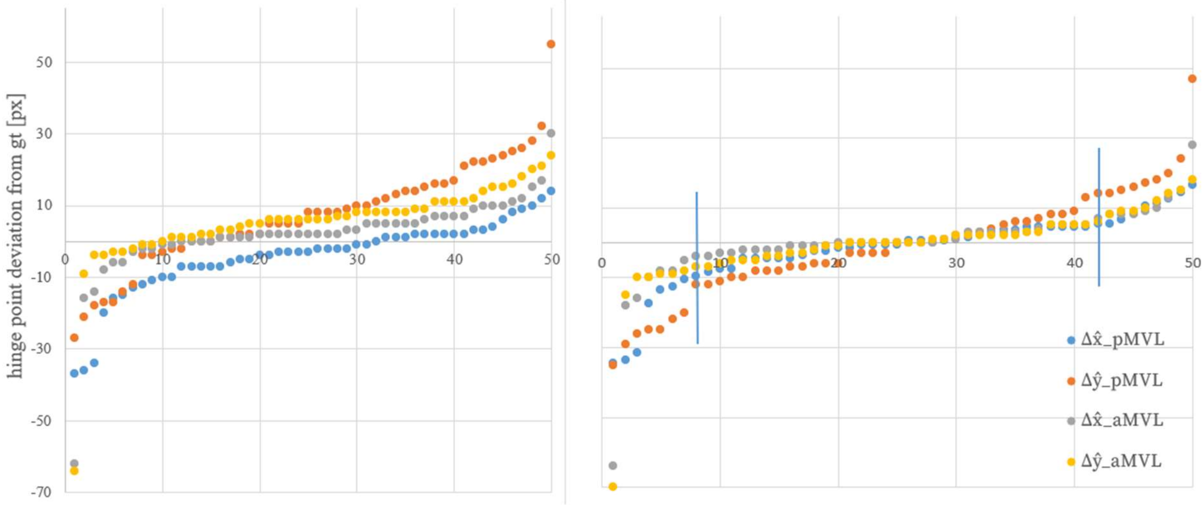


Figure 6. Sorted hinge point coordinate errors [px] before (left) and after (right) calibration of the systematic error $\Delta\hat{z}^{(\mathrm{bias})} = \left\{\Delta\hat{x}_{\mathrm{pMVL}}^{(\mathrm{bias})}, \Delta\hat{y}_{\mathrm{pMVL}}^{(\mathrm{bias})}, \Delta\hat{x}_{\mathrm{aMVL}}^{(\mathrm{bias})}, \Delta\hat{y}_{\mathrm{aMVL}}^{(\mathrm{bias})}\right\}$. The blue vertical lines in the right diagram show the 15th and 85th percentiles, which are evaluated in the results section.

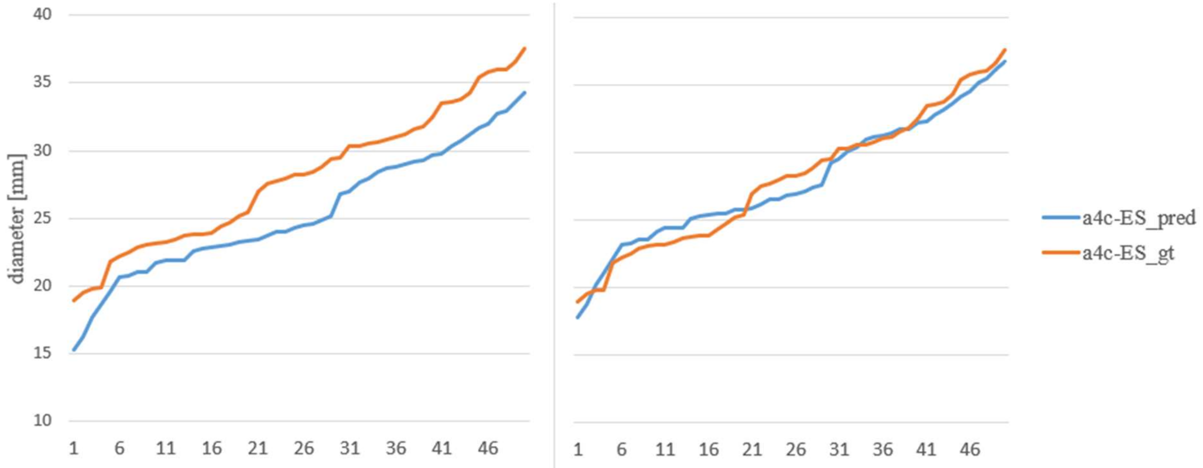


Figure 7. Sorted ground truth and predicted MV diameters (mm) before (left) and after (right) calibration of the systematic error $\Delta\hat{z}^{(\text{bias})}$. The predicted MV diameters were systematically underestimated by 13.8%.

These tests resulted in p-values of $p < 0.05$ for all but one data series, thus the assumption of normal distribution is rejected.

Therefore, we characterize the distributions by their 15-, 50- (median), and 85-percentiles instead, as equivalents to $\mu - \sigma$, μ and $\mu + \sigma$.

To account for systematic errors $\Delta\hat{z}^{(\text{bias})}$ and subsequently only evaluate random errors $e^{(\hat{z})}$, median deviations $\Delta\hat{z}^{(\text{bias})}$ ($\Delta\hat{x}_{\text{aMVL}}^{(\text{bias})}$, $\Delta\hat{y}_{\text{aMVL}}^{(\text{bias})}$, $\Delta\hat{x}_{\text{pMVL}}^{(\text{bias})}$, $\Delta\hat{y}_{\text{pMVL}}^{(\text{bias})}$) are subtracted from measured values $\hat{z}$ (calibration). We then evaluate the 15-85 percentile range of calibrated values. (Fig. 6).Calibrated, random x- and y- coordinate errors of the aMVL and the pMVL ($e_{\text{aMVL}}^{\hat{x}}, e_{\text{aMVL}}^{\hat{y}}, e_{\text{pMVL}}^{\hat{x}}, e_{\text{pMVL}}^{\hat{y}}$) will be evaluated individually.

In addidition, the segmentation accuracy of the CNN as well as the resulting MV diameters will be evaluated.

4. RESULTS

4.1 Chamber segmentation accuracy

To evaluate the segmentation accuracy of our network, we use the Dice Coefficient (D).

	$D_{\text{LV\_ED}}$	$D_{\text{LV\_ES}}$
proposed	0.931	0.915
[Lec19]	0.939	0.916

Table 1. Dice Coefficients of LV U-Net segmentation.

The combined Dice Coefficient of all LVs (at ED and ES) is D_{LV} = 0.923, with segmentations at ED performing slightly better than at ES ($D_{\text{LV\_ED}}$ = 0.931, $D_{\text{LV\_ES}}$ = 0.915). Unlike the procedure described in [Lec19], we did not perform any post-processing (e.g. connected component analysis) on the segmentation result, yet are still in line with their best U-Net

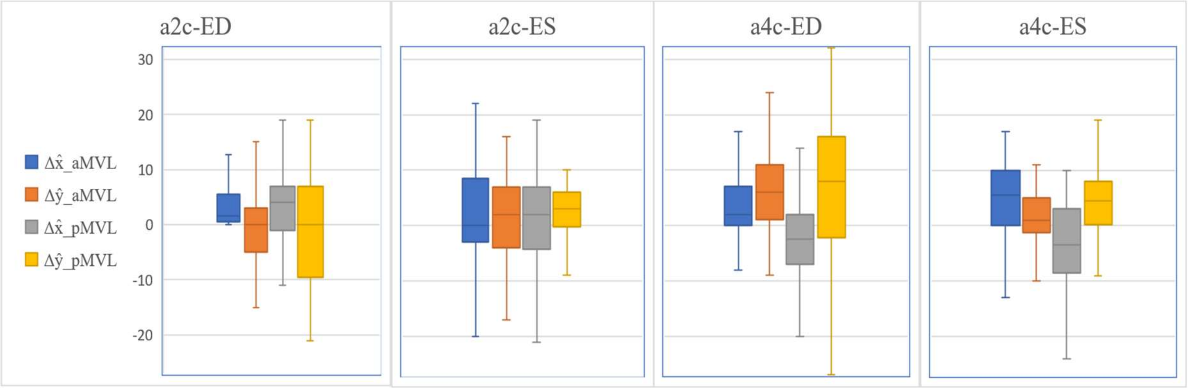


Figure 8. Boxplot diagrams of hinge point coordinate errors $\Delta\hat{x}_{\text{aMVL}}$, $\Delta\hat{y}_{\text{aMVL}}$, $\Delta\hat{x}_{\text{pMVL}}$, $\Delta\hat{y}_{\text{pMVL}}$ [px] of a4c, a2c images at ES and ED before calibration.

segmentation accuracy ($D_{\mathrm{LV\_ED}}$ = 0.939, $D_{\mathrm{LV\_ES}}$ = 0.916).

4.2 Mitral valve annulus diameter

Our experimental measurements of the MV annulus diameters (Table 1) lie well within the range of empiric MV measurement data. E.g., in [Dwi14], 5-95 percentile ranges for the mitral annulus diameter of 22-38 mm are stated for women and 25-41 mm for men.

While median diameter estimations for a2c images and a4c images (Table 2) at ED conformed well with ground truth values, a significant systematic estimation bias was observed in a4c images at ES. Here, the median diameter was underestimated by 13.8% (Fig. 7). This systematic error can also be seen in the individual hinge point coordinates in section 4.3.

	predicted	ground truth
a4c-ED	27.9 mm	28.8 mm
a4c-ES	24.4 mm	28.3 mm
a2c-ED	31.3 mm	31.7 mm
a2c-ES	26.3 mm	26.1 mm

Table 2. Predicted and ground truth median MV annulus diameter values a2c and a4c views at ES and ED.

4.3 MV hinge point coordinates

Evaluation of individual hinge point coordinate errors (Fig. 8) shows further systematic estimation errors $\Delta\hat{z}^{(\mathrm{bias})}$. Since both the anterior and posterior hinge points are estimated too far inwardly (medial) in a4c view at ES, the corresponding underestimation in diameters (see section 4.2) is explained.

Figure 9 displays the individual systematic errors $\Delta\hat{z}^{(\mathrm{bias})}$ for the aMVL- and pMVL hinge point of each view type. The coordinate estimation is, on average, biased towards the bottom (0.5 mm) and the right (0.3 mm). The estimation accuracy in terms of absolute coordinate error distance in mm was much lower for the x- compared to the y-coordinate, as can be seen in Table 3.

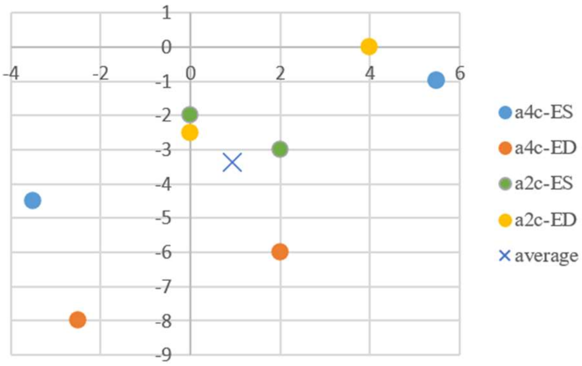


Figure 9. Systematic errors $\Delta\hat{z}^{(\mathrm{bias})}$ [px] of each individual subgroup (a4c-ED, a4c-ES, a2c-ED, a2c-ES). On average (marked by blue x) hinge points are estimated about 0.3 mm too far posterior (in the image: right) and about 0.5 mm too far cranial (in the image: down).

This is almost fully explained by the spatial resolution of the images, which is 0.3 mm along the x- and 0.15 mm along the y- axis, as described in section 3.1.

This results in absolute median coordinate errors of 1.35 mm for all x-coordinates and 0.75 mm for all y-coordinates. When comparing the median of absolute error distances $\mathrm{median}\left(\left|e^{(\hat{z})}\right|\right)$ of the different views (a4c-ED, a4c-ES, a2c-ED, a2c-ES), estimation accuracy was approximately equal in the four subgroups.

4.4 Impact of off-center images

Looking at the correlation plot (Fig. 10) between predicted and ground truth x_{aMVL}, x_{pMVL} coordinates of the MV hinge points in a4c views, a subdivision of data points into two groups can be observed. We suspect this is likely due inaccurate centering of the displayed portion of the a4c view. Since most of the misaligned, atypical a4c images are heavily centered on the LV (Fig. 11), the LA is not properly displayed, which leads to lower segmentation accuracy and thus higher estimation errors of the MV hinge point coordinates. No similar phenomenon of data subdivision was observed in hinge point coordinates in a2c view.

	$e^{\hat{x}}_{\mathrm{aMVL}}$		$e^{\hat{y}}_{\mathrm{aMVL}}$		$e^{\hat{x}}_{\mathrm{pMVL}}$		$e^{\hat{y}}_{\mathrm{pMVL}}$	
	signed	median(abs)	signed	median(abs)	signed	median(abs)	signed	median(abs)
a4c-ED	[−1.2; 1.5]	1.35	[−0.9; 0.75]	1.2	[−2.2; 1.35]	0.9	[−1.65; 1.8]	0.75
a4c-ES	[−2.3;1.95]	1.65	[−0.75; 0.9]	0.53	[−2.55; 2.3]	1.8	[−1.8; 0.83]	0.53
a2c-ED	[−1.5; 1.2]	1.35	[−1.8; 1.35]	0.83	[−2.1; 2.1]	1.2	[−1.88; 0.8]	1.05
a2c-ES	[−1.8; 2.8]	1.65	[−1.35; 0.9]	0.45	[−2.6; 2.1]	1.2	[−1.2; 0.75]	0.75

Table 3. Results of hinge point estimations [mm]. In addition to the signed 15-85 percentile ranges of $e^{(\hat{z})}$ ($e^{\hat{x}}_{\mathrm{aMVL}}$, $e^{\hat{y}}_{\mathrm{aMVL}}$, $e^{\hat{x}}_{\mathrm{pMVL}}$, $e^{\hat{y}}_{\mathrm{pMVL}}$), the median of absolute error distances $\mathrm{median}\left(\left|e^{(\hat{z})}\right|\right)$ is stated.

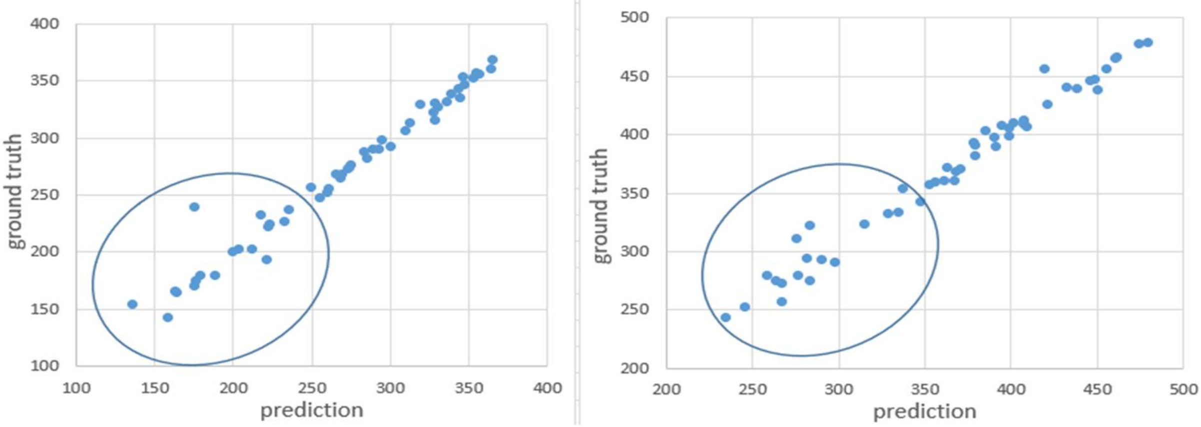


Figure 10. Correlation plot of ground truth and predicted x_{aMVL} (left) and x_{pMVL} (right) of a4c images at ED. The circled subgroup of hinge points is positioned far more anterior than the rest of the datapoints and shows higher deviations from the ground truth.

5. CONCLUSION

We demonstrated a two–step method for estimating MV hinge point coordinates using deep learning segmentations of the left ventricle and left atrium and subsequent feature based image processing. With 15-85 percentile ranges of random coordinate estimation errors $e^{(\hat{z})}$ between [–0.9 mm; 0.75 mm] and [–2.55 mm; 2.3 mm] and absolute median coordinate errors of 1.35 mm and 0.75 mm respectively, the resulting estimations are satisfactory, but further improvements can be made.

If the LV and LA can be adequately segmented by the neural net, the resulting segmentation mask can be used to reliably determine the MV hinge point coordinates.

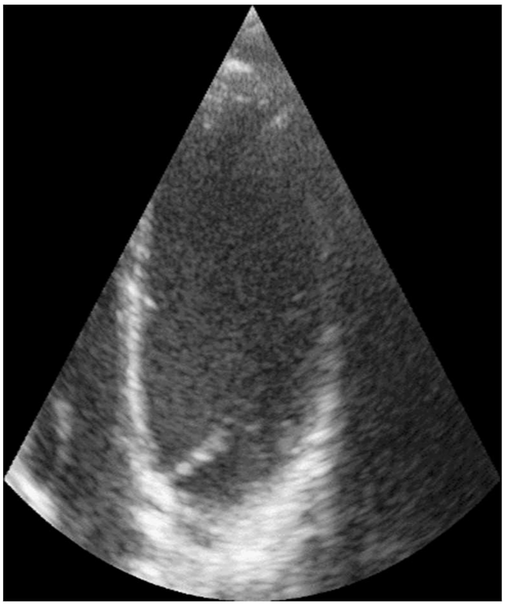


Figure 11. Example of inaccurately centered image of the four-chamber view, only the LV and part of the RV are visible.

We used the CAMUS dataset in this work, which is quite heterogeneous and, as such, close to clinical practice, as described above. This is beneficial for the generalizability of the network. On the other hand, the heterogeneity (e.g., low-quality images, edge cases described in section 4.4) is detrimental to estimation accuracy. Depending on the use case, adjustments to the training data set can be made. If generalizability is the highest priority, further low-quality and off-centered images should be added to adequately represent them in the training data. Otherwise, if estimation accuracy is the priority, low-quality images can be removed from the dataset with instructions to the physician to record more appropriate images.

6. REFERENCES

[Che20] Chen, C., Qin, C., Qiu, H., Tarroni, G., Duan, J., Bai, W., and Rueckert, D. „Deep Learning for Cardiac Image Segmentation: A Review, " Front. Cardiovasc. Med., doi.org/10.3389/fcvm.2020.00025, 2020.

[Dwi14] Dwivedi, G., Mahadevan, G., Jimenez, D., Frenneaux M. and Steeds, R.P. „Reference Values for Mitral and Tricuspid Annular Dimensions Using Two-Dimensional Echocardiography," Echo Research and Practice, pp. 43-50, 2014.

[Fla00] Flachskampf, F.A., Chandra, S., Gaddipatti, A., Levine, R.A., Weyman, A.E. Ameling, W. et al. „Analysis of Shape and Motion of the Mitral Annulus in Subjects With and Without Cardiomyopathy by Echocardiographic 3-Dimensional Reconstruction," Journal of the American Society of Echocardiography, Vol.13, pp. 277-287, 2000.

[Gar15] Garbi, M. and Monaghan, M.J. „Quantitative Mitral Valve Anatomy and Pathology, “ Echo Research and Practice, pp. 63-72, 2015.

[HeK16] He, K., Zhang, R., Ren, S., and Sun, J. „Deep Residual Learning for Image Recognition, “IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.

[Kin14] Kingma, D.P. and Ba, J. „Adam: A Method for Stochastic Optimization, “ arXiv preprint arXiv:1412.6980, 2014.

[Kwa09] Kwan, J., Jeon, M., Kim, D., Park, K., and Lee, W. „Does the Mitral Annulus Shrink or Enlarge During Systole? A real-time 3D-Echography Study, “ Korean Med Sci, pp. 203-208, 2009.

[Lec19] Leclerc, S., Smistad, S.E., Pedrosa, J., Ostvik, A., Cervenansky, F., Espinosa, F. et al. „Deep Learning for Segmentation Using an Open Large-Scale Dataset in 2D Echocardiography,“ IEEE Trans Med Imaging, pp. 2198-2210, 2019.

[Lon15] Long, J., Shelhamer, E., and Darrell, T. „Fully Convolutional Networks for Semantic segmentation, “IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431-3440, 2015.

[Nol19] Nolan, M.T. and Thavendiranathan, P. „Automated Quantification in Echocardiography“, Cardiovascular Imaging, pp. 1073-1092, 2019.

[Okt14] Oktay, O., Shi, W., Keraudren, K., Caballero, K., Rueckert, D., and Hajnal, D. „Learning Shape Representations for Multi-Atlas Endocardium Segmentation in 3D Echo Images,“ Midas Journal, pp. 57-64, 2014.

[Pou12] Pouch, A.M., Xu, C., Yushkevich, P.A., Jassar, A.S., Vergnat, M., Gorman, I. et al. „Semi-Automated Mitral Valve Morphometry and Computational Stress Analysis Using 3D Ultrasound,“ Journal of Biomechanics, pp. 903-907, 2012.

[Ron15] Ronneberger, O., Fischer P., and Brox, T. „U-Net: Convolutional Networks for Biomedical Image Segmentation, “ LNCS, vol. 9351, pp. 234-241, 2015.

[Sim15] Simonyan, K. and Zisserman, A. „Very Deep Convolutional Networks for Large-Scale Image Recognition, “ CoRR abs/1409.1556 , 2015.

[Smi17] Smistad, E., Ostvik, A., Haugen, B. O., and Lovstakken, L. „2D Left Ventricle Segmentation Using Deep Learning,“ 2017 IEEE International Ultrasonics Symposium (IUS), pp. 1-4, 2017.

[Sze15] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D. et al. „Going Deeper with Convolutions,“ IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9, 2015.

[Tav13] Tavakoli, V. and Amini, A.A. „A Survey of Shape-Based Registration and Segmentation Techniques for Cardiac Images, “ Computer Vision and Image Understanding, pp. 966-989, 2013.

[Ton17] Tong, Q., Ning, M., Si, W., Liao X., and Qin, J. „3D Deeply-Supervised U-Net Based Whole Heart Segmentation,“ Proceedings, STACOM at MICCAI, pp. 224-232, 2017.

[Wol17] Wolterink, J., Leiner, T., Viergrever, M., and Isgum, I „Automatic Segmentation and Disease Classification Using Cardiac Cine MR Images, “ Lecture Notes in Computer Science, Vol. 10663, pp. 101-110, 2017.

[YuL17] Yu, L., Cheng, J., Dou, Q., Yang, X., Chen, H., Qin, J. et al, „Automatic 3D Cardiovascular MR Segmentation with Densely-Connected Volumetric ConvNets,“ Proceedings, Part II, MICCAI, pp. 287-295, 2017.

[Zha18] Zhang, J., Gajjala, S., Agrawal, P., Tison, G., Hallock, L., Beussnik-Nelson, L. et al, „Fully Automated Echocardiogram Interpretation in Clinical Practice: Feasibility and Diagnostic Accuracy,“ CIRCULATIONAHA, pp. 1623-1635, 2018.

[Zha21] Zhang, Q., Liu, Y., Mi, J., Wang, X., Liu, X., Zhao, F. et al „Assessment of Mitral Regurgitation Severity Using the Mask R-CNN Algorithm with Color Doppler Echocardiography Images,“ Computational and Mathematical Methods in Medicine, 2021.

[Zre16] Zreik, M., Leiner, T., de Vos, B. D., van Hamersvelt, R. W., Viergrever M. A., and Isgum, I. „Automatic Segmentation of the Left Ventricle in Cardiac CT Angiography Using Convolutional Neural Network,“ International Symposium on Biomedical Imaging, pp. 40-43, 2016.

Fast Shape Classification Using Kolmogorov-Smirnov Statistics

Alexander Köhler
BTU
Cottbus-Senftenberg
Platz der deutschen
Einheit 1
03046, Cottbus,
Germany
koehlale@b-tu.de

Ashkan Rigi
BTU
Cottbus-Senftenberg
Platz der deutschen
Einheit 1
03046, Cottbus,
Germany
Ashkan.Rigi@b-tu.de

Michael Breuß
BTU
Cottbus-Senftenberg
Platz der deutschen
Einheit 1
03046, Cottbus,
Germany
breuss@b-tu.de

Abstract

The fast classification of shapes is an important problem in shape analysis and of high relevance for many possible applications. In this paper, we consider the use of very fast and easy to compute statistical techniques for assessing shapes, which may for instance be useful for a first similarity search in a shape database. To this end, we construct shape signatures at hand of stochastic sampling of distances between points of interest in a given shape. By employing the Kolmogorov-Smirnov statistics we then propose to formulate the problem of shape classification as a statistical hypothesis test that enables to assess the similarity of the signature distributions. In order to illustrate some important properties of our approach, we explore the use of simple sampling techniques. At hand of experiments conducted with a variety of shapes in two dimensions, we give a discussion of potentially interesting features of the method.

Keywords

Statistical Shape Analysis; Shape Classification; Shape Similarity; Kolmogorov-Smirnov; Hypothesis Testing; Sampling Methods

1 INTRODUCTION

Shape classification is the problem of finding similar shapes among a set of shapes. The corresponding techniques have numerous applications in many fields such as computer vision [1], medical imaging [9, 10] and engineering [20]. Besides of numerous advances in the use of neural networks in the field, there is still a need for approaches that are easy to interpret and give rise to simple and fast computations without the need to deal with intricate neural network architecture, training and data generation issues. In this paper we follow the statistical shape analysis approach, a recent overview may be found in [6]. A more general overview on classic shape analysis methods including some statistical approaches can be found in [3, 4].

Turning to the construction of shape analysis methods in the statistical approach, usually a computational representation of shapes, called a descriptor or signature, is defined that is used for comparison and classification. There is a wide range of possible descriptors which often represent geometric information that can be derived from a given shape [21]. To this end the shape of an object may suitably be described by its boundary. Basically, the descriptors may be classified in two categories: local and global descriptors. Local descriptors are defined for each point of the shapes' boundary and often show the local geometric structure of the shape around the point, whereas global descriptors are defined based on information derived over the entire shape. Examples for some of the most commonly used shape descriptors are centroid distance function, tangent angle, curvature function, area function, triangle-area representation or chord length [19]. In this paper we follow a common terminology in shape analysis by denoting the shape signature as the global representation for comparison, whereas the signature is constructed at hand of local descriptors. Once useful shape signatures have been computed, often a metric is introduced in order to asses quantitatively how close any two given shapes are in terms of the metric distance, see [2] for a recent discussion.

In order to assess shape statistics and employ them for shape comparison, there have also been some efforts to formulate the task as a hypothesis test, see for instance [8, 17]. In this work, we explore the use of the Kolmogorov-Smirnov test in order to assess if two shape signatures are similar. While the use of Kolmogorov-Smirnov statistics appears to have found few applications in image processing, see e.g. [18], it appears that it has not been used for the purpose of shape classification in a similar way as in this paper.

https://www.doi.org/10.24132/CSRN.3201.22

Turning to the construction of statistical shape distribution functions, we build upon the article [15] which is at the same time the most related work to the current paper. It is one of the first works that formulates the shape comparison task in 3D in terms of shape distribution comparison. The comparison of easy-to-compute, statistical shape distribution functions is potentially highly attractive for shape comparison and classification, since this is methodically much simpler than traditional shape matching methods that often rely on pose registration, feature correspondence or model fitting.

In [15], the idea is to represent the signature of an object as a shape distribution sampled stochastically from a shape, measuring in this way global geometric properties of an object. Therefore the authors of [15] have studied several ways to construct shape distributions, including the use of the following descriptors among others: *(i)* the distance between the centroid of a shapes' boundary and randomly selected points of the boundary, as well as *(ii)* the distance between two random points on the boundary. We will also make use of these two basic building blocks in the current work. After choosing the descriptor, [15] calculate the shape signature by stochastic sampling the needed points over an interpolated mesh, followed by a binning procedure of corresponding descriptor values. For shape comparison, in their work dissimilarity measures based on L^p norms have been employed.

Our Contribution. In this paper we adapt the 3D method from [15] to the 2D setting, which allows us to investigate some interesting properties of the proceeding. In doing this, we perform a few technical adaptations, for example we do not interpolate between shape boundary points for stochastic sampling. As our main contribution, we propose how to formulate the shape classification task as a statistical hypothesis test in this setting, making use of the Kolmogorov-Smirnov statistics. This test is efficient to conduct, however, its use appears to be uncommon in shape analysis. It is one of the benefits of the proposed framework that it represents a natural methodical fit to Kolmogorov-Smirnov testing. In order to illustrate some properties of the discussed setting, we consider an adaptive sampling strategy as well as a simple coarsening routine. By adaptive sampling it is possible to emphasize the role of shape-specific points like corners within the statistics, following the classic idea of shape analysis by exploring landmarks, cf. [6]. We exemplify and illustrate our proceeding by several tests with shapes from standard shape datasets.

2 DESCRIPTION OF OUR MODEL

Now we give a detailed account on the used methods for shape classification.

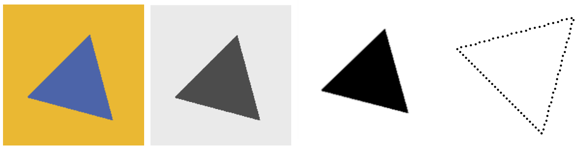


Figure 1: Illustration of the basic process of generating point clouds from color images (left). We take the first color channel (middle left) and transform it into a black and white image (middle right). After that we can obtain a point cloud (right) via the MATLAB routine *bwboundaries*.

2.1 About Shapes

Our shapes are given as a curve in $S \subset \mathbb{R}^2$. Any of these curves is represented as a point cloud consisting of $n \in \mathbb{N}$ points $p_i = (x_i, y_i)$, $i \in \{1, 2, \dots, n\}$, where x_i and y_i describe the x- and y-coordinates of the i-th point, respectively. Our point clouds are in practice already ordered, so that the points p_i and p_{i+1} are neighbors. We make use of the established neighbourhood relationship within the adaptive sampling scheme.

For illustrating our proceeding, we will consider in this section few shapes taken from the Mendeley 2D geometric shapes dataset [12]. While we discuss the database and its use for experimental validation in more detail later on, let us mention here that typically shape databases contain colour or gray value images of shapes. The Figure 1 illustrates how we obtain from such given images the shapes in terms of point clouds with ordered boundary points.

2.2 Adaptive Boundary Sampling

As indicated in the introduction, for some studies we employ an adaptive sampling of shape boundary points. The aim of the adaptive method we employ is to reduce the number of boundary points in regions that are close to line segments, and to keep salient points of the boundary like at corners or regions with many details. The adaptivity is realised here via the concept of adaptive areas.

The idea of adaptive area comes from [7]. To find out if a point p_i is of interest we calculate the area A_i of the triangle spanned by p_i and its direct neighbours p_{i-1} and p_{i+1}, cf. Figure 2. Then, for an arbitrary triangle the area A_i can be computed via

$$A_i = \frac{1}{2} |\vec{v}_1| |\vec{v}_2| \sin(\gamma) \qquad (1)$$

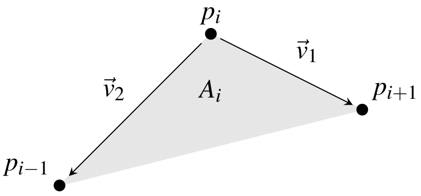


Figure 2: Triangle corresponding to point p_i

where $|\vec{v}|$ denotes the length of the vector $\vec{v}$, and γ the angle enclosed by these vectors. The angle can be computed with

$$\cos(\gamma) = \frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1|\,|\vec{v}_2|} \tag{2}$$

with $\vec{v}_1 \cdot \vec{v}_2$ being the scalar product of these two vectors. So the area A_i depends on the angle between the vectors $\vec{v}_1$ and $\vec{v}_2$. If both vectors are parallel the area reaches a minimum. The maximal area is obtained when both vectors are orthogonal to each other. With this in mind we say, that a point p_i is of interest if the area is larger than a user-defined threshold T. If the area is below this threshold we can assume that the point p_i is located at a line-like segment.

To make this approach a bit more robust against scaling we normalize the vectors $\vec{v}_1$ and $\vec{v}_2$. This means we redefine $\vec{v}_k := \vec{v}_k/|\vec{v}_k|$, for $k = 1,2$. Then the formula for $\cos(\gamma)$ and A_i simplify to

$$\cos(\gamma) = \vec{v}_1 \cdot \vec{v}_2 \quad \text{and} \quad A_i = \frac{1}{2}\sin(\gamma) \tag{3}$$

In the end we neglect all points p_i for which the corresponding triangle has an area smaller that a threshold parameter T, i.e. in case $A_i < T$.

2.3 Shape Descriptors and Signature

In this section we want to describe how to construct our shape signature functions. With these functions we then perform the classification with the Kolmogorov-Smirnov test.

The signature functions we consider are based on shape descriptors. These should be geometrically meaningful as well as fast to compute. Several possible signature functions were presented in [15]. We opt here to adopt the D1 and D2 shape descriptors from the latter work and renaming them into d^1 and d^2, respectively, since both of them are readily interpreted and obtained in low computing time:

$\mathbf{d^1}$ distance For the d^1 shape descriptor we calculate the distance between a sampled point on the shape and a fixed point $p_c = (x_c, y_c)$.

$$d^1(p_i) = \|p_i - p_c\|_2 \quad \text{with} \quad p_c = \frac{1}{n}\sum_{i=1}^{n} p_i \tag{4}$$

with $\|\cdot\|_2$ the Euclidean norm. For the fixed point p_c we, thus, consider here the geometrical centre of a given shape, determined by arithmetic averaging of its boundary points. Thereby n is again the total number of boundary points of the given shape.

$\mathbf{d^2}$ distance For the d^2 shape descriptor we calculate the Euclidean distance between two points p_i and p_j, $i \neq j$, from our shape:

$$d^2(p_i, p_j) = \|p_i - p_j\|_2 \tag{5}$$

For demonstration purposes we focus on the d^1 distance descriptor. The Figure 3 gives an account of the descriptor d^1, when selecting all boundary points of depicted shapes in the order they make up the boundary. At hand of this figure, it is surely easy to perceive how the d^1 descriptor works and that it gives a useful account of a shape. Let us note already here that the descriptor values of triangle and pentagon appear to be very characteristic, while descriptor values of nonagon and circle appear just within a certain narrow band width, which may make these shapes hard to distinguish at hand of d^1.

Having defined our shape descriptors d^1 and d^2, we draw m stochastic samples of them, obtained by random selection of boundary points for the computation of d^1 and d^2 distances, respectively. This means, by taking m random samples of boundary points we can evaluate m times the descriptors d^1 and d^2, respectively, and store the corresponding values as d^k_j with $j = 1,\ldots,m$ and $k = 1,2$. The corresponding shape signatures are obtained then as the collections $D^1 = (d^1_1,\ldots,d^1_m)$ and $D^2 = (d^2_1,\ldots,d^2_m)$ of these samples. For ease of notation, we consider in the following mainly (i.e. if not stated otherwise) the d^1 distance descriptor, and denote by D_i the signature of the i-th shape obtained by the corresponding D^1 collection of descriptor values.

2.4 Comparing Distributions

Having computed the signatures D_i for each shape, we want to test if they belong to the same distribution.

By default we can not expect that the signatures obtained by the procedure explained in previous section are comparable. Imagine a small and a large triangle. The possible distances from the sample of the small triangle will on average be smaller than the distances from the sample of the larger triangle. To tackle this normalization issue the idea is to scale the different samples D_i to one reference sample D_{ref}. See for example [8] for a thorough discussion of normalization methods. In this work we adopt a *mean-scaling*, which equalizes the means of two given samples. In a slight abuse of notation, we keep D_i for the signature of the i-th shape after normalization in the following.

Now we can test if two signatures D_i and D_j are from the same shape distribution. As indicated, we propose to do this at hand of testing the corresponding hypothesis. There is a large variety of possible hypothesis testing setups in the statistical literature, see for instance, [14] for an overview. One could, for example, test if the medians of the populations from which two or more samples are drawn are equal or not, or one may compare the maximum mean discrepancy of two given samples. However, among the various possibilities, the best fitting test for our circumstances is the Kolmogorov-Smirnov test, since this is designed to evaluate if two

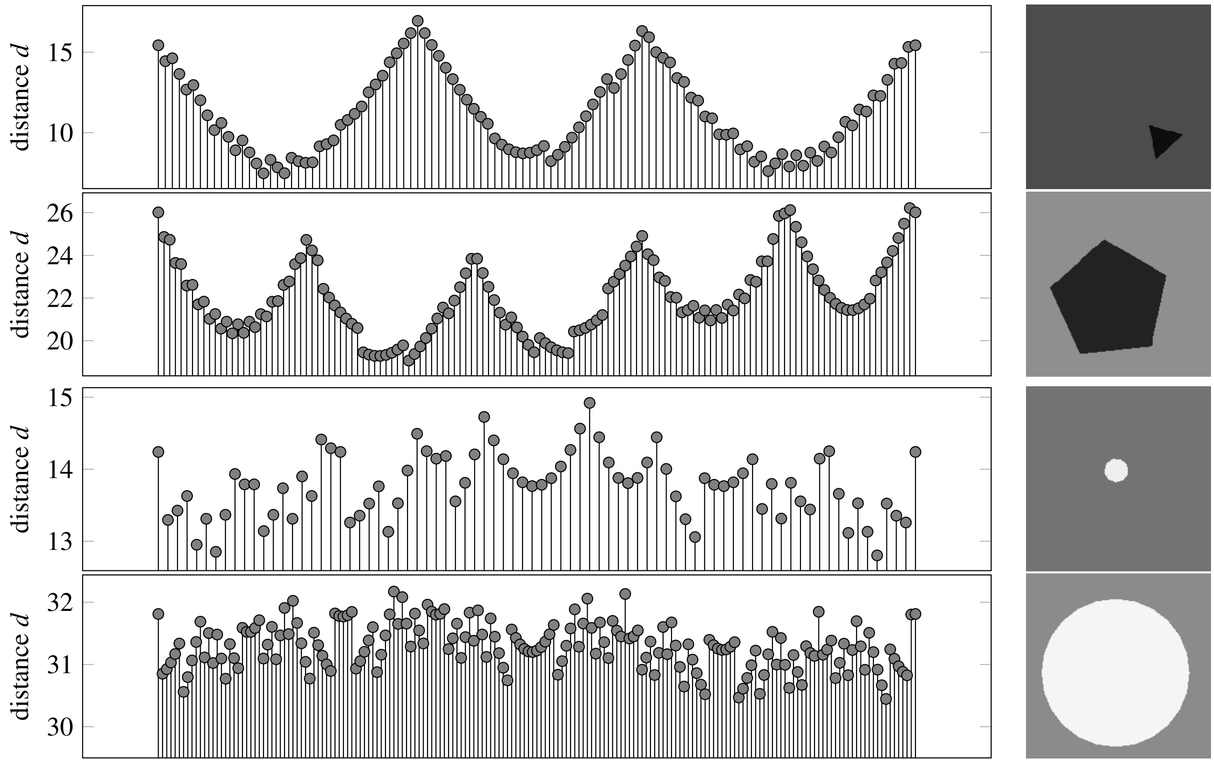


Figure 3: We plotted d^1 distance samples for four geometrical shapes, taking all available boundary points. From top to bottom we see the samples of a triangle, pentagon, nonagon and a circle. Additionally, we show the resource images of the samples on the right side.

samples have the same underlying probability distribution, which means in our case, if they belong to the same shape category.

2.4.1 The Kolmogorov-Smirnov Test

We now recall the setting for the Kolmogorov-Smirnov hypothesis test, cf. [14] for more details.

The Kolmogorov-Smirnov test will give us the value for the hypothesis H which we here cast like

$$H_{i,j} = \begin{cases} 0 & \text{if } F_{D_i}^n = F_{D_j}^m \\ 1 & \text{if } F_{D_i}^n \neq F_{D_j}^m \end{cases} \tag{6}$$

where $F_{D_i}^n$ is the empirical cumulative distribution function for the sample D_i containing n elements. This means H is zero if both samples are drawn from the same distribution and 1 if not. You cannot generally assume that n and m are equal. Even the normalization process does not change this condition.

Thus, in our application, $H_{i,j} = 0$ indicates that the shapes described by the normalized signatures D_i and D_j are of the same category.

Letting for simplicity $D_i = (d_1, \ldots, d_n)$, we can calculate $F_{D_i}^n$ via

$$F_{D_i}^n(x) = \frac{1}{n} \sum_{j=1}^{n} \xi_{(-\infty,x]}(d_j) \tag{7}$$

where $\xi_{(-\infty,x]}(d_j)$ is the indicator function given by

$$\xi_{(-\infty,x]}(d_j) := \begin{cases} 1 & \text{if } d_j \in (-\infty,x] \\ 0 & \text{if } d_j \notin (-\infty,x] \end{cases} \tag{8}$$

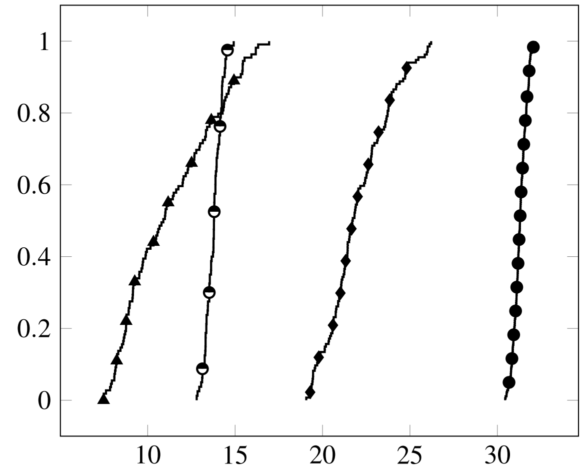


Figure 4: The empirical cumulative distribution function from the samples in Figure 3. The functions from the triangle shape is marked with triangles, the pentagon shape with diamonds, the nonagon with half filled circles and the circle is marked with circles. We can see the different slopes that will make the comparison possible.

Some examples of empirical distribution functions that arise given some shapes and their normalized signatures are presented in Figure 4.

We may then calculate the Kolmogorov-Smirnov statistics in general via

$$S_{n,m} = \sup_x \left| F^n_{D_i}(x) - F^m_{D_j}(x) \right| \tag{9}$$

After [11, eq. (15), section 3.3.1] [16, p. 402, theorem 9.8.3], the hypothesis is accepted if

$$S_{n,m} \leq c(\delta, n, m) = \sqrt{-\ln\left(\frac{\delta}{2}\right)\frac{n+m}{2nm}} \tag{10}$$

where $c(\delta,n,m)$ is the critical value for Kolmogorov-Smirnov test, which depends on the significance level δ and the sample sizes n and m.

Putting all things together, we thus obtain by Kolmogorov-Smirnov test:

$$H_{i,j} = \begin{cases} 0 & \text{for } S_{m,m} \leq \sqrt{-\ln\left(\frac{\delta}{2}\right)\frac{n+m}{2nm}} \\ 1 & \text{otherwise} \end{cases} \tag{11}$$

2.4.2 *Hit Rate*

If we test for the similarity of a signature D_i with a reference signature D_{ref} making use of the Kolmogorov-Smirnov test we get an result $H_{i,\text{ref}} \in \{0,1\}$. And now we want to quantify the test results when comparing a set of signatures with a reference signature. With N the total number of samples we can define the hit rate in the following manner

$$[0,1] \ni \text{hit rate} = 1 - \frac{1}{N}\sum_{i=1}^{N} H_{i,\text{ref}} \tag{12}$$

This is a natural definition, since if the Kolmogorov-Smirnov test failed we obtain a 1 and if the test succeeded we get a 0. Adding up these results and dividing by the number of samples we compared, we end up with the percentage value of failures. Subtracting this value from 1 finally gives us the value for successful Kolmogorov-Smirnov tests.

We expect that the hit rate is close to 1 if, for example, we compare all triangles with a reference triangle. Conversely, the value should be close to 0 if we compare the triangles with other geometrical shapes.

The hit rates for different shape types and different reference shapes will form a so called hit rate matrix, as seen in discussion of experiments. The rows of this matrix represent the different shape types and the columns represent the reference shape type. The grey value shading of the matrix entries will give a visual account of the hit rate.

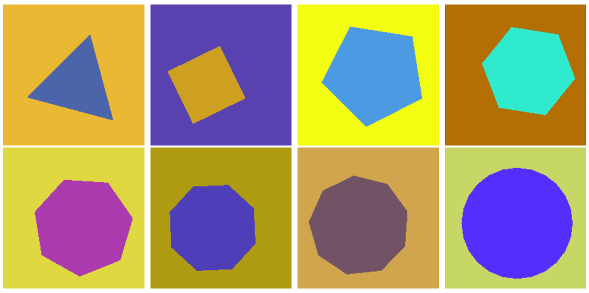


Figure 5: Examples of 8 different shapes one can find in the 2D image shape database [12]. From top left to bottom right we see a triangle, square, pentagon, hexagon, heptagon, octagon, nonagon, and a circle. All images have a resolution of 200 × 200 pixel.

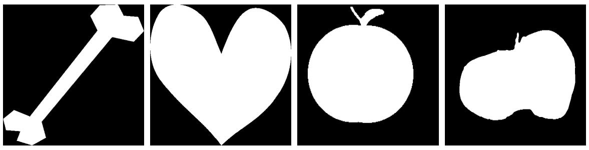


Figure 6: Selection from the MPEG-7 Core Experiment database [13]: pictures from a bone, heart, apple and a shoe.

3 EXPERIMENTS

Before we talk about the experiments we need to consider our database. After that we give a clear account of the general procedure when conducting the experiments and indicate how fast the method works. After that we give a detailed account of the experiments. For this we concentrate first on geometric shapes and proceed after that with a discussion of some additional experiments.

3.1 Database for the Experiments

We use two different databases. The *first database* is a large 2D geometrical shape database [12] containing 90,000 pictures of nine geometrical object. Taken from this we consider images of triangles, squares, pentagons, hexagons, heptagons, octagons, nonagons and circles. Each of these objects is given in total in 10,000 different sizes, rotations and positions.

It should be noted that all images of the triangle pictures the same triangle in different positions, size and rotation. This also applies to the other geometrical shapes. This means, the angles within the shapes do not vary, and they are not transformed in a non-rigid way.

Let us note that for our experiments we ignored the star shapes also contained in the Mendeley database [12] due to some difficulties encountered in the process of point cloud generation. Examples of the shape classes we mentioned used for our experiments can be seen in the Figure 5.

The *second database* we consider is the MPEG-7 Core Experiment database [13]. This database contains binary (black and white) images of variety of objects like apples, bat, Teddy's, birds and many more. We chose the apple, bone, cup, heart and shoe shapes for our experiments, which can be seen in Figure 6.

In the next step we need to generate point clouds out of all the considered pictures of the two databases. In the case of the MPEG-7 Core Experiment database, [5] gives access to the point clouds via github. For the 5 example shapes we chose, the point clouds contain between 102 and 118 points. For each shape there are 20 samples.

For the database of the geometric shapes we needed to build the point clouds by ourselves. The basic steps can be seen in Figure 1. For this process we first need to transform the color images into a binary format. With this in mind, we take the first color channel of a given picture and produce a binary image via the MATLAB function *imbinarize*. Out of this image we can extract a point cloud through the *bwboundaries* MATLAB routine. After that we end up with point clouds with round about 90 up to 700 points for the geometric figures.

The point clouds obtained in the one or other way are already ordered representing the neighbourhood relation. So we do not need to do some further preprocessing.

As one may observe at hand of Figure 7, the considered Mendeley shape database allows to study invariant properties of shape analysis methods that are important for tackling applications. These properties are the invariances with respect to translation, scaling and rotation, which are often considered as the most basic and fundamental properties of useful schemes.

It is quite obvious that the schemes we consider should give by construction invariant results with respect to translations and rotations of a given shape. These invariances are by construction given since the descriptors we study work by considering distances taken from the shape barycenters to boundary points, respectively, distances between points on the shape boundaries. These again are supposed to stay invariant under translation or rotation. The invariance with respect to scaling is addressed by the mean-scaling normalization.

3.2 Procedure and Processing Time

In this section we summarize the general procedure for our experiments, giving a clear account of our method.

1. Generate point clouds for the shapes and define a reference shape.
2. When applying adaptive sampling:
   Compute the adaptive area for every shape using equation (3)
3. Generate descriptor samples for all shapes using the functions (4) or (5).
4. Normalize the samples with respect to the reference shape sample. In the end, all samples should have the same mean value. This gives the shape signatures.
5. Compare the signatures to the reference signature via Kolmogorov-Smirnov test using equations (7), (9) and (11).
6. Compute the hit rate via equation (12).

Let us note that for performing the Kolmogorov-Smirnov test, there exist in several software packages builtin functions like: *kstest2* (MATLAB), *ks.test* (R) or *scipy.stats.ks_2samp* (Python with SciPy).

For all our experiments we used the first shape of a given group as the reference shape. Furthermore, we used the MATLAB builtin function for the Kolmogorov-Smirnov test. By default this test is implemented with a significance level of $\delta = 5\%$.

All calculations were executed with MATLAB R2021a installed on a computer with Intel Xeon W-2145 CPU and 62,4 GiB of memory.

Proof of Fastness

For processing $80,000$ points without adaptive sampling ≈ 50 seconds. That will be $\approx 6.25 \cdot 10^{-4}$ seconds per shape. With adaptive sampling we need ≈ 60 seconds, which will be $\approx 7.5 \cdot 10^{-4}$ seconds per shape. We conjecture that this is very fast and makes the method suitable for potential applications like a first similarity search in a large shape database.

3.3 Experiments and Discussion

We now proceed along the lines of several experiments that allow to point out several properties of our method.

For all the basic experiments discussed first, we followed the steps from the previous section but do not consider adaptive sampling or a coarsening of points yet, which will follow in subsequent experiments.

3.3.1 Basic Experiment with Geometric Figures

In Figure 7 we find the hit rate matrix for the geometrical Mendeley shape dataset. We see the diagonal structure in the hit rate matrix. Let us recall that this dataset contains the eight mentioned shapes in different sizes, rotations and positions. Because of the invariance properties of the method by construction, a strong diagonally dominant entry structure in the hit matrix has also been expected. However, it is surely a very reasonable result with respect to the use of the Kolmogorov-Smirnov test for classification. Let us also note that simple geometries like the triangle, square and heptagon are classified very robustly.

Let us comment on the less pronounced but visible off-diagonal entries here. The more corners are involved

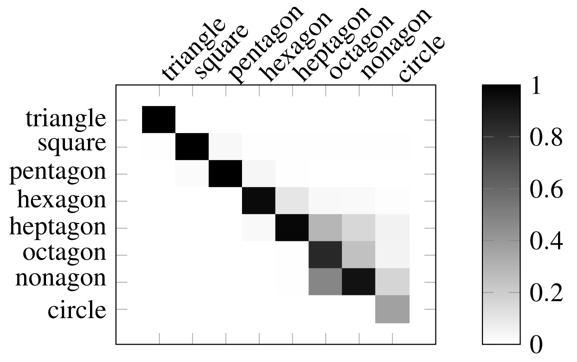


Figure 7: Hit rate matrix for an experiment without adaptive sampling. On the x-axis we wrote the reference shape type and the y-axis we put the different shape types. We see that our basic setup is invariant to rotation, translation and scaling.

in the geometric objects, the more they resemble each other already in the descriptor values, cf. Figure 3 and corresponding discussion. Therefore the basis for performing the Kolmogorov-Smirnov test becomes less discriminative which is reflected in the results.

3.3.2 Basic Experiments with Shapes from MPEG-7

In this experiment, we consider the shapes displayed in Figure 6 supplemented by the cup image as seen in Figure 10. Furthermore, we compare here the two distances d^1 and d^2 introduced in Section 2.3. The results are displayed in terms of the hit rate matrices in Figures 8 and 9.

For these non-geometric objects we may observe that the proceeding appears to be less discriminative in some cases. However, we conjecture that one can mainly notice the properties of the underlying distances used for building the descriptors. Let us note that in [15], the d^2 descriptor gives best results among the tested distances, while by Figure 9 it appears to be less discriminative.

3.3.3 Sampling Experiments

The motivation behind the study of adaptive sampling and a coarsening as we will apply here, can be summarized as follows. By adaptive sampling, salient points of a given shape can be stressed. We compute these points and add them to the list of points from which we sample. By coarsening, we take every second point of the boundary and add it again, which gives a uniform distribution of additional points. This is conceptually in contrast to stressing salient points and shows at the same time the influence of number of points in a shape. The Figure 10 gives an account of resulting points for the cup shape.

The main results of this study are depicted in Figure 11 and Figure 12. We compare the cup using the sampling strategies with the chosen shapes from MPEG-7.

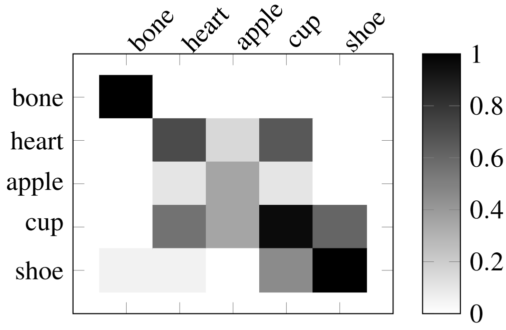


Figure 8: Hit rate matrix for a basic experiment without adaptive sampling. Here we used the d^1 distance function to produce samples with a sample size equal to number of points in the point cloud. The reference shape type is listed on the x-axis and on the y-axis we find the shape type. We notice the diagonal like structure of this plot.

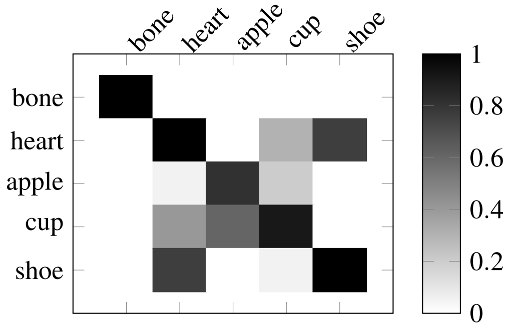


Figure 9: Hit rate matrix for a basic experiment without adaptive sampling, analogously to Figure 8 but using the d^2 distance. We observe more pronounced off-diagonal entries than the hit rate matrix obtained from the d^1 distance.

Exploring just the additional points from adaptive sampling, meaning that salient points are stressed within sampling, we observe that in most cases the accuracy of classification by Kolmogorov-Smirnov test declines. Turning to combination of adaptive and coarse sampling, we observe a similar effect of adaptivity, while

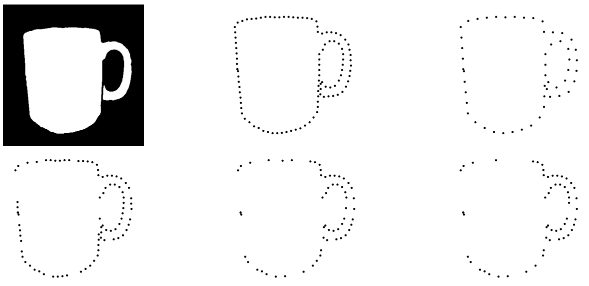


Figure 10: Different variations of the point cloud of the cup shape. **Top Left:** Cup Image. **Top Middle:** Point cloud of this shape. **Top Right:** Every second point. **Bottom Left:** Adaptive area sampling with $T = 0.02$. **Bottom Middle:** $T = 0.04$. **Bottom Right:** $T = 0.06$.

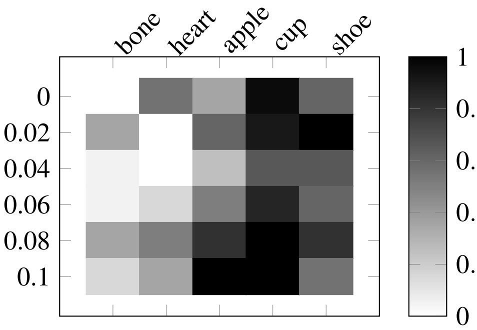


Figure 11: Cup only experiment. Hit rate matrix for the cup, exploring different adaptive sampling parameters. First row coincides with basic experiment.

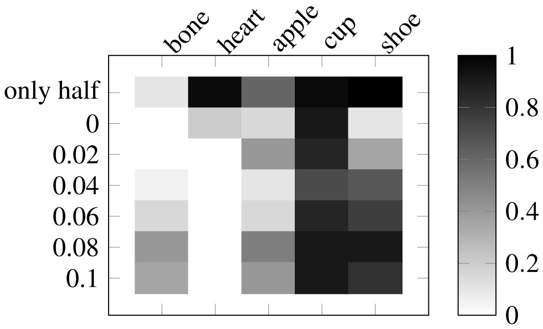


Figure 12: Cup only experiment. Here we combine the adaptive sampling with the coarse shape representation, adding both point sets for sampling to the original shape. Especially by results for choice $T = 0$ (see y-axis) we observe in the second row that adding a coarse shape representation gives favourable results over adding salient points to the sampling set.

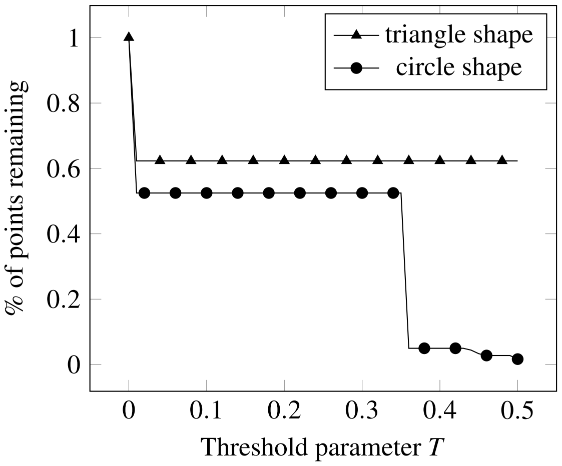


Figure 13: Study of point numbers when applying adaptive sampling, in dependence of area threshold parameter T

adding the coarsened version of the cup gives significant better results than for original cup only.

As a result, one may conjecture that a higher uniform density of shape boundary points is beneficial for the overall strategy. Let us note that this does not imply that offering just the original points several times to the sampling routine is to be favoured. Doing this will just lead to randomness of the complete result so that no useful classification can be performed.

Let us note that we refrain from performing the same study for geometric shapes as in Mendeley dataset. The reason for this may be observed via Figure 13. In geometric shapes with many straight lines, at certain stages (that depend on the way the shape boundaries are determined on the discrete pixel grid) the adaptive routine drops out many boundary points at once. Therefore, we opt to perform the study on sampling on the mentioned subset of MPEG-7.

4 CONCLUSION

In this paper a new method for shape classification is introduced based on Kolmogorov-Smirnov test. To do this, we propose an adaptation of the method from Osada and co-authers [15] and showed how to set up a useful framework. One may conclude that the Kolmogorov-Smirnov hypothesis test is well suitable for the task. The whole proceeding is computationally very efficient and may be explored, for instance, for a first quick search for similarities in a shape database.

One of the main construction points is the shape signature. We tested two underlying descriptors, and for future work we may infer that it could be reasonable to combine several descriptors for a joint inference.

By the discussion of the sampling experiments, we may deduce that the complexity of shapes tied to their resolution may be a major factor for the quality of classification results. In future work we aim to explore this point and consider a mathematical investigation of sampling strategies. Furthermore, in future work we will consider other datasets or create our own so that we can transfer our approach to 3D.

5 REFERENCES

[1] J. Arias-Nicolás and F. Calle-Alonso. Poster: A novel content-based image retrieval system based on bayesian logistic regression. *WSCG*, page 19, 2010.

[2] B. S. Bigham and S. Mazaheri. A survey on measurement metrics for shape matching based on similarity, scaling and spatial distance. In *The 7th International Conference on Contemporary Issues in Data Science*, pages 13–23. Springer, 2019.

[3] M. Breuß, A. Bruckstein, and P. Maragos. *Innovations for shape analysis: models and algorithms*. Springer Science & Business Media, 2013.

[4] M. Breuß, A. Bruckstein, P. Maragos, and S. Wuhrer. *Perspectives in Shape Analysis*. Springer, 2016.

[5] A. Carlier, K. Leonard, S. Hahmann, G. Morin, and M. Collins. The 2D shape structure dataset: A user annotated open access database. *Computers & Graphics*, 58:23–30, 2016. Shape Modeling International 2016.

[6] M. Dai, S. Kurtek, E. Klassen, and A. Srivastava. *Statistical Shape Analysis*, pages 1–16. Springer International Publishing, Cham, 2020.

[7] L. H. de Figueiredo. Adaptive sampling of parametric curves. In *Graphics Gems V*, pages 173–178. Elsevier, 1995.

[8] I. L. Dryden and K. V. Mardia. *Statistical shape analysis: with applications in R*, volume 995. John Wiley & Sons, 2016.

[9] H. Hufnagel, X. Pennec, J. Ehrhardt, H. Handels, and N. Ayache. Shape analysis using a point-based statistical shape model built on correspondence probabilities. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 959–967. Springer, 2007.

[10] M. S. Junayed, N. Anjum, A. Noman, and B. Islam. A deep cnn model for skin cancer detection and classification. *WSCG 2021: Full Papers Proceedings*, 2021.

[11] D. E. Knuth. *The Art of Computer Programming, Volume 2 (3^{rd} Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., USA, 1997.

[12] A. E. Korchi. 2D geometric shapes dataset, 2020. Mendeley Data, V1, doi: 10.17632/wzr2yv7r53.1.

[13] L. J. Latecki and R. Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000.

[14] E. L. Lehmann and J. P. Romano. *Testing Statistical Hypotheses*. Springer Texts in Statistics. Springer New York, New York, 2005.

[15] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3d models with shape distributions. In *Proceedings international conference on shape modeling and applications*, pages 154–166. IEEE, 2001.

[16] G. R. Shorack and J. A. Wellner. *Empirical Processes with Applications to Statistics*. John Wiley & Sons, Dec. 1986.

[17] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning, and testing. *IEEE Transactions on pattern analysis and machine intelligence*, 27(4):590–602, 2005.

[18] R. Sun and C. H. Lampert. Ks (conf): a lightweight test if a multiclass classifier operates outside of its specifications. *International Journal of Computer Vision*, 128(4):970–995, 2020.

[19] M. Yang, K. Kpalma, and J. Ronsin. A survey of shape feature extraction techniques. *Pattern recognition*, 15(7):43–90, 2008.

[20] T. Zawadzki, S. Nikiel, and E. Ribeiro. 3-d mesh-classification method based on angular histograms. *WSCG 2013: Poster Proceedings*, 2013.

[21] J. Žunić. Shape descriptors for image analysis. *Zbornik Radova*, (23):5–38, 2012.

Impact of PCA-based preprocessing and different CNN structures on deformable registration of sonograms

Christian Schmidt
Westfälische Hochschule
University of Applied Sciences
Neidenburger Strasse 43
45897 Gelsenkirchen
Germany
christian.schmidt@w-hs.de

Heinrich Martin Overhoff
Westfälische Hochschule
University of Applied Sciences
Neidenburger Strasse 43
45897 Gelsenkirchen
Germany
heinrich-martin.overhoff@w-hs.de

ABSTRACT

Central venous catheters (CVC) are commonly inserted into the large veins of the neck, e.g. the internal jugular vein (IJV). CVC insertion may cause serious complications like misplacement into an artery or perforation of cervical vessels. Placing a CVC under sonographic guidance is an appropriate method to reduce such adverse events, if anatomical landmarks like venous and arterial vessels can be detected reliably. This task shall be solved by registration of patient individual images vs. an anatomically labelled reference image. In this work, a linear, affine transformation is performed on cervical sonograms, followed by a non-linear transformation to achieve a more precise registration. Voxelmorph (VM), a learning-based library for deformable image registration using a convolutional neural network (CNN) with U-Net structure was used for non-linear transformation. The impact of principal component analysis (PCA)-based pre-denoising of patient individual images, as well as the impact of modified net structures with differing complexities on registration results were examined visually and quantitatively, the latter using metrics for deformation and image similarity. Using the PCA-approximated cervical sonograms resulted in decreased mean deformation lengths between 18% and 66% compared to their original image counterparts, depending on net structure. In addition, reducing the number of convolutional layers led to improved image similarity with PCA images, while worsening in original images. Despite a large reduction of network parameters, no overall decrease in registration quality was observed, leading to the conclusion that the original net structure is oversized for the task at hand.

Keywords

Medical image registration, deformable registration, sonograms, Voxelmorph, CNN

1 INTRODUCTION

Placement of a central venous catheter (CVC) is a procedure that carries risk for multiple complications, e.g., arterial puncture of the common carotid artery has an occurrence rate of $6\% - 9\%$ [Bee03]. This work aims to further improve the ultrasound guided CVC placement into the internal jugular vein (IJV) (Fig. 1), by detecting the IJV and indicating a needle target position in a manually acquired patient individual ultrasound image. The needle target position in such an image is to be determined by automated, computer-based analysis. It is assumed that an optimal needle target position is defined in a reference image. The task at hand is to map the optimal needle target position onto patient individual images. In order to realize this mapping, the patient individual images are to be registered vs. the reference image.

Overfitting occurs when a model learns the training data well, but does not generalize the acquired information to new data. This is an issue in medical machine learning applications, since these datasets are usually small compared to the complexity of deep neural network structures, or due to low signal-to-noise ratio in the data.

The hypothesis of this work is: An improvement of the signal-to-noise ratio in image data and a systematic reduction of network size can yield improved registration results with overall less deformation, and thus a more regular registration field. In this work, principal com-

https://www.doi.org/10.24132/CSRN.3201.23

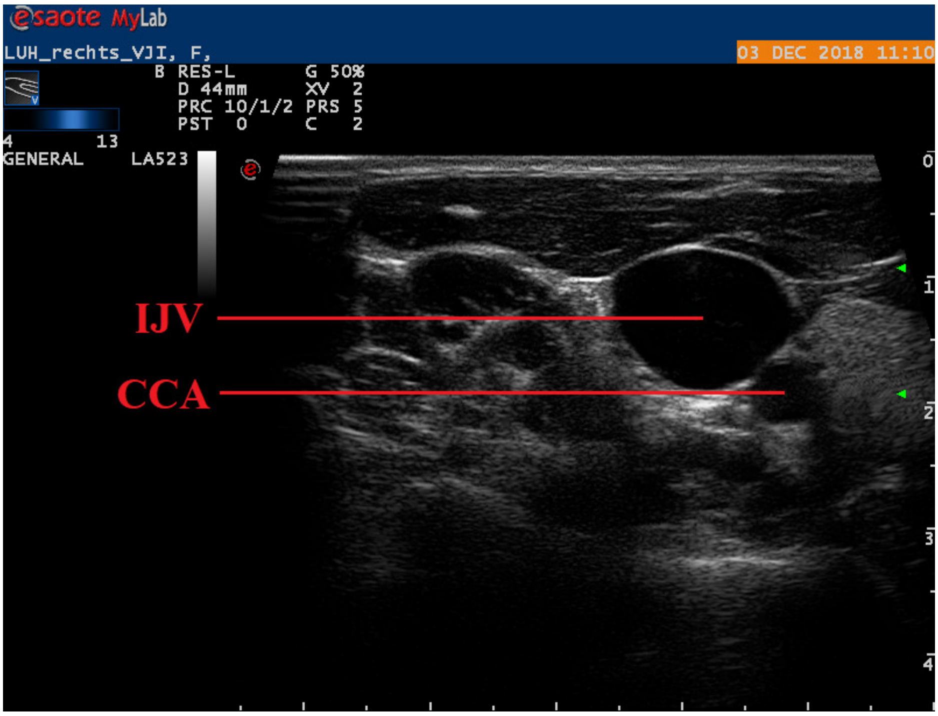


Figure 1: **Example image of an original cervical sonogram. Internal jugular vein (IJV) and common carotid artery (CCA) are labelled.**

ponent analysis (PCA) is used for noise reduction, and Voxelmorph, a U-Net-based convolutional neural network (CNN), is used as a reference network structure. To evaluate the hypothesis, three models for both image types are parameterized for a fine registration of affinely pre-registered ultrasound images of the human neck. The main tasks of this work are:

- reduce the size of original ultrasound images to a region of interest (ROI) that contains mainly the IJV. This shall be done by feature-based image segmentation. Subsequently, apply affine pre-registration to the ROI images. Because only few clinical images are available, and those have varying anatomical structures and image contrast, this procedure shall make the defomable image registration less error-prone.
- Perform a PCA on the image data set and approximate it by linear combination of the most relevant principal components.
- Train neural networks with three different structures and different number of free parameters to register image pairs (non-linear, deformable transformation). For each net structure, train two versions, one for original images, and one for PCA-approximated images.
- Quantitatively analyze the impact of the number of net parameters and the image type on the registration result, using evaluation metrics for deformation and image similarity.

2 RELATED WORK

Many different methods for deep learning-based medical image registration have been proposed in the past. For rigid transformations in particular, deep reinforcement learning (RL) techniques [Mni15] have gained some popularity. [Lia17] proposed a RL strategy for rigid 3D-3D registrations in computer tomography images, which is based on finding the optimal sequence of motion actions (rotations and translations) for image alignment. Since RL networks are constrained to low dimensionality of outputs, they have been used almost exclusively for rigid registrations, since those can be expressed by a small number of transformation parameters. With the rise of networks, which can directly estimate deformation vector fields (DVF) and are not constrained to rigid transformations, RL-based methods fell out of favor in recent years [FuY20].

Networks, which directly estimate the DVF can be classified into supervised and unsupervised methods. Supervised networks require ground truth transformations (either DVF, in case of deformable registration, or rigid transformation parameters). Ground truth transformations can be obtained by artificially de-aligning images with random rotations and translations [Sal19, Epp18] or by using traditional, non-learning methods to register image pairs and use the resulting DVFs as ground truth

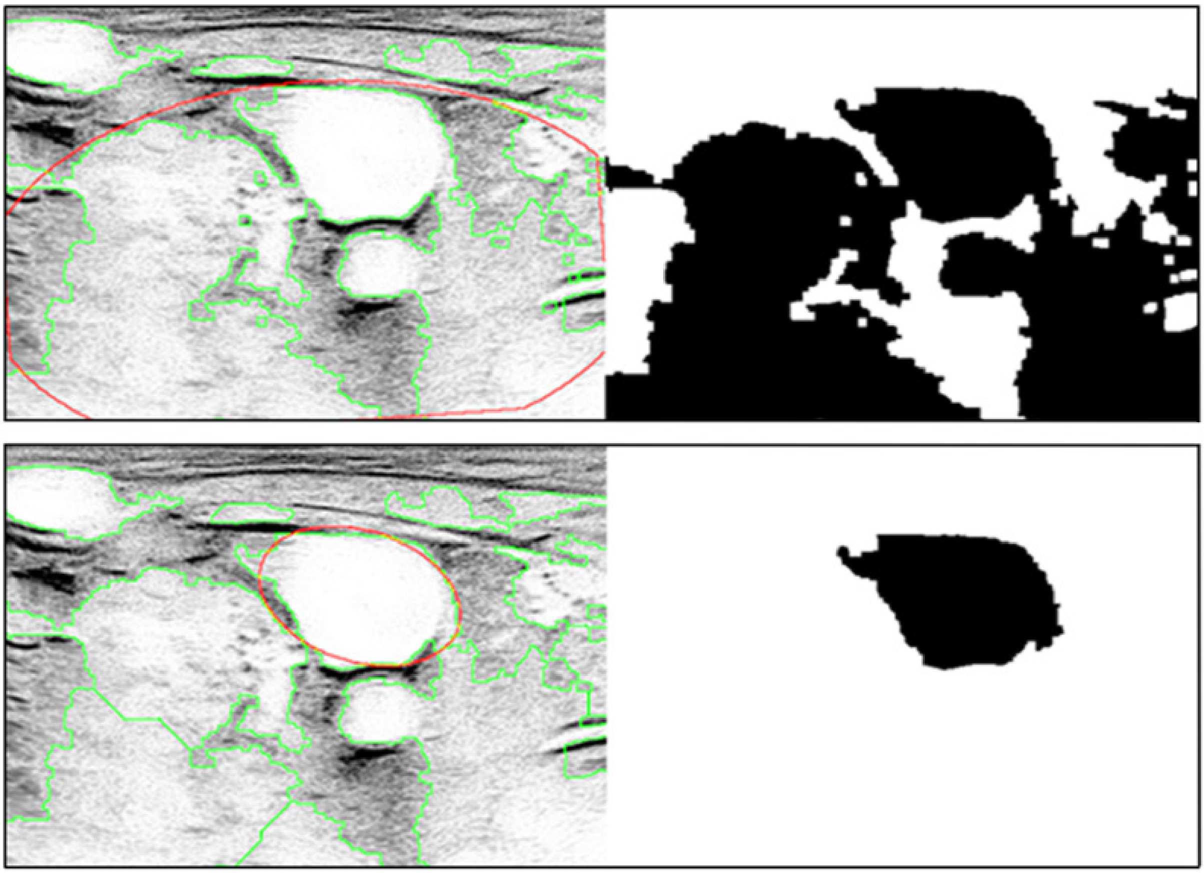


Figure 2: n **largest objects after binarization are delineated with green contours, the object identified as correct is marked with a red ellipse (left). Top row shows objects before, bottom row objects after watershed-transformation. Corresponding binary images after thresholding with g_{thresh} are shown on the right. Sonograms are dispayed with inverted grayscale pixel values for better visibility.**

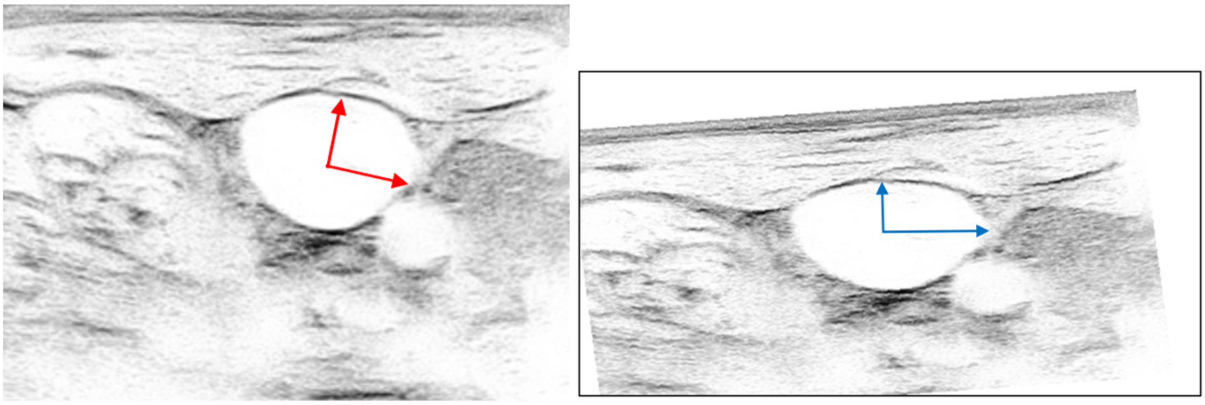


Figure 3: **Example sonogram of the IJV before (left) and after (right) affine transformation. Ellipse parameters resulting from the preceding feature-based segmentation, are used to translate ($\mathbf{x}_{OR}$), rotate ($-\varphi$) and scale (s_x, s_y) the images to coarsely pre-register them for the subsequent deformable deformation by the CNN.**

q	1	3	5	7	9	11	13	15	17	19
$cEVR$	0.12	0.31	0.46	0.55	0.62	0.67	0.71	0.75	0.77	0.79

Table 1: **Cumulative explained variance ratio $cEVR$ by number of first q principal components.**

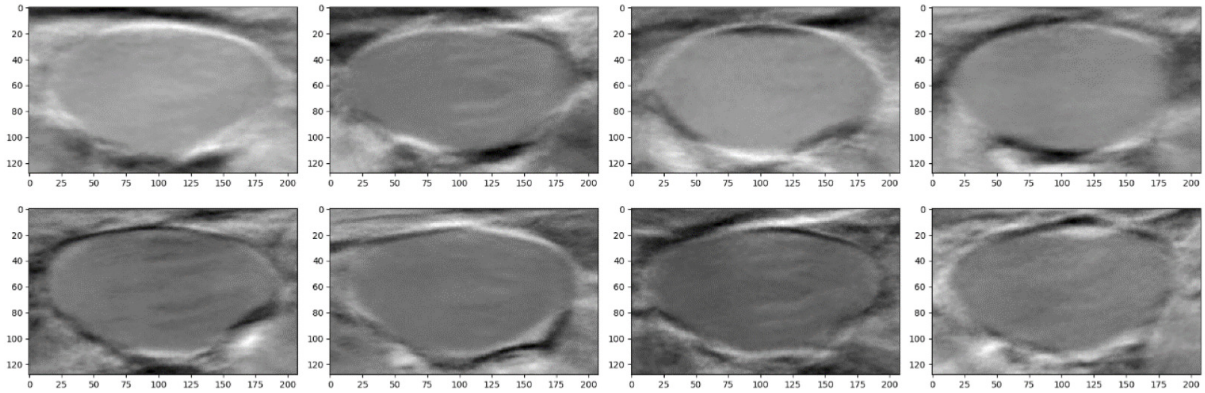


Figure 4: **First $q = 1 \dots 8$ principal component images $\mathbf{G}(\mathbf{y}_1)$ through $\mathbf{G}(\mathbf{y}_8)$ from performing a PCA on the ultrasound image data set (top row $\mathbf{G}(\mathbf{y}_1) \dots \mathbf{G}(\mathbf{y}_4)$, bottom row $\mathbf{G}(\mathbf{y}_5) \dots \mathbf{G}(\mathbf{y}_8)$).**

for training [Sen18]. Lack of medical datasets with known ground truth DVFs led to a rising demand for unsupervised networks. With the introduction of spatial transformer networks [Jad15], calculating image similarity losses was made possible during training. This is achieved by warping the estimated DVF with the input image and comparing the resulting image with the reference image. These networks do not require supervision by ground truth annotations and in addition to image similarity loss, employ a regularization loss term, to ensure smooth and anatomically plausible transformations [Zha18, Bal19].

Looking at image modalities, DL-based registration of ultrasound (US)-images only played a subordinate role in recent research, despite the high prevalence of sonography in clinical practice. A review of current publications in medical image registration [Bov20] found, that US-images were only used in about 5% of research papers with the topic of DL-based medical image registration, while magnetic resonance imaging (MRI) (52%) and computer tomography (CT) (19%) dominated the field. This is mainly due to higher availability of public MR training datasets and the fact, that the majority of papers examined registrations of brain images, which are predominantly recorded in MR and CT scans. US-images were most often used in multimodal registration tasks [HuY18, Yan18], in which, e.g., pre-procedural MR scans were aligned with intraprocedural US-images.

3 PROPOSED SOLUTION

Segmentation and affine pre-registration

Firstly, the original ultrasound image is being cropped and the interface of the ultrasound machine is removed. To segment the image into foreground and background, it is binarized using a binarization threshold value g_{thresh}. Subsequently, everything but the n largest objects are removed, to filter out structures that are too small to reasonably be considered. A watershed transform is implemented (Fig. 2), since as g_{thresh} increases, the increasingly forming clusters need to be separated, in order to be detected as individual entities. Choosing the correct IJV-object out of the remaining n, is done via the y-coordinate of the object's center and the distance between the common carotid artery (CCA) and the respective object. The object center's y-coordinates can be utilized because the distal location of the IJV is roughly the same for all subjects. Additionally, since CCA and IJV are in close anatomical proximity, all objects outside of a certain distance to the CCA can be excluded.

To obtain a smoother contour and an object which is geometrically parameterizable, the correctly identified object is approximated with an ellipse. The ellipse parameters major and minor axis length (a, b), and major axis rotation angle vs. the x-axis (φ) are extracted from the ellipse approximation. These parameters are subsequently used in the affine transformation (pre-registration).

This affine transformation between object IJV (O) and reference IJV (R) (Fig. 3) was performed as follows: At first, the image is translated by $\mathbf{x}_{OR}$, which aligns the center of the approximated ellipse with the image center. This is also the origin of the new reference coordinate system. Secondly, a rotation of $-\varphi$ degrees is applied to the image. This alignment of the center and rotation angle of the object IJV and the new reference coordinate system can be described as a 2D rigid body transform. Subsequently, image coordinate axes are scaled using the scaling factors s_x and s_y. Values for $s_x = \frac{a_R}{a_O}$ and $s_y = \frac{b_R}{b_O}$ are used to match the major and minor axes lengths of object vs. reference IJV ellipses. Finally, a rectangle of size 208 × 128 pixels around the object center is cropped, to only leave relevant parts of the image for later use as training data.

Principal Component Analysis

PCA is used to reduce dimensionality in the ultrasound image data set. For this purpose, $p = 81$ pre-registered ultrasound images of the human neck (transversal plane) from 14 different subjects (five to six images per subject) of size 208 × 128 pixels are investigated.

PCA is a statistical method, which is used for projecting a p-dimensional data set into a q-dimensional sub-set ($q < p$), while preserving characteristic data variability [Jol16, Kon17]. A data set consists of p variables x_i, $1 \leq i \leq p$, with n observations. Each variable x_i has a mean μ_i and a variance σ_i^2 calculated over its n observations. The sum over the variance of all variables is the total variance

$$\sigma_{\text{total}}^2 = \sum_{i=1}^{p} \sigma_i^2$$

Observations x_{mi} of variable x_i, $1 \leq m \leq n$, are noted as a $n \times 1$ vector $\mathbf{x}_i$. With the observed mean for vector $\mathbf{x}_i$ being $\boldsymbol{\mu}$, the PCA is calculated over centered observations $\mathbf{X}_i = \mathbf{x}_i - \boldsymbol{\mu}$. The eigenvalues λ_j of the co-variance matrix are indexed in descending order, they represent variances and fulfill

$$\sigma_{\text{total}}^2 = \sum_{j=1}^{p} \lambda_j$$

The PCA yields new variables $\mathbf{y}_j$, the so-called principal components (PCs). PCs are linear combinations of centered observations and have an identical co-variance matrix, i.e., the eigenvalues λ_j are the variances of variables $\mathbf{y}_j$. The total variance σ_{total}^2 is identical for PCs, original data, and centered observations, but is distributed differently among the variables. Goal of the PCA is to explain a major part of the total variance with a small number of variables q, the cumulative explained variance ratio ($cEVR$) is given by

$$cEVR = \frac{\sum_{j=1}^{q} \lambda_j}{\sigma_{\text{total}}^2}$$

The first q of all p new variables $\mathbf{y}_1 \ldots \mathbf{y}_q$ determine the data sub-set, such that

$$\mathbf{x}_i \approx \tilde{\mathbf{x}}_i = \sum_{j=1}^{q} \beta_{ij} \mathbf{y}_j + \boldsymbol{\mu}$$

To perform the PCA, pre-registered ultrasound images of the human neck $\mathbf{G}_i$ are reshaped into column vectors $\mathbf{x}(\mathbf{G}_i)$ with $n = 26624$ observations each. After dimensionality reduction, the above-described image vectors can be reorganized as images $\mathbf{G}(\tilde{\mathbf{x}}_i) \approx \mathbf{G}$. The first q PCs (Fig. 4) are used to approximate the original dataset. In the upcoming sections, $q = 8$ is used, which accounts for about 58% of the data's variance (Table 1) while reducing 90% of dimensionality (from $208 \times 128 \times 81$ to $208 \times 128 \times 8$).

Voxelmorph and variation of net structures

Voxelmorph (VM) [Bal19], a learning-based library for deformable image registration, which uses a U-Net-based [Ron15] net structure, is used to perform the deformable sonogram registrations. An atlas-based registration approach is used in this work; thus, an image pair consists of a varying patient individual image (moving image m) and a reference image (fixed image f).

Voxelmorph uses a two-part loss function

$$J = \mathscr{L}_{\text{sim}}(f, m \circ \phi) + \gamma \mathscr{L}_{\text{smooth}}(\phi),$$

which consists of a similarity term $\mathscr{L}_{\text{sim}}(f, m \circ \phi)$ and a deformation term $\mathscr{L}_{\text{smooth}}(\phi)$. The loss function J penalizes differences in grayscale values as well as deformations and is minimized by learning optimal convolutional kernels (filters). When registering an image pair, the network yields pixel-wise displacement vectors

$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \end{bmatrix} - \begin{bmatrix} x_m \\ y_m \end{bmatrix}, l = \|\mathbf{u}\|$$

between moving image m and fixed image f. The registration field

$$\phi = Id + \mathbf{u}$$

is formed by adding $\mathbf{u}$ to the identity transform. The resulting registration field ϕ generates a moved image $m \circ \phi$, which is similar to f.

By employing a regularization term, Voxelmorph encourages smooth, diffeomorphic deformations, i.e. deformations which are anatomically reasonable. γ serves as the regularization parameter, in this work we used $\gamma = 0.001$. As γ increases, deformation becomes more costly and the resulting deformation field, therefore, becomes more regular (smooth) and vice versa. The resulting registration field ϕ is applied to the moving image m by a spatial transformer function, to obtain the moved image $m \circ \phi$ (m warped by ϕ).

To examine the effects of reducing the number of free parameters in the CNN by cutting down its size, three net structures are introduced:

- The "full" structure proposed in the original VM paper, consisting of four encoder and seven decoder convolutional layers with 16 or 32 filters (convolutional kernels) each (16, 32, 32, 32 | 32, 32, 32, 32, 32, 16, 16). This net structure contains about 110,000 parameters.
- The "reduced" structure. Two encoder and decoder layers are removed for the second configuration (16, 32 | 32, 32, 32, 16 16), resulting in about 53,000 parameters, a reduction of 52% compared to the full net.

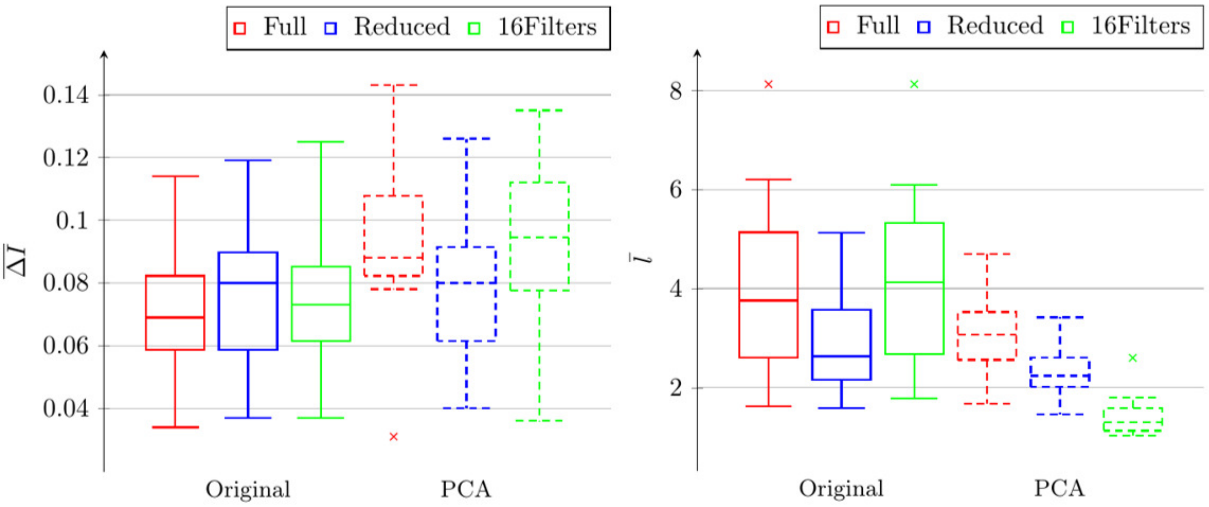


Figure 5: **Results of registrations with different net structures, using original and PCA-approximated images. Mean of differences of absolute grayscale intensities $\overline{\Delta I}$ is shown on the left, mean deformation vector length $\bar{l}$ on the right.**

- The "16 filters" structure contains all eleven convolutional layers, with 16 filters in each layer (16, 16, 16, 16 | 16, 16, 16, 16, 16, 16, 16), resulting in about 33,000 parameters, 70% less than the full net.

For each net structure, two versions are trained:

- Original image data vs. the reference image, i.e., $\mathbf{G}(\mathbf{x}_i) = \mathbf{G}(\tilde{\mathbf{x}}_i(q = 81))$ vs. $\mathbf{G}(\mathbf{x}_{(\text{Ref})}) = \mathbf{G}(\tilde{\mathbf{x}}_{(\text{Ref})}(q = 81))$
- PCA-approximated image set vs. the reference image, i.e., $\mathbf{G}(\mathbf{x}_i) = \mathbf{G}(\tilde{\mathbf{x}}_i(q = 8))$ vs. $\mathbf{G}(\mathbf{x}_{(\text{Ref})}) = \mathbf{G}(\tilde{\mathbf{x}}_{(\text{Ref})}(q = 81))$

In the upcoming results section however, PCA images are visually evaluated against the PCA-approximation $\mathbf{G}(\tilde{\mathbf{x}}_{(\text{Ref})}(q = 8))$ of the reference image to account for the difference in brightness and contrast between original and PCA images.

Quantitative analysis

To quantitatively analyze the properties and quality of performed registrations, two evaluation metrics are introduced:

- Let $0 \leq I \leq 1$ be the normalized image grayscale intensities, and $\Delta I = I(f) - I(m \circ \phi)$ the pixel-wise differences between intensities of fixed image f and moved image $m \circ \phi$. Therefore, $-1 \leq \Delta I \leq 1$ holds. We define $\overline{\Delta I}$ as the mean of absolute grayscale intensity differences ΔI.
- Mean deformation vector lengths $\bar{l}$ of the registration field ϕ, where l measures the pixel-wise deformation (in pixels) that is applied to the moving image m.

We use $\overline{\Delta I}$ and $\bar{l}$ analogously to the similarity term $\mathscr{L}_{\text{sim}}(f, m \circ \phi)$ and the deformation term $\mathscr{L}_{\text{smooth}}(\phi)$ of the Voxelmorph loss function. Since the region around the IJV's contour is of primary importance in this work, $\overline{\Delta I}$ and $\bar{l}$ are only evaluated in a belt-like along the IJV contour.

In addition, significances α of metric differences between the above mentioned net and image pair variants are determined with a two-tailed, paired t-test. We used a 70/30 split between training and test data, training the net's parameters on a NVIDIA GeForce RTX 2060 GPU takes 2.5 to 3 minutes, depending on net structure. Registering a single image pair takes 1 to 2 seconds.

4 RESULTS

Post-registration mean of absolute intensity differences $\overline{\Delta I}$ for all net structures and image types are shown in Fig. 5. With original images, an increase in the mean $\overline{\Delta I}$ of around 12% can be observed, when using the reduced net instead of the full net (mean $\overline{\Delta I}$: 0.078 vs. $0.070, \alpha = 0.027$). When registering PCA-approximated images however, a 17% decrease was measured when using the reduced over the full net structure (mean $\overline{\Delta I} : 0.079$ vs. $0.094, \alpha = 0.007$). Comparing the full net structure to their respective 16 filters version showed no significant change in mean $\overline{\Delta I}$.

Looking at mean deformation vector lengths $\bar{l}$ (Fig. 5), networks trained with PCA-approximations showed decreases in mean $\bar{l}$ vs. their original image counterpart of 24% for the full, 18% for the reduced and 66% for the 16 filters net structure. In addition, registrations with PCA-approximated images display the expected smoothing and noise reducing properties, removing unwanted artifacts from the vessel lumen (Fig. 6).

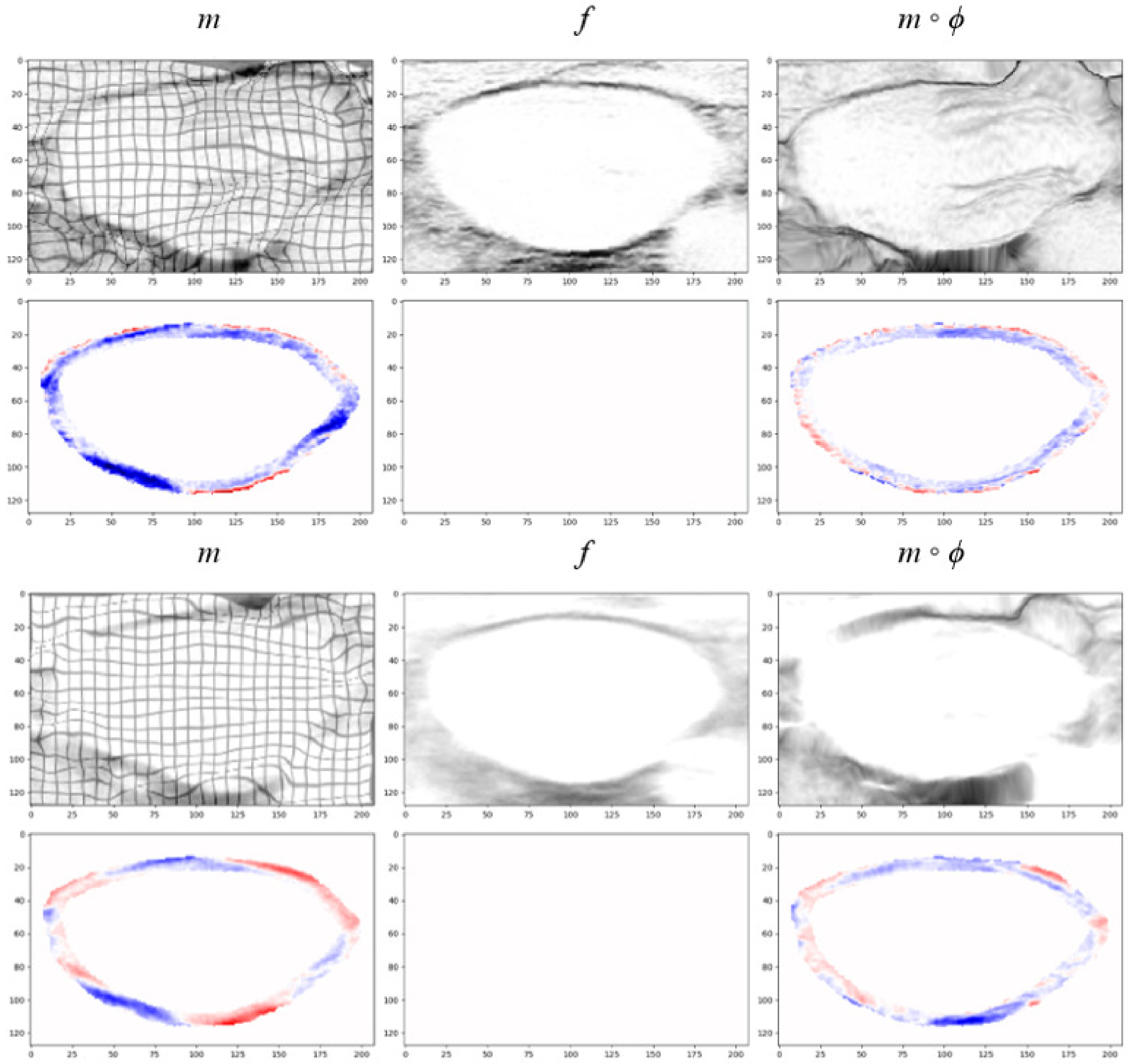


Figure 6: **Example registration results of original image (top) and PCA-approximation (bottom), using the full net structure. The registration field ϕ is warped with a regular square grid and superimposed over the moving image *m*, to show the extent and direction of local deformation which is applied to individual image parts. In addition, values of the pixel-wise pre- and post-registration grayscale difference ΔI are displayed color-coded, to illustrate the effect of registration on image similarities (colors ranging from dark red for $\Delta I = 1$ to dark blue for $\Delta I = -1$).**

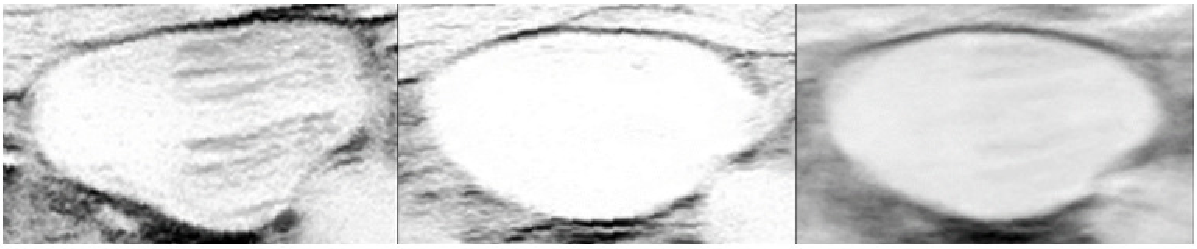


Figure 7: **Illustration of negative original image features being transferred to PCA-approximations. Reverberation artifacts of original image of subject A (left) appear in the PCA-approximation of subject B (right), even though no such artifacts are present in the original image of subject B (middle).**

5 CONCLUSION

Despite a reduction in net parameters of up to 70% compared to the originally proposed full net and reducing the mean deformation vector lengths $\bar{l}$ by 18% - 66%, no overall reduction in registration quality was measurable in the downscaled net structures. Specifically, for the combination of reduced net structure with PCA-approximated images, a significant decrease of $\bar{l}$ ($\bar{l} = 2.32$ vs. $2.85, \alpha = 0.045$) vs. original images was observed, while $\overline{\Delta l}$ remained nearly unchanged ($\overline{\Delta l} = 0.079$ vs. 0.078). This confirms the hypothesis described in the introduction section, and leads to the conclusion that the full net structure is unnecessarily oversized for the problem at hand.

The net structure can be reduced in size to diminish problems like overfitting, while also running up to 15% faster during training compared to the full net structure. In case of images which contain similar, regularly shaped structures, it is recommended to pre-process them with the proposed PCA procedure and employ reduced net structures, to reduce mean deformations and yield more regular registration fields. Since PCA is based on variances, it is highly sensitive to outliers. Thus, noisy images (outliers) in the original data set negatively affect the quality of the principal components, which then in turn affect the approximated PCA images (Fig. 7).

6 ACKNOWLEDGMENTS

We thank the Federal Ministry of Education and Research (BMBF) Germany, which funded this work in the program "Gründungen: Innovative Start-ups für Mensch-Technik-Interaktion", grant no. 16SV8153.

7 REFERENCES

[Bal19] Balakrishnan, G., Zhao, A., Sabuncu, M., Guttag, J., Dalca, A. (2019). VoxelMorph: A Learning Framework for Deformable Medical Image Registration. IEEE Transactions on Medical Imaging.

[Bee03] Beer, F. Preventing complications of central venous catheterization. The New England journal of medicine 348 (2003), 2684-6, author reply 2684.

[Bov20] Boveiri, HR., Khayami, R., Javidan, R., Mehdizadeh, A. (2020). Medical Image Registration Using Deep Neural Networks: A Comprehensive Review.

[Epp18] Eppenhof, K., Lafarge, M., Moeskops, P., Veta, M., Pluim, J., (2018). Deformable image registration using convolutional neural networks. 27. 10.1117/12.2292443.

[FuY20] Fu, Y., Lei, Y., Wang, T., Curran, WJ., Liu, T., Yang, X. Deep learning in medical image registration: a review. Phys Med Biol. 2020;65(20):20TR01. Published 2020 Oct 22

[HuY18] Hu, Y., Modat, M., Gibson, E., Li, W., Ghavami, N., Bonmati, E. et al. Weakly-supervised convolutional neural networks for multimodal image registration. Med Image Anal. 2018 Oct;49:1-13. doi: 10.1016/j.media.2018.07.002.

[Kon17] Kong, X., Hu, C., Duan, Z. Principal Component Analysis Networks and Algorithms. 1st. Springer Publishing Company, Incorporated, 2017.

[Jad15] Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K. (2015). Spatial Transformer Networks. Advances in Neural Information Processing Systems 28 (NIPS 2015).

[Jol16] Jolliffe, I., Cadima, J. Principal component analysis: A review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 374 (2016).

[Lia17] Liao, R., Miao, S., de Tournemire, P., Grbic, S., Kamen, A., Mansi, T., Comaniciu, D. (2017). An Artificial Agent for Robust Image Registration. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1).

[Mni15] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu AA, Veness J, Bellemare MG, Graves A. et al. Human-level control through deep reinforcement learning. Nature. 2015 Feb 26;518(7540):529-33. doi: 10.1038/nature14236. PMID: 25719670.

[Ron15] Ronneberger, O., Fischer, P., Brox. T. U-Net: Convolutional Networks for Biomedical Image Segmentation. LNCS 9351 (2015), pp. 234-241.

[Sal19] Salehi, M., Khan S, Erdogmus D, Gholipour A. Real-Time Deep Pose Estimation With Geodesic Loss for Image-to-Template Rigid Registration. IEEE Trans Med Imaging. 2019;38(2):470-481. doi:10.1109/TMI.2018.2866442

[Sen18] Sentker, T., Madesta, F., Werner, R. (2018). GDL-FIRE4D: Deep Learning-Based Fast 4D CT Image Registration.10.1007/978-3-030-00928-1.

[Yan18] Yan, P., Xu, S., Rastinehad, A., Wood, B. (2018). Adversarial Image Registration with Application for MR and TRUS Image Fusion.

[Zha18] Zhang, Jun. (2018). Inverse-Consistent Deep Networks for Unsupervised Deformable Image Registration.

Light Direction Reconstruction Analysis and Improvement using XAI and CG

Markus Miller[1] markus.miller@hm.edu http://orcid.org/0000-0001-9571-5997

Stefan Ronczka[1, 3] stefan.ronczka@hm.edu http://orcid.org/0000-0001-6738-9127

Alfred Nischwitz[1] nischwitz@cs.hm.edu http://orcid.org/0000-0003-3826-5584

Rüdiger Westermann[2] westermann@tum.de http://orcid.org/0000-0002-3394-0731

[1] *Dept. of Computer Science and Mathematics, University of Applied Sciences Munich, Lothstr. 64, D-80335, Munich, Bavaria*

[2] *Chair of Computer Graphics and Visualization, Technical University Munich, Boltzmannstr. 3/II, D-85748, Garching, Bavaria*

[3] *w&co MediaServices GmbH & Co KG, Charles-de-Gaulle-Str. 8, D-81737, Munich, Bavaria*

ABSTRACT

With rapid advances in the field of deep learning, explainable artificial intelligence (XAI) methods were introduced to gain insight into internal procedures of deep neural networks. Information gathered by XAI methods can help to identify shortcomings in network architectures and image datasets. Recent studies, however, advise to handle XAI interpretations with care, as they can be unreliable. Due to this unreliability, this study uses meta information that is produced when applying XAI to enhance the architecture – and thus the prediction performance – of a recently published regression model. This model aimed to contribute to solving the photometric registration problem in the field of augmented reality by regressing the dominant light direction in a scene. Bypassing misleading XAI interpretations, the influence of synthetic training data, generated with different rendering techniques, is furthermore evaluated empirically. In conclusion, this study demonstrates how the prediction performance of the recently published model can be increased by improving the network architecture and training dataset.

Keywords

Light, direction, estimation, reconstruction, explainable AI, photometric, registration, deep learning.

1 INTRODUCTION

After the tremendous progress of deep learning (DL) research in the past decade, gathering information on how deep neural networks (DNNs) make decisions from given input is gaining importance, as it may help to identify weaknesses and flaws in datasets or network architectures. This specific knowledge constitutes the foundation of certification processes in security- or safety-critical applications.

Therefore, in the past years, several explainable artificial intelligence (XAI) methods to achieve a human-comprehensible explanation of a DNN's decision process have been introduced, such as class activation mapping (CAM), gradient-weighted CAM (GradCAM), local interpretable model-agnostic explanations (LIME) and layer-wise relevance propagation (LRP). Both CAM (Zhou et al., 2016) and GradCAM (Selvaraju et al., 2020) result in heat maps showing what is considered important in an input image by a network for a specific inferencing task. However, CAM requires changes to a network's architecture, which renders this adjusted network incomparable to its previous architecture. LIME (Ribeiro et al., 2016) uses sampling masks to manipulate elements in the input images so that the influence of a specific element on the output function of a network can be estimated. As a sampling-based approach, LIME requires high-resolution masks to yield meaningful results when investigating filigree feature structures in the input images. LRP (Bach et al., 2015) propagates the relevance from the output all the way back to the input layer and generates a relevance map, highlighting the pixels in the input images that contributed most to a network's output decision. Most approaches deploy XAI methods to investigate classification problems. In an approach to count leaves of plants (Dobrescu et al., 2019), LRP is used to investigate how the number of counted leaves is derived from a given photograph.

Applying these XAI methods, we present an analysis of the reconstruction process of our DNN Net$_{s_x,s_y}$, which was proposed in an earlier publication (Miller et al., 2021) to predict the dominant light direction of a scene in stereographic coordinates, and derive architectural adjustments from it. We further investi-

https://www.doi.org/10.24132/CSRN.3201.24

gate the influence of computer graphics (CG) rendering techniques used in synthetic training data, such as different shading models (Blinn, 1977; Cook and Torrance, 1981; Lambert, 1760; Oren and Nayar, 1994) and shadow algorithms (Boksansky et al., 2019; Fernando, 2005; Williams, 1978), on the reconstruction performance and derive recommendations to generate synthetic training data.

The main contributions of this work can be summarized as follows:

- Insights on how to improve the reconstruction results of our DNN Net_{s_x,s_y} when regressing the dominant light direction from real scene images using XAI.
- Reduction of the reconstruction error by optimising the net architecture.
- Reduction of the reconstruction error by optimising the training dataset.

2 RELATED WORK

Previous work (Miller et al., 2021) showed that the dominant light direction of a real scene can be estimated more accurately from red-green-blue (RGB) images by using a stereographic coordinate representation of the dominant light direction, resulting in the stereographically predicting neural network Net_{s_x,s_y}. This network, trained on synthetically generated images, achieved an average angular reconstruction error $\overline{E_\measuredangle}$ of 3.7° on synthetic reference test data T_{SYN}, as well as 25.5° on real reference test data T_{REAL}, which could be improved to 7.1° by training Net_{s_x,s_y} on mixed data (99.2 percent synthetic and 0.8 percent real training images). The higher reconstruction error on real test data was assumed to be caused by a notable domain gap between synthetic and real datasets.

One possibility to improve the reconstruction performance achieved by Net_{s_x,s_y} is to analyse its decision process using an XAI model (such as LRP or GradCAM) and derive architecture optimisations from any insight gained.

LRP (Bach et al., 2015), when applied to a DNN, propagates the relevance, which is the contribution of a pixel or hidden neuron to the predicted output value, layer by layer from the output layer back to the input layer. Between two consecutive layers, relevance distribution is associated with the connections between each layer's neurons, following a given distribution rule. The distribution rule determines how relevance, i. e. positive or negative contributions to the regression result, or combinations of both, is being propagated, which may allow the importance of specific features in an input image to be investigated.

To analyse a given layer, as a first step, GradCAM (Selvaraju et al., 2020) computes weighted feature maps by scaling each feature map in the layer with its average gradient. Those weighted feature maps are then combined into a layer saliency map. By upscaling the layer saliency map to the input resolution and overlaying the input image with it, sensitive regions in the input image can be highlighted. Unlike LRP, GradCAM analyses one layer at a time.

Sanity checks (Adebayo et al., 2018) were introduced to gain intuition on how reliable explanations of different XAI methods may be by applying randomisation tests for both model parameters, as well as data labels and comparing the changes in the produced saliency maps. According to Adebayo et al. (2018), visual inspection of explanations alone may result in misleading conclusions. An extending study (Sixt et al., 2020) concludes that the gradient of most back-propagation based XAI approaches, such as LRP with certain relevance distribution rules, converges to a rank-1 matrix, which is why saliency maps of those approaches tend to highlight features of rather shallow network layers, not sufficiently showing decisions in deeper layers. Hence, despite the benefits XAI methods may provide, they also need to be handled with care.

Though not referring to this, in an approach to count the leaves of a plant (Dobrescu et al., 2019), LRP is applied to analyse a VGG-16 DNN, similar to Net_{s_x,s_y}, when regressing the number of leaves in a given image. By investigating the features extracted by the convolutional section and other experiments, such as manually covering leaves, it was concluded that the investigated DNN has indeed learned to regress the number of leaves from actual depictions of leaves. However, the content displayed in the investigated features was unhesitantly accepted, disregarding a potential unreliability.

Another possibility to improve the reconstruction performance is to optimise the dataset used for training, in particular the synthetic dataset, by investigating the influence of CG rendering techniques on the reconstruction result and tailoring a well-performing training dataset.

When optimising the training datasets to predict illumination situations, CG rendering techniques responsible for illumination and shadows are most relevant. Most basic illumination is achieved by applying the Lambertian reflection model (Lambert, 1760), which creates diffusely illuminated surfaces. The Phong-Blinn illumination model (Blinn, 1977) adds specular highlights to Lambertian reflecting surfaces by incorporating a half-way vector between light and view direction into the illumination computation. Extending the Lambertian reflection model, a more realistic diffuse illumination was achieved by assuming surfaces consisted of microfacets (Oren and Nayar, 1994), modelling surface roughness with a probability function. By taking physical models for refraction, roughness and self-shadowing into account, specular reflections could be improved to appear more realistic (Cook and Torrance, 1981). Since local illumination models disregard global phenomena like shadows, shadow mapping (Williams, 1978) introduced the ability to add hard shadows to a CG scene, independent from the used illumination model, by comparing computed depth values to values sampled from a previously generated depth or shadow map. The percentage closer soft shadows (PCSS) approach (Fernando, 2005) introduced soft shadows with variable penumbra by taking the distance between the shaded surface and blocker into account. When hardware acceleration for ray tracing became widely available, algorithms (Boksansky et al., 2019) for both hard and soft ray traced shadows could be incorporated into real-time applications using APIs, such as Vulkan or NVidia OptiX (Parker et al., 2010).

The presented study uses available XAI methods and tailored datasets to further improve the prediction performance achieved by Net$_{s_x,s_y}$.

3 APPLYING EXPLAINABLE AI

The architecture of Net$_{s_x,s_y}$ (Fig. 1) inherits the convolutional section of the VGG-16 (Simonyan and Zisserman, 2014) architecture, initialized with ImageNet (Russakovsky et al., 2015) pre-trained weights. Convolutional block C_5 was unlocked for training to adjust to the regression task, so its weights have changed. C_5 is followed by a custom fully connected (FC) block, consisting of a single FC layer L_h and a linear output layer L_o. L_h contains 4,096 neurons, is activated by a rectified linear unit (ReLU) function and uses a dropout value of 0.25. L_o consists of two output neurons without an activation function to regress stereographic coordinates s_x and s_y, representing the dominant light direction. Net$_{s_x,s_y}$ was trained using Adam as the optimizer, a batch size of 32 and a uniform learning rate of $1\mathrm{e}-3$[1].

In order to improve the prediction results of Net$_{s_x,s_y}$, a deeper understanding of its internal function and decision process may be helpful. However, additional architecture elements, as required by CAM, might notably change the reconstruction performance of Net$_{s_x,s_y}$, as well as the net itself, which is why CAM cannot be reasonably applied. Initial evaluation of LIME indicated that a fine-grained sampling mask would be required to produce meaningful explanations, which requires impractically high computing time even on high-performance computers.

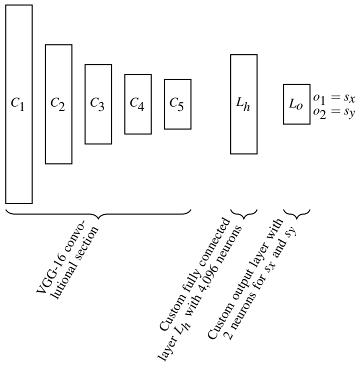

C_1
C_2
C_3
C_4
C_5
L_h
L_o
$o_1 = s_x$
$o_2 = s_y$
VGG-16 convo-lutional section
Custom fully connected layer L_h with 4,096 neurons
Custom output layer with 2 neurons for s_x and s_y

Figure 1: Diagram of the architecture of Net$_{s_x,s_y}$.

LRP and GradCAM neither require high computing power, nor changes to the architecture when investigating a network. Thus, LRP and GradCAM are applied to analyse Net$_{s_x,s_y}$.

Ideally, XAI methods yield easily interpretable saliency maps, identifying distinct image regions in the input, such as surface shading or shadow edges. Those image regions, when changed, may significantly influence the reconstruction performance. Consequently, to investigate positive and negative contributions to the regression result, when applying LRP, the relevance is distributed with the $\alpha\beta$-rule (Bach et al., 2015)

$$R_i = \sum_j \left(\alpha \cdot \frac{(a_i w_{ij})^+}{\sum_i (a_i w_{ij})^+} - \beta \cdot \frac{(a_i w_{ij})^-}{\sum_i (a_i w_{ij})^-} \right) R_j \quad (1)$$

with $\alpha = 1$ and $\beta = 0$ for positive, or $\alpha = 0$ and $\beta = 1$ for negative contributions, respectively. When propagating the relevance back through a network using the $\alpha\beta$-rule, the relevance R_i of a particular neuron i in the target layer is computed by proportionally summing up the relevance R_j of each neuron j in the source layer that it is connected to. The proportion to which each R_j contributes to R_i is given as the quotient of the connection contribution, i. e. the product of i's activation value a_i and the connection weight w_{ij} between neurons i and j, and the sum over all connection contributions between neuron j and any neuron in the target layer. This proportion is then separated into a positive and negative partial sum, indicated by superscript plus sign and minus sign, respectively. However, distributing the relevance with the $\alpha\beta$-rule causes LRP to converge to a rank-1 matrix (Sixt et al., 2020) and thus may not reliably provide insight into the decision process of Net$_{s_x,s_y}$. GradCAM as a gradient-based XAI method may provide valid insights, as it is not affected by this issue.

Interpreting image regions highlighted by the saliency maps of LRP or GradCAM remains difficult, nonetheless, as Net$_{s_x,s_y}$ does not predict discrete classes, but continuous values of s_x and s_y, denoting the dominant

[1] Given as uniform learning rate of 1 in Miller et al. (2021), which means 0.001 when using Adam as the optimizer.

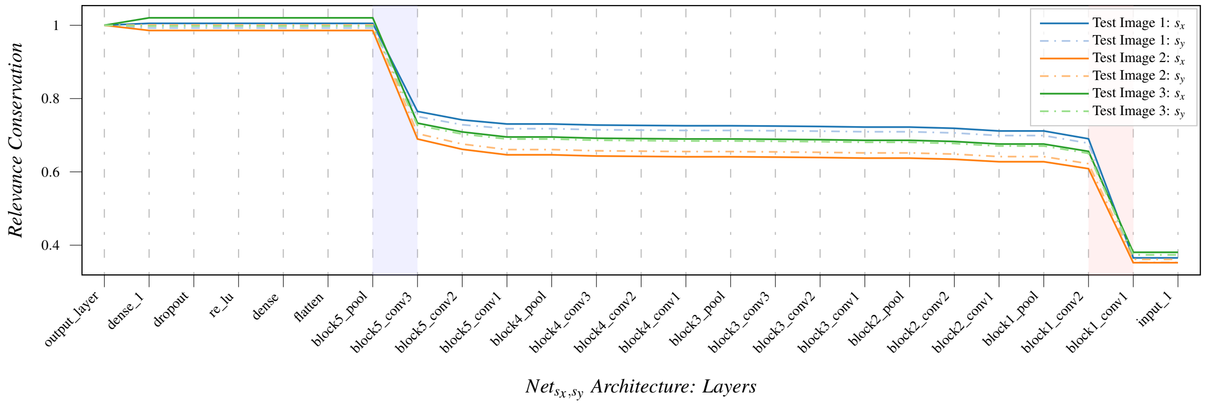


Figure 2: Relevance conservation of mixed trained Net_{s_x,s_y} for s_x (solid) and s_y (dash-dotted) neurons. Steep changes between layers indicate loss of relevance (highlighted), most likely due to the bias (note: the red highlight is inherent to the pre-trained VGG-16).

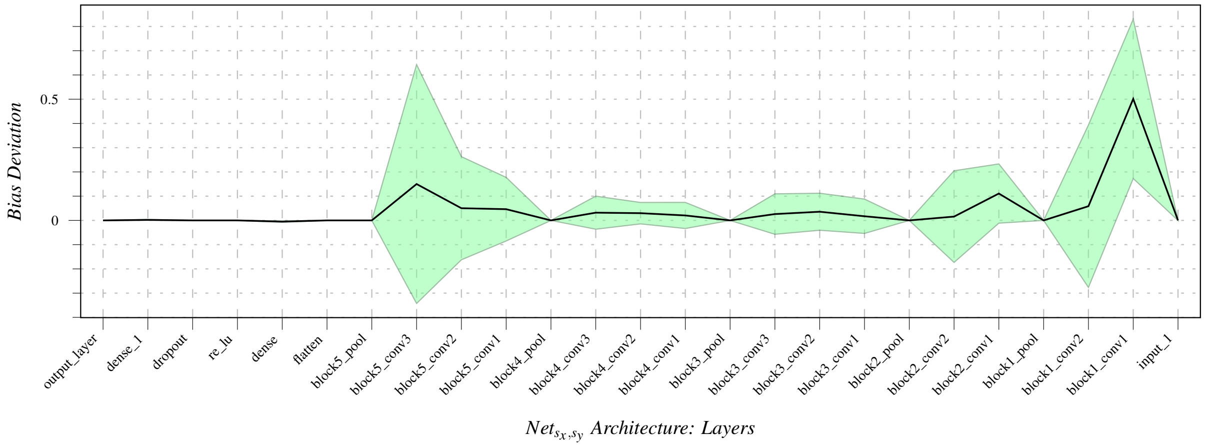


Figure 3: Bias deviation of mixed trained Net_{s_x,s_y}. The high bias deviation in *block5_conv3* may suggest the reason for the occurring relevance loss.

light direction in the scene. Different from classification, no statistical accumulation of saliency maps, which relates to a certain class, is formed due to the continuous values. Therefore, deriving statistical information, which image regions are important for a certain prediction, from saliency maps of single images, which constitute merely a momentary snapshot, cannot be considered reliable. A statistical evaluation, indicating how significant particular image regions are associated to a regression result, might accumulate saliency maps over a series of input images with same predictions and thus identify important image regions. However, this is most likely bound to fail as well due to the scene diversity (meaning different objects, light directions and camera directions) depicted in the images and the spatial variation of the image regions, leading to a map with scattered accumulated highlights not containing helpful information.

So, how can XAI be applied to analyse Net_{s_x,s_y} as a regression model? During the calculation of XAI methods, meta information is generated, such as relevance conservation when applying LRP, which can be used to gain intuition about the inner structure of a network. Relevance conservation means that relevance is assumed to be constant across the layers of a DNN and is a constraint of LRP computation. Inside a layer, the relevance is distributed to the bias and all neurons. However, only the relevance distributed to the latter is propagated to the next layer. Hence, jumps in a relevance conservation plot indicate a bias-heavy decision contribution in the affected layer. Initially, the convolutional section of Net_{s_x,s_y} was assumed to extract relevant features from the input image, and the FC block would connect this feature information into knowledge about the illumination situation. Analysing the relevance conservation of Net_{s_x,s_y} (Fig. 2), however, indicates a significant loss of relevance, particularly between the third convolutional and pooling layer of C_5 (*block5_conv3* and *block5_pool*), suggesting the weighted sum that is passed to the activation function is significantly influenced by bias. This is further supported by analysing the bias deviation in each layer (Fig. 3), showing a high bias deviation in *block5_conv3*. The bias deviation is analysed by plotting the deviation of bias values from the average bias value in each layer. The bias value of single bias layers, such as FC layers, are captured by the average bias value in that layer, denoting the actual bias value. Interpreting the graphs, the convolutional section is not only extracting features, but also appears to preselect important features in *block5_conv3* through bias values. Each feature map of *block5_conv3* maintains a dedicated bias value, which may suppress or dampen a feature map with negative or small bias values in favour of feature maps with a significant positive bias value when passed to the ReLU activation function. Due to this pre-selection, it appears that the FC block merely combines already existing information into regression values without providing new knowledge in that sense, leading to the hypothesis that the FC block may be replaceable with a linear layer. This hypothesis is investigated by replacing the FC block of Net_{s_x,s_y} with a single linear layer with two output neurons. The changed architecture (Fig. 4) is called linear feature aggregation network (LFAN).

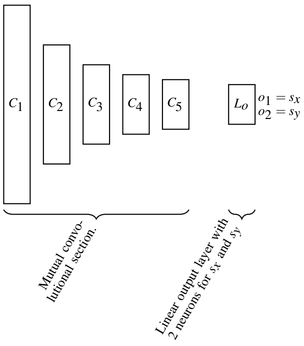

C_1
C_2
C_3
C_4
C_5
L_o
$o_1 = s_x$
$o_2 = s_y$
Mutual convolutional section.
Linear output layer with 2 neurons for s_x and s_y

Figure 4: Diagram of the LFAN architecture.

When analysing the regression of the amount of leaves in a given image, Dobrescu et al. (2019) interpreted the contents of features extracted by the convolutional section of their VGG-16 and concluded their DNN would count leaves in given images using intended information, though disregarding research (Adebayo et al., 2018; Sixt et al., 2020) recommending to handle results of XAI methods based on back-propagation with care. However, extracted features, computed by forward-propagating an image through the convolutional section of Net_{s_x,s_y}, choosing a particular feature map and propagating its relevance back through the net, may contain a different form of meta information, such as certain regions a feature map is sensitive to in the input image (Fig. 6). Seemingly, the convolutional section of Net_{s_x,s_y} extracts abstract chunks as features, containing small sections of the input scene. These abstract chunks show combinations of unrelated image content, such as partial shadow edges or fractions of shaded surfaces. When trying to predict the dominant light direction of a scene, considering a wide or even global area of the input scene may be beneficial, leading to the hypothesis that a deeper convolutional section may improve the regression result. This hypothesis is investigated by using a fully convolutional network (FCN), which adds two additional convolutional blocks, each consisting of one convolutional and one max-pooling layer, to the convolutional section and completes the network with a single linear layer with two output neurons (Fig. 5).

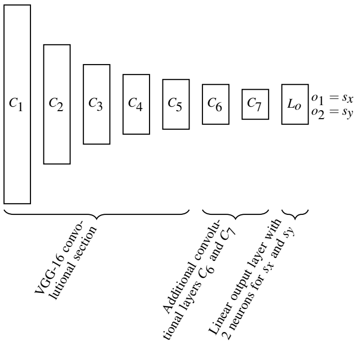

C_1
C_2
C_3
C_4
C_5
C_6
C_7
L_o
$o_1 = s_x$
$o_2 = s_y$
VGG-16 convolutional section
Additional convolutional layers C_6 and C_7
Linear output layer with 2 neurons for s_x and s_y

Figure 5: Diagram of the FCN architecture.

Analysing the extracted features this way does not rely on a subjective interpretation of feature content, but on neutral metrics, such as the receptive field of a certain feature map.

3.1 Training LFANs and FCNs

Since LFANs and FCNs share architecture elements with Net_{s_x,s_y}, synthetically and mixed pre-trained weights are available to be used for transfer learning. Hence, the same training process used for Net_{s_x,s_y} applies, meaning the training is split into hyperparameter-tuning (where needed) and fine-tuning. During hyperparameter-tuning, the convolutional section inherited from VGG-16 is disabled for training. Then, convolutional block C_5 is enabled for training during fine-tuning and the learning rate is reduced to a thousandth. Due to the FC block only being replaced by a single linear layer, LFAN training solely requires fine-tuning and is performed with the same hyperparameters used to train Net_{s_x,s_y} (Section 3, mentioned optimizer, batch size and learning rate). Training the FCNs requires hyperparameter-tuning prior to fine-tuning in order to find suitable parameter values, such as the number of filters in both the first and second added convolutional layer, whether to use a bias in these layers, the optimizer to use, the learning rate and the batch size. To tune the hyperparameters, the Bayesian optimisation module of Keras Tuner[2]

[2] After a comparison of Keras Tuner (https://github.com/keras-team/keras-tuner), Optuna (Akiba et al., 2019) and Talos (http://github.com/autonomio/talos), Keras Tuner was selected due to its tuning performance compared to the required tuning time.

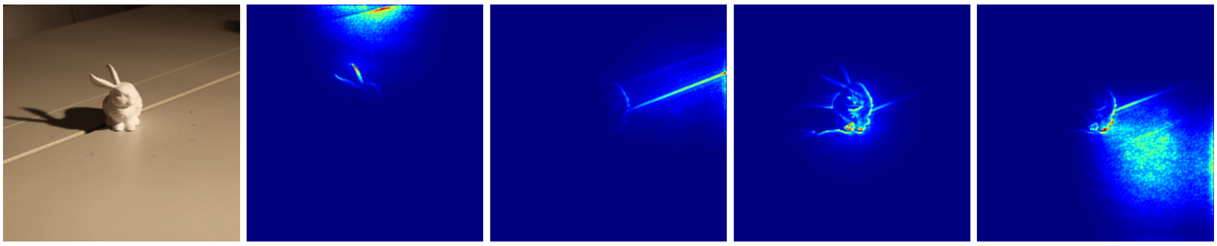


Figure 6: Given an input sample (left most), the convolutional section of mixed trained Net$_{s_x,s_y}$ extracts abstract feature chunk samples (remaining images).

Hyperparameter	FCN$_{\text{SYNTH}}$	FCN$_{\text{MIXED}}$
optimiser	Adam	RMSProp
learning rate	$1.5e-4$	$6.1e-5$
batch size	16	64
filters$_1$	682	1371
filters$_2$	1347	64
use_bias	false	false

Table 1: Best hyperparameter combinations for FCNs with synthetic (FCN$_{\text{SYNTH}}$) and mixed (FCN$_{\text{MIXED}}$) base weights of Net$_{s_x,s_y}$.

is used over 50 trials, training for five epochs each and identifying the best hyperparameter combinations (Table 1). Further, all trainings are conducted with the same datasets, synthetic and mixed, Miller et al. (2021) used to train Net$_{s_x,s_y}$ due to comparability. Additionally, the influence of data augmentation is investigated by conducting each training with enabled and disabled augmentation, i. e. the variation of image brightness, image translation and image zoom. All trained networks are predicting the dominant light direction in stereographic coordinates.

4 DATASETS

To investigate the influence of CG rendering techniques on the prediction performance of Net$_{s_x,s_y}$, DNNs using the same architecture and hyperparameters (Section 3, first paragraph) as Net$_{s_x,s_y}$ are trained on datasets with varying rendering techniques and tested on T$_{\text{SYN}}$ and T$_{\text{REAL}}$ (Section 2, first paragraph).

The datasets are generated with combinations of varying rendering techniques, each combination consisting of technique selections from three categories: diffuse reflections, specular reflections and shadows. Generating the datasets is realized in a dedicated application, using OpenGL and NVidia OptiX (Parker et al., 2010) and implementing rendering techniques for each category. As common CG rendering techniques for diffuse illumination, the Lambertian (Lambert, 1760) and Oren-Nayar (Oren and Nayar, 1994) reflectance models are implemented. Phong-Blinn (Blinn, 1977) and Cook-Torrance (Cook and Torrance, 1981) reflection models are implemented for specular reflections. Shadows of varying quality are evaluated by implementing plain shadow mapping (Williams, 1978) for hard shadows and PCSS (Fernando, 2005) for shadows with variable penumbra. When implementing PCSS, soft shadows are simulated by implementing percentage closer filtering (PCF), introduced by Reeves et al. (1987), applying a randomly rotated Poisson disc for each fragment when sampling the shadow map to avoid artefacts at the edges of the shadow. All shadow mapping techniques are implemented with an adaptive depth bias (Ehm et al., 2015) to avoid further shadow artefacts. Additionally to shadow mapping, ray tracing is used to render shadows. When implementing ray traced shadows, NVidia OptiX (Parker et al., 2010) is used, as de-noising is handled automatically by the framework. Both hard and soft ray traced shadows (Boksansky et al., 2019) are implemented by sending shadow rays from a possibly shaded surface location towards the light source. Since no real-time requirement applies to generate the datasets, the naive approaches are implemented. For hard shadows, caused by a point light source, one single shadow ray is cast; for soft shadows, up to 256 shadow rays are cast to randomly sample a spatially extended light source. Additionally, a naive approach for ambient occlusion (AO) with ray tracing (Nischwitz et al., 2019) is implemented by sampling the close proximity of a fragment with fixed length rays checking for hits with structures and reducing the light intensity proportionally.

Light directions and camera positions used in the generated datasets are similar[3] to the directions and positions used in the reference dataset presented by Miller et al. (2021) to train and test Net$_{s_x,s_y}$, ranging from $[0°, 360°[$ azimuth and $[5°, 90°]$ elevation in 5° steps each for the light direction distribution and in steps of 45° and 30° for azimuth and elevation of the camera positions, respectively.

To avoid an explosion of datasets and trainings required to be evaluated, the influence of the chosen reflection models is investigated systematically by starting out with shadowless combinations of Phong-Blinn and Lambert, Phong-Blinn and Oren-Nayar, and Cook-

[3] Identical, except for the starting value of the elevation angle, which is 5° instead of 1° due to visibility. Next iteration of camera positions is 30° instead of 35°.

Torrance and Lambert, varying the surface roughness with discrete values of 0.25, 0.5, 0.75 and 1.0 for combinations with either Oren-Nayar or Cook-Torrance. Two additional combinations using Cook-Torrance and Oren-Nayar with a roughness of 0.5 and 0.75 are being investigated, after the first evaluation, resulting in a total of 11 datasets with different illumination. Each of the illumination datasets is then rendered with a shadow algorithm: hard shadows with shadow mapping, hard shadows with ray tracing, soft shadows with PCSS, ray traced soft shadows with 256 shadow rays and ray traced soft shadows with ray traced AO (both sampled with 256 rays). This way, 55 datasets with illumination and shadows, as well as 11 datasets with illumination but without shadow, are generated and investigated, resulting in a total of 66 different datasets. The three datasets that perform the best on $\mathrm{T_{REAL}}$ are eventually mixed with the same small fraction of real data used in the mixed data to train Net_{s_x,s_y} and evaluated on $\mathrm{T_{SYN}}$ and $\mathrm{T_{REAL}}$ to determine the dataset with best performance overall.

5 RESULTS

To remain comparable to results achieved by Net_{s_x,s_y}, all introduced network architectures were tested with the same synthetic and real reference test dataset as used in Miller et al. (2021), aforementioned as $\mathrm{T_{SYN}}$ and $\mathrm{T_{REAL}}$.

The reference network Net_{s_x,s_y}, trained with mixed data, achieved an average angular error $\overline{E_\measuredangle}$ of 7.1° on $\mathrm{T_{REAL}}$. When being trained on synthetic data, Net_{s_x,s_y} achieved an error of $\overline{E_\measuredangle} = 3.7°$ on $\mathrm{T_{SYN}}$ and 25.5° on $\mathrm{T_{REAL}}$, respectively. No data was given for Net_{s_x,s_y} being tested on synthetic data and trained with mixed data (Table 2).

	$\mathrm{T_{SYN}}$	$\mathrm{T_{REAL}}$
SYNTH TRAINED	3.7°	25.5°
MIXED TRAINED	n/a	7.1°

Table 2: Net_{s_x,s_y} reference results.

Each architecture variant, LFAN and FCN, was trained on different base weights, i. e. the inherited convolutional section was initialized before training with different pre-trained weights: the weights of both the synthetically and mixed trained Net_{s_x,s_y}, and weights of an ImageNet pre-trained VGG-16. Using ImageNet pre-trained base weights did not show any improvement and thus are not displayed.

Investigating the LFAN architecture performance (Table 3), initializing the convolutional section with base weights of the synthetically trained Net_{s_x,s_y} improved the average angular error $\overline{E_\measuredangle}$ on $\mathrm{T_{REAL}}$ from 25.5° (previously achieved by the synthetically trained Net_{s_x,s_y}) to 22.8° when using synthetic data to train the LFAN. This could be further improved to 18.3° by training the LFAN with mixed data. On $\mathrm{T_{SYN}}$, these synthetically based LFANs achieved an error of 1.6° using synthetic and 2.4° with mixed training data. Using the base weights of mixed trained Net_{s_x,s_y}, the average angular error $\overline{E_\measuredangle} = 7.1°$ on $\mathrm{T_{REAL}}$ (achieved by mixed trained Net_{s_x,s_y}) could be improved to 6.8° when training the LFAN with synthetic data and to 6.2° with mixed training data. When tested on $\mathrm{T_{SYN}}$, similar values as before with 1.7° with synthetic and 2.4° with mixed training data are achieved.

	$\mathrm{T_{SYN}}$	$\mathrm{T_{REAL}}$	Base	Augm.
SYNTH TRAINED	1.6°	22.8°	SYNTH Net_{s_x,s_y}	NO
SYNTH TRAINED	1.7°	6.8°	MIXED Net_{s_x,s_y}	NO
MIXED TRAINED	2.4°	18.3°	SYNTH Net_{s_x,s_y}	NO
MIXED TRAINED	2.4°	6.2°	MIXED Net_{s_x,s_y}	NO

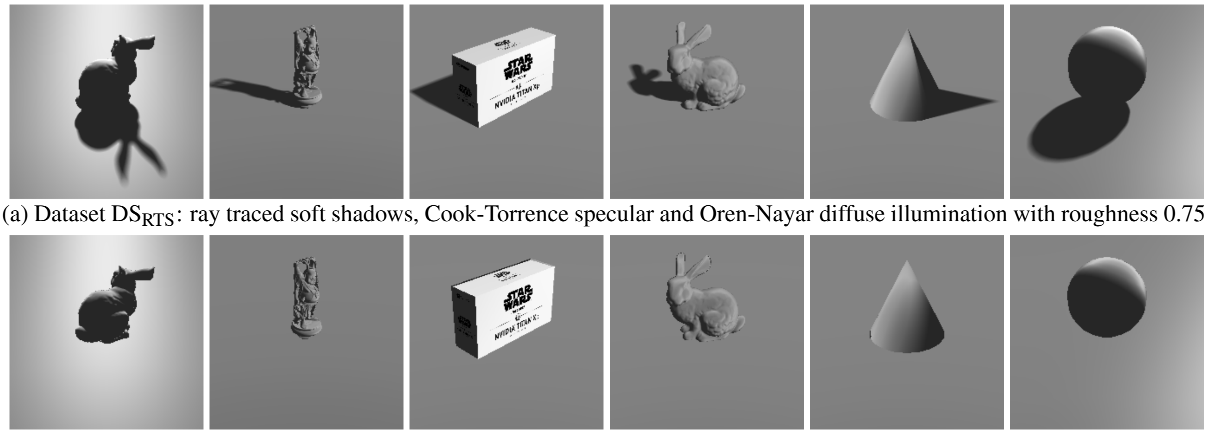
Table 3: LFAN evaluation results. Entries in the column *Augm.* indicate whether data augmentation was used.

	$\mathrm{T_{SYN}}$	$\mathrm{T_{REAL}}$	Base	Augm.
SYNTH TRAINED	1.4°	32.4°	SYNTH Net_{s_x,s_y}	NO
SYNTH TRAINED	1.4°	6.5°	MIXED Net_{s_x,s_y}	NO
MIXED TRAINED	1.3°	7.3°	SYNTH Net_{s_x,s_y}	NO
MIXED TRAINED	1.4°	5.7°	MIXED Net_{s_x,s_y}	NO

Table 4: FCN evaluation results. Again, entries in the column *Augm.* indicate whether data augmentation was used.

Compared to the synthetically trained Net_{s_x,s_y}, the prediction performance of the FCN architecture improves to $\overline{E_\measuredangle} = 1.4°$ from 3.7° on $\mathrm{T_{SYN}}$ using synthetic training data and synthetic base weights, but decreases from 25.5° to 32.4° on $\mathrm{T_{REAL}}$ (Table 4). Using synthetic base weights and mixed training data, similar behaviour is shown, as the prediction performance improves to 1.3° on $\mathrm{T_{SYN}}$ and declines to 7.3° on $\mathrm{T_{REAL}}$ from 7.1° originally. Contrary to that, when using mixed base weights, the FCN architecture achieves 1.4° on $\mathrm{T_{SYN}}$ with both synthetic and mixed training data. Also, using mixed base weights improves the prediction performance of this FCN variant with synthetic training data to 6.5° and with mixed training data to 5.7° on $\mathrm{T_{REAL}}$.

After investigating the various datasets, the dataset $\mathrm{DS_{RTS}}$ (Fig. 7a) using Cook-Torrance as specular, Oren-Nayar as diffuse reflection model and ray traced soft shadows without AO performed the best. Though not containing any shadow, dataset $\mathrm{DS_{NoS}}$ (Fig. 7b) using Cook-Torrance and Oren-Nayar without AO performed second best. To distinguish the models, the DNNs trained with $\mathrm{DS_{RTS}}$ and $\mathrm{DS_{NoS}}$ are named $\mathrm{Net_{RTS}}$ and $\mathrm{Net_{NoS}}$, respectively, though they use the same architecture and hyperparameters as Net_{s_x,s_y}, as well as ImageNet pre-trained weights. When trained solely on synthetic $\mathrm{DS_{RTS}}$ images, $\mathrm{Net_{RTS}}$ achieves an average angular error of $\overline{E_\measuredangle} = 20.2°$ on $\mathrm{T_{REAL}}$ and 35.5° on $\mathrm{T_{SYN}}$. Mixing $\mathrm{DS_{RTS}}$ with the small real training set



(a) Dataset DS$_{\text{RTS}}$: ray traced soft shadows, Cook-Torrence specular and Oren-Nayar diffuse illumination with roughness 0.75.

(b) Dataset DS$_{\text{NoS}}$: No shadows, Cook-Torrance specular and Oren-Nayar diffuse illumination with roughness 1.0.

Figure 7: Dataset samples of the two best performing datasets.

used by Miller et al. (2021), the thereby mixed trained Net$_{\text{RTS}}$ achieves an error of 4.4° on T$_{\text{REAL}}$ and 23.4° on T$_{\text{SYN}}$, indicating a significant domain gap between DS$_{\text{RTS}}$ and T$_{\text{SYN}}$ (Table 5). Similar behaviour is shown by DS$_{\text{NoS}}$, though accuracy is worse overall compared to DS$_{\text{RTS}}$. Net$_{\text{NoS}}$, when trained solely with synthetic images of DS$_{\text{NoS}}$, achieves an angular error of 20.2° on T$_{\text{REAL}}$ and 35.5° on T$_{\text{SYN}}$. Adding the small real training images to DS$_{\text{NoS}}$ and training Net$_{\text{NoS}}$ with this mixed dataset achieves an error of 5.7° on T$_{\text{REAL}}$ and 38.5° on T$_{\text{SYN}}$, again indicating a significant domain gap between DS$_{\text{NoS}}$ and T$_{\text{SYN}}$.

	T$_{\text{SYN}}$	T$_{\text{REAL}}$	Augm.
synth Net$_{\text{NoS}}$	35.5°	20.2°	yes
synth Net$_{\text{RTS}}$	32.1°	20.2°	yes
mixed Net$_{\text{NoS}}$	38.5°	5.7°	yes
mixed Net$_{\text{RTS}}$	23.4°	4.4°	yes

Table 5: Evaluation results of the dataset investigation. Prefixes in front of the network names indicate the used training dataset. Entries in the column *Augm.* indicate, whether data augmentation was used.

It is noteworthy that both LFAN and FCN architectures perform best exclusively without data augmentation, whereas DNN architectures with FC elements performed best with data augmentation enabled.

In summary, the previous state of the art with an error of 7.1° (mixed trained Net$_{s_x,s_y}$) on T$_{\text{REAL}}$ is improved to 6.2° with a mixed trained LFAN. It could be further improved to 5.7° with a mixed trained FCN. Both DNN architectures were pre-trained with mixed trained Net$_{s_x,s_y}$ base weights. Mixed trained Net$_{\text{RTS}}$ with an error of 4.4° achieves the best performance (Table 6).

6 DISCUSSION

In conclusion, this study demonstrates successful use of XAI meta information to systematically improve the prediction performance of the recently published regression model Net$_{s_x,s_y}$, which predicts the dominant light direction in a given scene, from an average angular error $\overline{E_{\measuredangle}} = 7.1°$ to an error of 6.2° using the presented LFAN architecture. Eventually, the improvement goes down to an error of 5.7° with FCNs on real reference test data T$_{\text{REAL}}$ by deriving architectural adjustments from aforementioned meta information.

	T$_{\text{REAL}}$
mixed Net$_{s_x,s_y}$	7.1°
mixed LFAN	6.2°
mixed FCN	5.7°
mixed Net$_{\text{RTS}}$	4.4°

Table 6: Summary of results gathered from tables 2 to 5.

An investigation of the influence of CG rendering techniques on the prediction result of Net$_{s_x,s_y}$ reveals that the dataset rendered with techniques that most accurately approximate reality, i. e. Oren-Nayar for diffuse, Cook-Torrence for specular illumination and ray traced soft shadows without the naive AO implementation, achieved the best result with the mixed trained Net$_{\text{RTS}}$ on the real reference test set T$_{\text{REAL}}$, achieving an average angular error $\overline{E_{\measuredangle}} = 4.4°$ and outperforming the mixed trained FCN using mixed base weights as best performing architecture adjusted DNN. Though DS$_{\text{NoS}}$ does not contain shadows, the reconstruction performance of mixed trained Net$_{\text{NoS}}$ is noteworthy, as this DNN may have learned to reduce the domain gap between DS$_{\text{NoS}}$ and T$_{\text{REAL}}$ from few training examples. Similar behaviour of Net$_{s_x,s_y}$ is presumably shown on T$_{\text{REAL}}$, since the edge between the two tables (Fig. 6, left most image) appears to be extracted by the convolutional section as a distinctive feature (Fig. 6, image in the middle). However, this interpretation may be inaccurate and misleading due to the findings of Adebayo et al. (2018) and Sixt et al. (2020).

While analysing features in different layers is a common and reasonable approach to optimise the prediction results of a network, deriving conclusions from meta information, such as relevance conservation and bias analysis, as shown in this work, appears to be unprecedented, as other approaches, despite analysing the conservation of relevance across the layers of a network, do not derive information in a similar way as described in this work.

However, a major drawback of the LFAN and FCN architectures are their inherent lack of regularisation, such as dropout, and thus their inherent possibility to overfit, which becomes most likely apparent in the FCN variant using synthetic base weights of Net$_{s_x,s_y}$ and synthetic training data, considering the angular average error $\overline{E}_{\measuredangle}$ of 1.4° on synthetic test data compared to an error of 32.4° on real test data. Additionally, the derived LFAN and FCN architectures are likely to be less robust when regressing from images with deviating brightness, as well as sufficiently non-centred or zoomed objects, as this appears to be too difficult when mapping the extracted features linearly to the output neurons. One indication for this is that both architectures perform worse when being trained with data augmentation enabled, which produces training images with according changes. Another indication is that, despite taking global features into account (Fig. 8), FCNs are affected, nonetheless.

A fundamental problem when applying LRP in the investigated situation occurs when investigating regressed values of $s_x, s_y = (0,0)$. Propagating a value of 0, meaning a relevance value $R_j = 0$ (eq. 1), would not yield a meaningful result, despite the fact that regressing values of $s_x, s_y = (0,0)$ are valid stereographic coordinates, denoting a light direction coming right from above in a given scene.

7 FUTURE WORK

Considering the improvements achieved by adding convolutional blocks to the FCN architecture, we intend to investigate whether applying attention-based DNNs may further improve the reconstruction performance.

Another opportunity for subsequent work is the investigation of possibilities to incorporate regularisation into the derived architectures and thus reduce the inherent potential to overfit. Furthermore, it is worth to investigate whether the FCN architecture may regain the ability to perform better when being trained with augmented data while maintaining its prediction performance when again adding a FC layer to map the extracted features to the output layer. Moreover, combining the FCNs architecture and further improvements to it with the datasets in this work may further improve the prediction results, too.

With larger real image data, containing more complex and diverse scenes, we will further investigate the robustness of the presented architectures.

Eventually, the presumed ability of Net$_{s_x,s_y}$ to generalise on unknown data (Section 6, end of second paragraph) may be investigated by applying further XAI methods.

ACKNOWLEDGEMENTS

We would like to thank the Competence Center Image Processing of the University of Applied Sciences in Munich[4] with all their funding companies[5], in particular w&co MediaServices GmbH & Co KG, as well as the Chair of Computer Graphics and Visualization[6] of the Technical University Munich for allowing us to conduct this research.

We would further like to thank IBM for their support and help. IBM kindly granted us access to their OpenPOWER[7] deep learning system.

Eventually, we would like to thank Simon Neureither for his assistance in creating synthetic datasets with varying CG techniques and Conrad Smith for his help with questions regarding linguistic matters.

REFERENCES

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. (2018). Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 9525 – 9536, Red Hook, NY, USA. Curran Associates Inc.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623 – 2631, New York, NY, USA. Association for Computing Machinery.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1 – 46.

Blinn, J. F. (1977). Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '77, pages 192 – 198, New York, NY, USA. Association for Computing Machinery.

[4] https://www.hm.edu/allgemein/forschung_entwicklung/forschungsfelder/competence_center/bildverarbeitung/index.de.html

[5] https://www.hm.edu/allgemein/forschung_entwicklung/forschungsfelder/competence_center/bildverarbeitung/partner.de.html

[6] https://www.in.tum.de/cg/

[7] https://openpower.ucc.in.tum.de

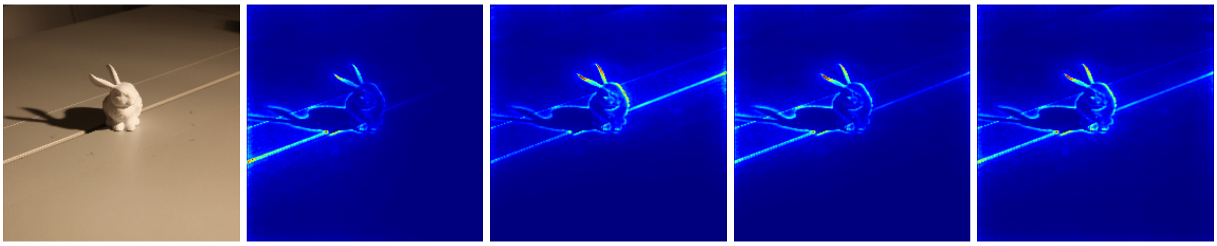


Figure 8: When investigating the extracted features from a given input image (left most), FCNs indeed take the entire image region into account (remaining images, all slightly different).

Boksansky, J., Wimmer, M., and Bittner, J. (2019). *Ray Traced Shadows: Maintaining Real-Time Frame Rates*, pages 159 – 182. Apress, Berkeley, CA.

Cook, R. L. and Torrance, K. E. (1981). A reflectance model for computer graphics. In *Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '81, pages 307 – 316, New York, NY, USA. Association for Computing Machinery.

Dobrescu, A., Giuffrida, M. V., and Tsaftaris, S. A. (2019). Understanding deep neural networks for regression in leaf counting. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2600 – 2608.

Ehm, A., Ederer, A., Klein, A., and Nischwitz, A. (2015). Adaptive depth bias for soft shadows. In *23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision: full papers proceedings*, pages 219 – 228. Computer Science Research Notes.

Fernando, R. (2005). Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches*, SIGGRAPH '05, page 35, New York, NY, USA. Association for Computing Machinery.

Lambert, J. H. (1760). *Photometria Sive De Mensura Et Gradibus Luminis, Colorum Et Umbrae*. Klett, Augsburg.

Miller, M., Nischwitz, A., and Westermann, R. (2021). Deep light direction reconstruction from single RGB images. In *29. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision: full papers proceedings*, pages 31 – 40. Computer Science Research Notes.

Nischwitz, A., Fischer, M., Haberäcker, P., and Socher, G. (2019). Schatten. In *Computergrafik: Band I des Standardwerks Computergrafik und Bildverarbeitung*, pages 480 – 554, Wiesbaden. Springer Fachmedien Wiesbaden.

Oren, M. and Nayar, S. K. (1994). Generalization of lambert's reflectance model. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 239 – 246, New York, NY, USA. Association for Computing Machinery.

Parker, S. G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., and Stich, M. (2010). Optix: A general purpose ray tracing engine. *ACM Trans. Graph.*, 29(4).

Reeves, W. T., Salesin, D. H., and Cook, R. L. (1987). Rendering antialiased shadows with depth maps. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 283 – 291, New York, NY, USA. Association for Computing Machinery.

Ribeiro, M., Singh, S., and Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97 – 101, San Diego, California. Association for Computational Linguistics.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211 – 252.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2020). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336 – 359.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, volume abs/1409.1556.

Sixt, L., Granz, M., and Landgraf, T. (2020). When explanations lie: Why many modified BP attributions fail. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9046 – 9057. PMLR.

Williams, L. (1978). Casting curved shadows on curved surfaces. In *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '78, pages 270 – 274, New York, NY, USA. Association for Computing Machinery.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921 – 2929.

Supervised Learning for Makeup Style Transfer

Natalia Strawa
Warsaw University of Technology
Faculty of Electrical Engineering
ul. Koszykowa 75
00-662 Warsaw, Poland
natalia.strawa2.stud@pw.edu.pl

Grzegorz Sarwas
Warsaw University of Technology
Faculty of Electrical Engineering
ul. Koszykowa 75
00-662 Warsaw, Poland
grzegorz.sarwas@pw.edu.pl

Abstract

This paper addresses the problem of using deep learning for makeup style transfer. For solving this problem, we propose a new supervised method. Additionally, we present a technique for creating a synthetic dataset for makeup transfer used to train our model. The obtained results were compared with six popular methods for makeup transfer using three metrics. The tests were carried out on four available data sets.

The proposed method, in many respects, is competitive with the methods used in the literature. Thanks to images of faces with generated synthetic makeup, the proposed method learns to better transfer details, and the learning process is significantly accelerated.

Keywords

Makeup Transfer, Image Style Transfer, CycleGAN, GAN, Image Processing, Deep Learning

1 INTRODUCTION

The development of generative adversarial network (GAN) architectures triggers the proposal of many practical solutions to help us in life. We can observe the increased popularity of applications for virtualizing showrooms. Virtual trying on clothes, glasses, etc., became a popular e-commerce solution. Among these apps, we can find technology for makeup transfer, which refers to transferring a reference makeup to a face without makeup and maintaining the original appearance of the plain face and the makeup style of the reference face.

Makeup transfer entails many difficulties and challenges. This process can be divided into two main steps. The first one is responsible for extracting the makeup from the face with the makeup pattern we would like to transfer. Meanwhile, the second is connected with makeup applying.

In the first step, the most challenging task is properly separating the color of the foundation on the skin. The complete transfer of one person's skin color to another person's face is undesirable, especially in the case of people who are different in complexion. The same is true of other elements that may sometimes appear in the face photo. Examples of such details that should not be transferred are freckles, wrinkles, discoloration, and pieces that obscure the face in the image.

The second stage, applying the extracted makeup to the other face, is equally non-trivial. Usually, two people have different face shapes, so the makeup needs to be fitted to the face we want to transfer it. In addition, the photos may show differences in position and facial expressions. For example, only half of the makeup will be visible on a face positioned in profile. Still, the method should consider that the makeup is usually symmetrical and transfer the complete makeup to the target face. Similarly, the unusual facial expression should not disturb the algorithm.

To solve these problems, many makeup transfer techniques were developed [Ma21]. These methods can be categorized into two main groups: traditional makeup transfer [Ton07; Guo09; Sch11] and makeup transfer based on deep learning [Liu16; Joh16; Lia17; Li18; Che19].

The main contributions of this work are: (1) We propose an algorithm for generating synthetic makeup useful for creating synthetic dataset; (2) We propose a new, competitive supervised makeup transfer method. The conducted experiment confirmed that our solution is better at transferring makeup details, and the learning process was significantly accelerated.

2 RELATED WORK

Most of the traditional makeup transfer methods have high requirements regarding the reference image and

https://www.doi.org/10.24132/CSRN.3201.25

the target image [Ma21]. To do this process successfully, pose, and light conditions must be similar in both images, which is very hard to achieve. Among these solutions, we can find algorithms based on supervised [Ton07] and unsupervised learning [Guo09]. For the training process, supervised makeup transfer methods require a dataset with a target image and pair of reference images before makeup transfer and after. These algorithms are usually based on three main steps. First, calculate the color and lighting changes of the image before and after applying makeup. Then modify the skin texture and the color difference between the reference and target surfaces. Finally, transfer the makeup and adjust the makeup style to match the target face. Since the makeup transfer requires the transfer of many different elements, hence manual processing needs an extensive sequence of operations, such as, e.g., Bayesian matting [Chu01], graph cutting texture synthesis algorithm [Kwa03], Independent Component Correlation Algorithm (ICA) [Tsu03].

With some classic makeup methods, the makeup is transferred pixel-by-pixel, which is prone to the slightest facial shifts in the photo [Ton07]. Another possible approach is to use a 3D model of facial deformation, making it easier to target the right pixels [Bla99].

Traditional unsupervised makeup transfer methods use the distribution of the image in CIELAB color spaces. They then use the WLS algorithm or bilateral filtering method [Tom98] to perform edge smoothing and to smooth out the brightness layer to obtain the face structure layer [Ma21].

Whether it is a traditional makeup transfer method based on a supervised model or an unsupervised model, the pose and illumination requirements of the input image are relatively high [Ma21]. These disadvantages were eliminated by using deep learning technology. Deep neural network architectures allow achieving more realistic results. We can find solutions based on pixel iteration ([Liu16; Xu13; Gat16]) and model iteration methods based on GAN [Goo20] or Glow [Kin18]. Among the second and third types, we can distinguish the following methods [Li18; Cha18; Jia20; Che19].

The CycleGAN model [Zhu17] can be seen as a fusion of two GANs. This model can apply makeup without face makeup and remove makeup from the reference face but can only do general makeup transfer, and the quality of the generated image is not very high. Pix2pixHD [Wan18] uses the multi-scale cGAN structure [Mir14] for image transformation. The StarGAN model [Cho18] preserves more facial features, provides better image quality, and provides better transfer results compared to CycleGAN and cGAN by mapping across multiple domains using only a pair of generators and discriminators and effectively training images. In BeautyGAN [Li18] the discriminator distinguishes the generated image from the real samples of the domain. Based on the transfer of a set of domains, it uses pixels based on different areas of the face. Instance-level migration is achieved with the help of a level histogram loss. The preservation of facial integrity and elimination of artifacts is achieved by adding to the perceptual and cyclic consistency loss of the overall objective loss function. This model can transfer makeup images but not instance-level makeup images. Jiang et al. proposed the PSGAN [Jia20] to solve the problem of the difference between a reference face and a face without makeup. This model uses a bottleneck encoder in the generator structure in StarGAN to extract facial features and then uses the attention mechanism to adaptively modify the makeup matrix. The FSGAN [Nir19] assesses the occlusion area by combining facial segmentation. The SCGAN [Den21] breaks down the makeup transfer problem into two stages: extraction and allocation. The part-specific style encoder extracts features of each part and maps them into a disentangled style latent space, while the face identity encoder extracts the facial identity features of the target image. The makeup fusion is done by a decoder that combines the style code with the facial identity characteristics. [Ngu21a] proposed to build a unified template that can adjust the 3D head position, face shape, and facial expressions of the source and target images with the makeup transferring based on BeautyGAN method. They also proposed to use the UV texture map instead of the original image to replace the makeup.

The flow-based generation model was noticed after the publication of the Glow article [Kin18]. In the case of makeup transfer, this model does not require the training of two large networks of discriminators and generators, and the time of automatic synthesis of results is very short.

The Glow model introduces a reversible convolution based on RealNVP [Din17] and simplifies some of its components. An example of the Glow model for makeup transfer is BeautyGlow method [Che19]. It uses the latent space of the input image (the makeup reference image without the target image) and decomposes the latent space according to the facial features and makeup features, respectively. Finally, the reference image makeup features and the target image's facial features are added to get the target image's latent space with makeup. The Glow model is used to reverse and transform it into the target RGB image with makeup.

In this paper, we propose a supervised learning algorithm. To create this model, we prepared architecture for generating images with synthetic makeup used in the training process.

3 PROPOSED SOLUTION

3.1 Formulation

We consider two image domains, non-makeup image domain denoted as $N \subset \mathbb{R}^{H\times W\times 3}$ and makeup image domain denoted as $M \subset \mathbb{R}^{H\times W\times 3}$. Our goal is to learn the mapping function between these domains, denoted as $G : \{n_{src}, m_{ref}\} \rightarrow \{m^G_{src}, n^G_{ref}\}$. This means that given two input images: source image n_{src} and reference image m_{ref}, the network is expected to generate makeup transfer result m^G_{src} and makeup removal result n^G_{ref}. The first one receives makeup style from the reference image and preserves the facial features of the source image, while the second one has makeup removed from the reference image.

3.2 Dataset generation

The generator is trained in a supervised manner. Unfortunately, among the available datasets for makeup transfer, none contains pairs of before and after makeup images. We introduce a new dataset generated using an algorithm for synthetic makeup application to address this issue. The algorithm uses two available models for landmark detection: Dlib [Kaz14] and Mediapipe [Kar]. It can be used to apply eyeliner, lipstick, eye shadows, and blushes. The simplified diagram illustrating the creation of the specific makeup elements can be found in Fig. 1.

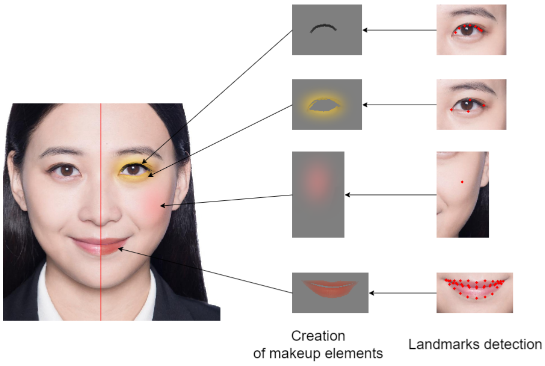


Figure 1: Dataset generation.

In the case of eyeliner, one can choose the position of lines on the eyelids (e.g., only on the upper eyelids, only on lower eyelids, or on both eyelids), their thickness, and transparency. When applying lipstick, the color, as well as its intensity and transparency, can be chosen. Eye shadows are determined by an ellipse with a center located approximately in the center of the eye and axes with lengths approximately equal to the width and height of the eye. The position of the center of the ellipse and the length of the axis can be modified by adding or subtracting from the initial values. This results in changing the shape of the shadows applied to the eyelids. The shape can also be controlled by changing the ellipse rotation angle. Besides, the color, transparency, and blur can also be modified. By shifting the center of the ellipse, it is possible to apply eyeshadows that appear only on the upper eyelid, only on the lower eyelid, or on both of them. Various unique shapes can be generated by modifying the ellipse's axis length and rotation angle. Blushes are created in a very similar way to eyeshadows. The ellipse center can be located at one of three points on the cheek. As with eyeshadows, it is possible to control the length of the ellipse axis, as well as the color and its parameters.

3.3 Dataset

To create the dataset, the parameters of the previously mentioned algorithm were randomized to make the makeups look natural but still have some diversity. For the generation of the synthetic dataset, non-makeup images from the Makeup Transfer [Li18] dataset were used. From this subset, 5000 pairs of images were sampled, and the same makeup style was applied to both images in every pair. One image from the pair represents the reference image, while the other represents the source image with the expected makeup style transfer. In total, 10000 images were created with synthetically generated makeup. The exemplary generated images are shown in Fig. 2.

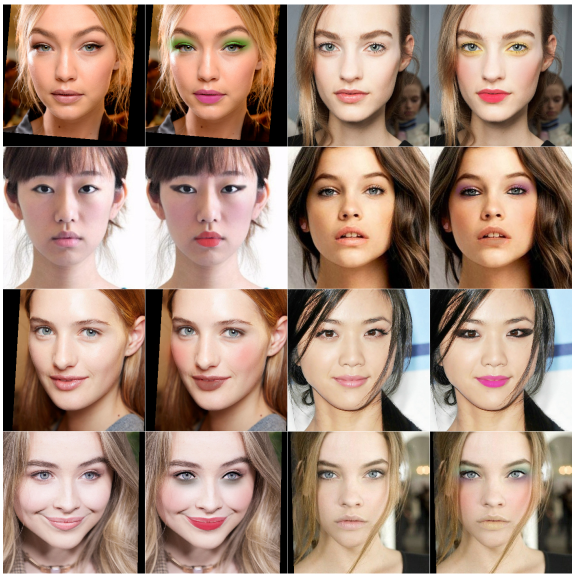


Figure 2: Examples of images from the newly generated dataset.

3.4 Framework

In our proposed method, we assume the training of four networks:

- Generator G,
- Discriminator D_N,
- Discriminator D_M,
- Discriminator D_S.

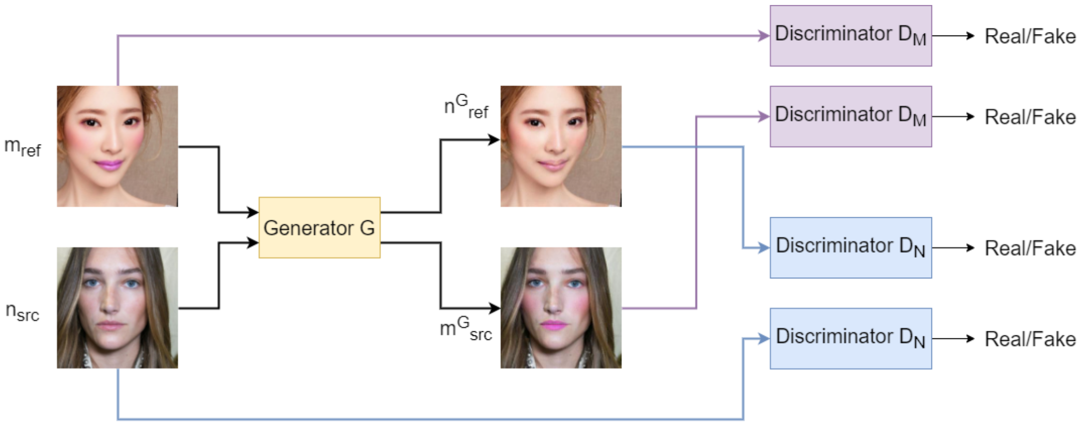


Figure 3: Framework. Domain-Level Makeup Transfer.

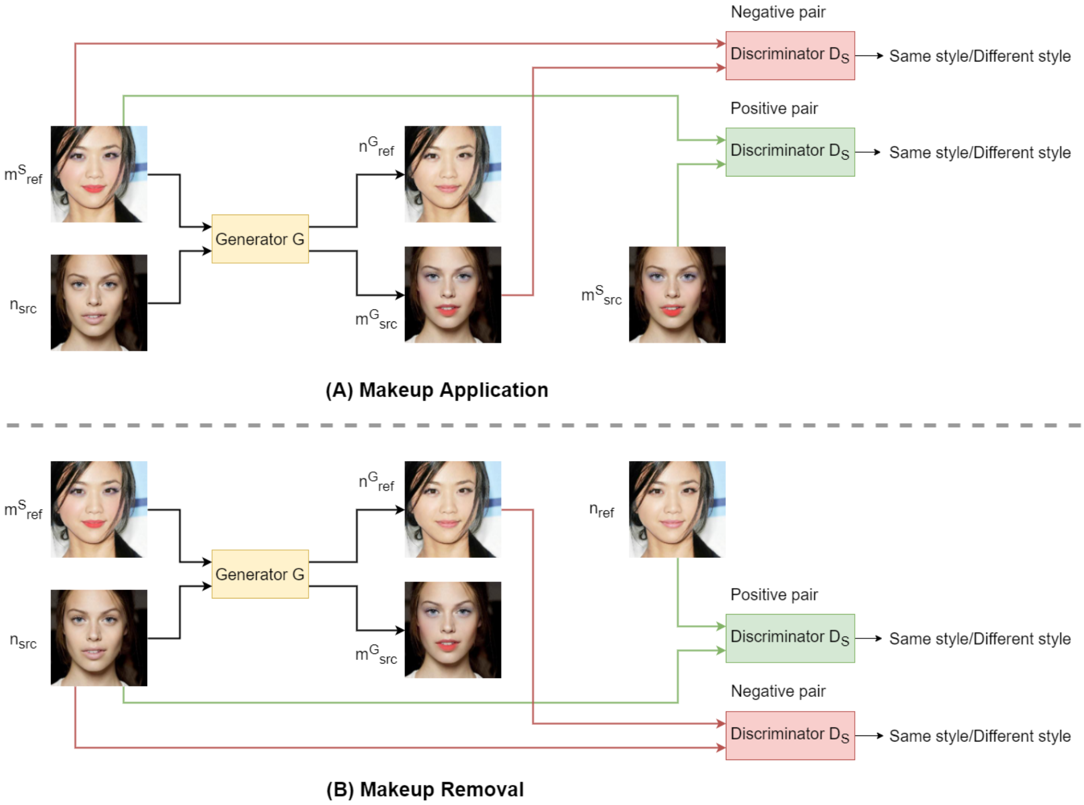


Figure 4: Framework. Instance-Level Makeup Transfer.

Generator takes as input a pair of images: source image n_{src} and reference image m_{ref} and generates two RGB masks R_1 for removing makeup and R_2 for applying makeup. In addition, weights W_1 and W_2 with values in the range $[0,1]$ are generated for each mask to determine their transparency in various areas. The parameters of the generator layers, such as filter size and stride, were derived from BeautyGAN. Additionally, as in PairedCycleGAN, we used dilated residual blocks. The discriminators D_N and D_M learn to determine the probability of whether the images belong to their corresponding domain or not, as shown in Fig. 3. Inspired by PairedCycleGAN [Cha18] we train an additional style discriminator D_S as shown in Fig. 4. It takes a pair of images as input and learns to determine whether they contain the same makeup style. The same discriminator is used to train the generator in the makeup transfer and removal process. Due to the lack of paired images with faces before and after makeup, a new dataset with synthetically generated makeup was created. In the case of makeup transfer, the positive pair were taken by the discriminator D_S consisting of a reference image m^S_{ref} and a source image with the same makeup m^S_{src}. The negative pair consists of the reference image m^S_{ref} and the makeup transfer result m^G_{src}. In the case of makeup removal, the positive pair includes the source image n_{src} and the reference image without makeup n_{ref}, and

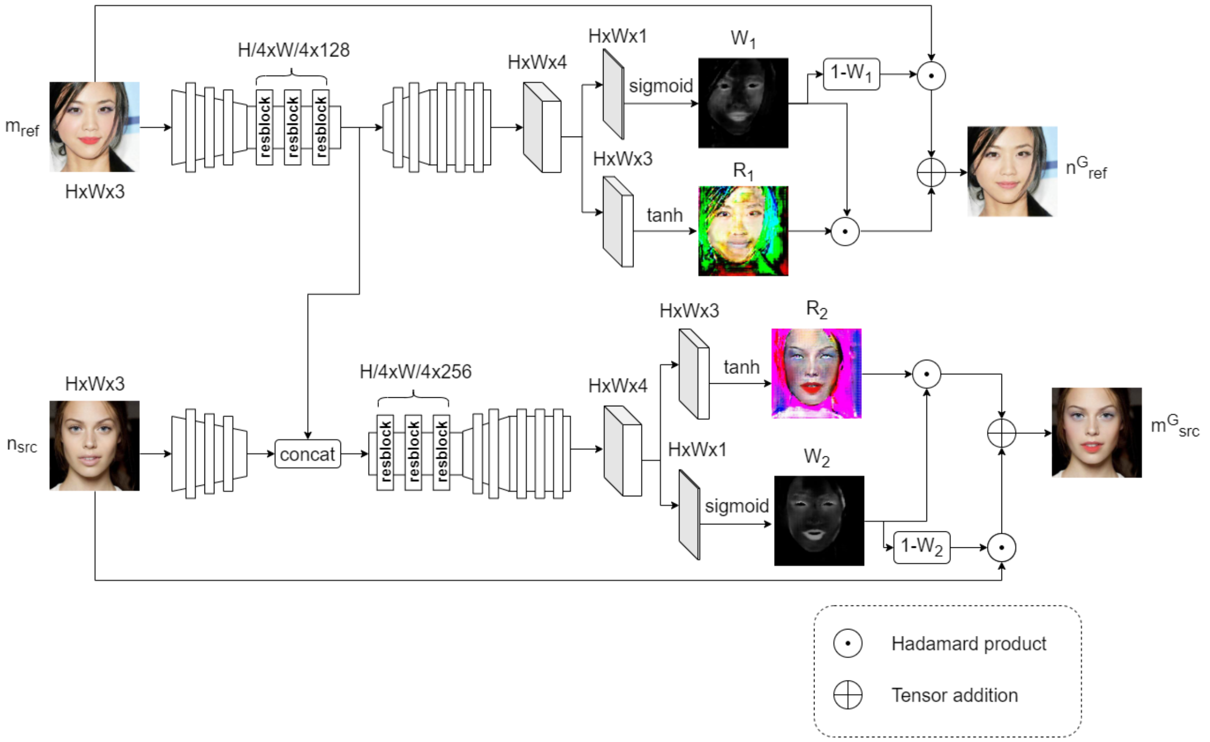


Figure 5: Architecture of generator.

the negative pair includes the source image n_{src} and the makeup removal result n^G_{ref}. The superscript ^{S} denotes the image with synthetically generated makeup, while the superscript ^{G} indicates the image generated by the generator.

Generator

The architecture of the proposed generator is shown in Fig. 5. In the beginning, two input images n_{src} and m_{ref} are passed into several downsampling convolutional layers of the two separate branches. The feature maps extracted from the makeup image are then forwarded to several residual blocks. Then the extracted feature maps are concatenated with the feature maps extracted from the non-makeup image and passed into several residual blocks. Finally, the feature maps pass through several upsampling convolutional layers, and an image with the same size as the input image and with four channels is returned. After applying the tanh activation function, the first three channels create an RGB mask for makeup transfer. The last channel represents the weights for the mask. A sigmoid activation function was used for the weights to be in the range $[0,1]$. The generation of the resulting image M^G_{src} can be written as follows:

$$m^G_{src} = n_{src} \odot (1 - W_2) + R_2 \odot W_2, \qquad (1)$$

where $\odot$ denotes the Hadamard product.

The same applies to the makeup image. The output feature maps from the residual blocks are passed to several upsampling convolutional layers, and then an RGB mask and weights for makeup removal are created in the same way as above. The resulting image is created as follows:

$$n^G_{ref} = m_{ref} \odot (1 - W_1) + R_1 \odot W_1, \qquad (2)$$

where $\odot$ denotes the Hadamard product.

3.5 Objective Function

Adversarial loss

The generator is guided by adversarial loss to provide more realistic results. We employed two discriminators, D_N and D_M, to distinguish the generated samples from real samples in domains N and M, respectively. Adversarial losses for discriminators D_N and D_M are defined as follows:

$$\begin{aligned} L_{D_N} =& \mathbb{E}_{n_{src}}[(D_N(n_{src}) - 1)^2] \\ &+ \mathbb{E}_{n_{src},m_{ref}}[D_N(n^G_{ref})^2], \end{aligned} \qquad (3)$$

$$\begin{aligned} L_{D_M} =& \mathbb{E}_{m_{ref}}[(D_M(m_{ref}) - 1)^2] \\ &+ \mathbb{E}_{n_{src},m_{ref}}[D_M(m^G_{src})^2], \end{aligned} \qquad (4)$$

where $\mathbb{E}$ is an expected value.

In order to train the generator to transfer a specific makeup, we introduced an additional style discriminator D_S to determine whether the same makeup is present in two images. Adversarial loss for this discriminator is defined as:

$$\begin{aligned} L_{D_S} =& \mathbb{E}_{n_{src}}[(D_S(n_{src}, n_{ref}) - 1)^2] \\ &+ \mathbb{E}_{n_{src}, m_{ref}}[D_S(n^G_{ref})^2] \\ &+ \mathbb{E}_{m_{ref}}[(D_S(m_{ref}, m^S_{src}) - 1)^2] \\ &+ \mathbb{E}_{n_{src}, m_{ref}}[D_S(m^G_{src})^2]. \end{aligned} \tag{5}$$

The adversarial loss for the generator is defined as follows:

$$\begin{aligned} L_{adv} =& \mathbb{E}_{n_{src}, m_{ref}}[(D_N(n^G_{ref}) - 1)^2] \\ &+ \mathbb{E}_{n_{src}, m_{ref}}[(D_M(m^G_{src}) - 1)^2] \\ &+ \mathbb{E}_{n_{src}, m_{ref}}[(D_S(n^G_{ref}) - 1)^2] \\ &+ \mathbb{E}_{n_{src}, m_{ref}}[(D_S(m^G_{src}) - 1)^2]. \end{aligned} \tag{6}$$

Mask loss

The mask loss L_{mask} is used to reduce the weights of the mask in areas of the image that should be unaffected. This includes the background, eyes, teeth, ears, hair, and neck. The mentioned loss can be expressed in the following way:

$$L_{mask} = ||W_{1background}||_1 + ||W_{2background}||_1, \tag{7}$$

where

$$W_{1background} = W_1 \odot L_{1background},$$

$$W_{2background} = W_2 \odot L_{2background},$$

and $\odot$ denotes the Hadamard product, $L_{1background}$ and $L_{2background}$ are the binary masks specifying the aforementioned background elements.

The full objective function of generator G contains two types of losses: adversarial loss and mask loss

$$L_G = L_{adv} + L_{mask}. \tag{8}$$

4 EXPERIMENTS

4.1 Training details

The generator convolutional layer parameters such as filter size and stride were derived from BeautyGAN [Li18]. However, the network structure has been slightly modified. At the same time, the normalization of weights was changed from Instance Norm to Batch Norm. Additionally, inspired by the PairedCycleGAN [Cha18], we also applied dilated residual blocks to the generator. The architecture of discriminators D_N and D_M was taken entirely from the BeautyGAN [Li18]. However, the discriminator D_S was modified to accept two images, which were then concatenated. The discriminator D_B was trained on a dataset containing real makeups, the style discriminator D_S was trained on the synthetically generated dataset, while the generator G was trained indirectly on both datasets.

For all experiments, the images have been resized to 256×256. We train the model for 25 epochs optimized by Adam [Kin15] with a learning rate of 0.0002 and a batch size of 4.

From initial experiments, it appears that our method learns faster than other methods. On the AMD Ryzen 1920X with NVidia RTX 2080Ti, it took about 4 hours to train. In comparison, the learning time of different methods is usually counted in tens of hours. E.g., implementation of BeautyGAN needs about one week with RTX 1080Ti. One branch of CPM consists of a generator from BeautyGAN, so this method requires at least as much time as BeautyGAN. The authors of BeautyGlow indicated that fine-tune Glow took 3 days. Unfortunately, the model of GPU used for training is unknown.

4.2 Comparisons to Baselines

We compared our method with six state-of-the-art methods for makeup transfer: BeautyGAN [Li18], BeautyGlow [Che19], CPM [Ngu21b], LADN [Gu19], PSGAN [Jia20] and SCGAN [Den21]. We used our implementation of the BeautyGlow model for testing, so the results may differ from those that the original model would have returned. Also, we used an available online implementation of the BiSeNet [Yu18] model to create segmentation masks for SCGAN so that the lower quality results may be the result of non-ideal masks.

4.2.1 Used Datasets

We performed tests on four available datasets: Makeup Transfer [Li18], Makeup Wild [Jia20], CPM-Synt-2 [Ngu21b], and a dataset shared by the authors of LADN [Gu19]. From the first three datasets, 2000 unique pairs consisting of a source and reference image were sampled. Later, 2000 examples were generated from the sampled pairs using each method. Based on the CPM-Synt-2 dataset, 1115 examples were generated. The CPM-Synt-2 dataset is the only one that contains pairs of before and after makeup images. First, pairs of pictures without makeup were sampled from the Makeup Transfer dataset to create the dataset. Then the same makeup was applied to both photos using BeautyGAN. The result was two images, where one was the reference image, and the other was the source image with the expected makeup.

4.2.2 Qualitative Comparison

The results of the visual comparison are shown in Fig. 6. BeautyGAN, PSGAN, and SCGAN significantly transfer skin color and shadows from the reference image to the source image. These methods

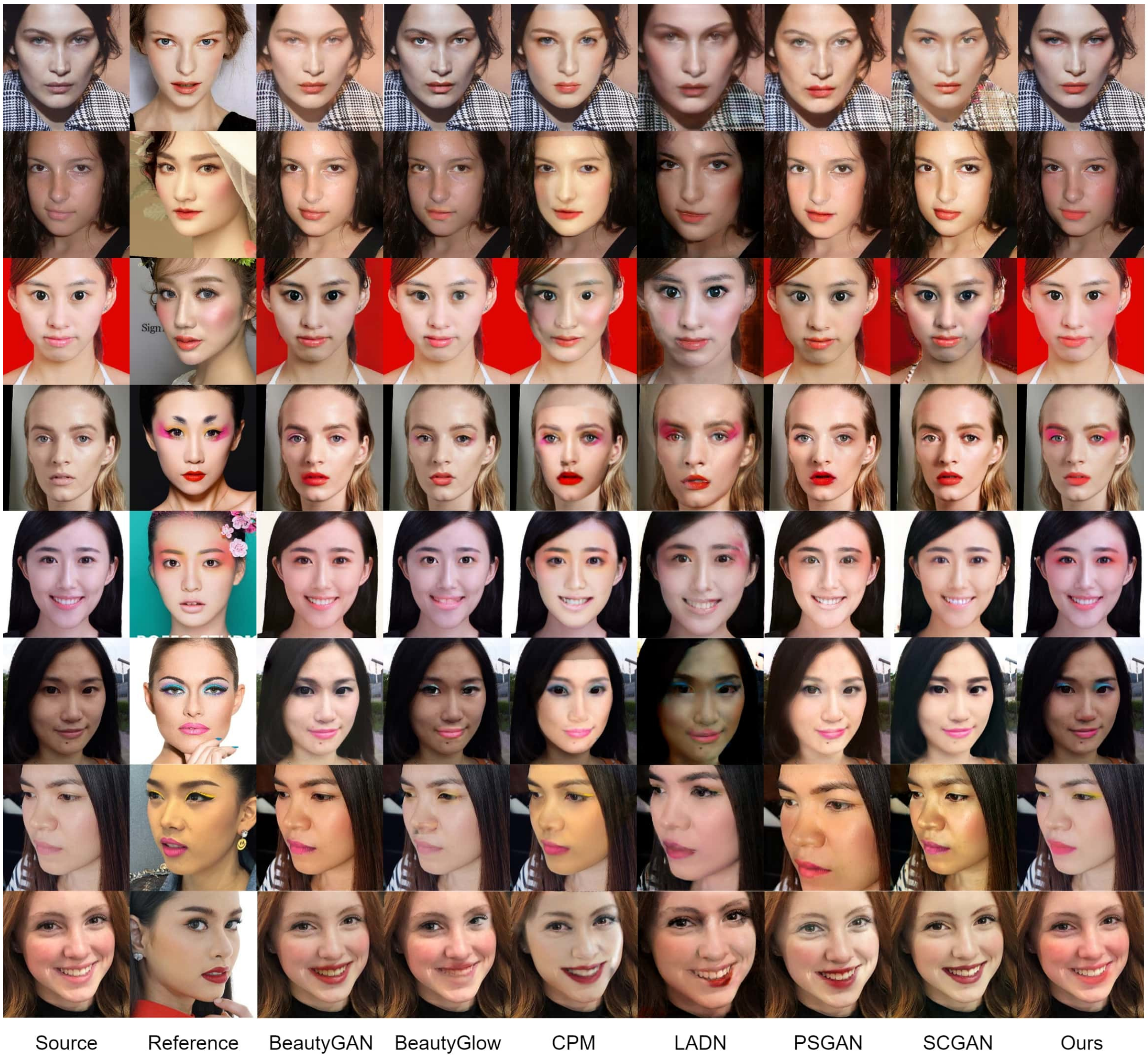


Figure 6: Comparison with state-of-the-art methods. First row: example from the CPM-Synt-2 [Ngu21b] dataset. Rows 2 and 3: examples from the Makeup Transfer [Li18] dataset. Rows 4, 5, and 6: examples from the dataset provided by the authors of the LADN [Gu19] method. Last two rows: examples from the Makeup Wild [Jia20] dataset.

transfer lip makeup very well, blush slightly less well, and eye makeup the least well. When they transfer eyeshadows, they lack details and color is often missing or blurred. BeautyGlow does not transfer makeup very well. Additionally, it transfers facial features from the reference face to the source face. CPM usually transfers almost the entire face from the reference image to the source image. Additionally, it performs poorly with makeup transfer when there are significant pose differences between the faces and often produces artifacts in images. On the other hand, it transfers makeup details and colors much better than the previously mentioned methods. LADN, like CPM, transfers makeup details quite well; however, it significantly lowers the quality of the images. The proposed method does not transfer skin color or shadows. In addition, it transfers color and makeup details well without affecting image quality. The disadvantage of this method is that it does not always transfer makeup well when there are differences in facial poses between images, as can be seen in the last row in Fig. 6.

4.2.3 Quantitative Comparison

Evaluation of the quality of generated images is a complex problem. It is even harder to evaluate whether the identical make-up has been transferred. Because of that, there is no one best metric to assess the quality of the model.

In our evaluation we used three metrics: FID (Fréchet inception distance) [Heu17], PSNR (Peak Signal-to-Noise Ratio) [Pon11], MS-SSIM (Multi-Scale Structural similarity) [Wan03]. FID correlates with the qual-

Method/Dataset	FID ↓				MS-SSIM ↑	PSNR ↑
	CPM-Synt-2	LADN	Makeup Transfer	Makeup Wild		
BeautyGAN	5.404	75.254	50.467	89.408	0.988	31.531
BeautyGlow	12.73	68.46	54.598	93.13	0.921	20.662
CPM	17.073	54.181	41.691	79.124	0.841	20.617
LADN	32.818	59.102	45.988	101.872	0.874	17.576
PSGAN	12.616	66.178	41.69	91.952	0.935	23.863
SCGAN	17.182	64.61	36.429	80.081	0.953	25.021
Ours	10.743	73.286	51.717	90.823	0.938	20.790
Unmodified images	-	-	-	-	0.944	20.944

Table 1: FID scores for the four datasets and MS-SSIM and PSNR scores for the CPM-Synt-2 dataset.

ity of the image generated but does not account for the transfer quality. The two other metrics evaluate how well the model transfers makeup from reference to the source image (based on labels from BeautyGAN as described in Section 4.2.1).

The FID score was calculated for all datasets. However, the other metrics were only calculated for the CPM-Synt-2 dataset because it was the only one labeled. Table 1 shows the values of the metrics.

The metrics results do not fully capture the quality of the makeup transfer for two reasons. First, some of the methods tested were trained on the datasets used for testing. The split between the training and testing datasets was unknown. Therefore, some results may be slightly biased. Second, the metric values do not fully correlate with the visual results, as seen in the CPM example. For several datasets, this method obtains the best FID metric scores. However, in Fig. 6, it can be seen that the results it produces often have visible artifacts. Since the scores obtained by our method are not significantly different from the scores obtained by other methods, and the visual results are similar, it can be concluded that this method is comparable to state-of-the-art methods.

5 CONCLUSION

In this paper, we propose a new supervised method for makeup transfer. Using a newly created dataset containing pairs of before and after makeup images, we were able to simplify the solution's architecture and accelerate the learning process. Methods using warping or histogram matching guide the generator towards results that are, by definition, not optimal. The proposed method does not suffer from such a problem, and the only thing that limits it is the quality and variety of makeup generated by the algorithm.

However, such an algorithm can be further developed and enriched with new types of makeup, which cannot be said about the other methods of this type. The new dataset helps better transfer makeup details such as eyeshadows and blush. One limitation of our approach is that it performs poorly with significant pose differences between faces. The presented visual and quantitative comparison shows that the proposed method is competitive with state-of-the-art methods for makeup transfer.

REFERENCES

[Bla99] V. Blanz and T. Vetter. "A Morphable Model for the Synthesis of 3D Faces". In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '99. USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194. ISBN: 0201485605. DOI: 10 . 1145 / 311535 . 311556. URL: https://doi.org/10.1145/311535.311556.

[Cha18] H. Chang et al. "PairedCycleGAN: Asymmetric Style Transfer for Applying and Removing Makeup". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 40–48. DOI: 10.1109/CVPR.2018.00012.

[Che19] H-J Chen et al. "BeautyGlow: On-Demand Makeup Transfer Framework With Reversible Generative Network". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10034–10042. DOI: 10.1109/CVPR.2019.01028.

[Cho18] Y. Choi et al. "StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8789–8797. DOI: 10.1109/CVPR.2018.00916.

[Chu01] Y-Y Chuang et al. "A Bayesian approach to digital matting". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 2. 2001, pp. II–II. DOI: 10.1109/CVPR.2001.990970.

[Den21] H. Deng et al. “Spatially-Invariant Style-Codes Controlled Makeup Transfer”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* June 2021, pp. 6549–6557.

[Din17] L. Dinh, J. S-D, and S. Bengio. “Density estimation using Real NVP”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017.

[Gat16] L. A. Gatys, A. S. Ecker, and M. Bethge. “Image Style Transfer Using Convolutional Neural Networks”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 2414–2423. DOI: 10.1109/CVPR.2016.265.

[Goo20] I. Goodfellow et al. “Generative Adversarial Networks”. In: *Commun. ACM* 63.11 (Oct. 2020), pp. 139–144. ISSN: 0001-0782. DOI: 10.1145/3422622. URL: https://doi.org/10.1145/3422622.

[Gu19] Q. Gu et al. “LADN: Local Adversarial Disentangling Network for Facial Makeup and De-Makeup”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV).* 2019, pp. 10480–10489. DOI: 10.1109/ICCV.2019.01058.

[Guo09] D. Guo and T. Sim. “Digital face makeup by example”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition.* 2009, pp. 73–79. DOI: 10.1109/CVPR.2009.5206833.

[Heu17] M. Heusel et al. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

[Jia20] W. Jiang et al. “PSGAN: Pose and Expression Robust Spatial-Aware GAN for Customizable Makeup Transfer”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2020, pp. 5193–5201. DOI: 10.1109/CVPR42600.2020.00524.

[Joh16] J. Johnson, A. Alahi, and L. Fei-Fei. “Perceptual Losses for Real-Time Style Transfer and Super-Resolution”. In: *Computer Vision – ECCV 2016*. Ed. by B. Leibe et al. Cham: Springer International Publishing, 2016, pp. 694–711. ISBN: 978-3-319-46475-6.

[Kar] Y Kartynnik et al. “Real-time facial surface geometry from monocular video on mobile GPUs. arXiv 2019”. In: *arXiv preprint arXiv:1907.06724* ().

[Kaz14] Vahid Kazemi and Josephine Sullivan. “One millisecond face alignment with an ensemble of regression trees”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1867–1874. DOI: 10.1109/CVPR.2014.241.

[Kin15] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6980.

[Kin18] D. P. Kingma and P. Dhariwal. “Glow: Generative Flow with Invertible 1x1 Convolutions”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[Kwa03] V. Kwatra et al. “Graphcut Textures: Image and Video Synthesis Using Graph Cuts”. In: *ACM Trans. Graph.* 22.3 (July 2003), pp. 277–286. ISSN: 0730-0301. DOI: 10.1145/882262.882264. URL: https://doi.org/10.1145/882262.882264.

[Li18] T. Li et al. “BeautyGAN: Instance-Level Facial Makeup Transfer with Deep Generative Adversarial Network”. In: *Proceedings of the 26th ACM International Conference on Multimedia*. MM ’18. Seoul, Republic of Korea: Association for Computing Machinery, 2018, pp. 645–653. ISBN: 9781450356657. DOI: 10.1145/3240508.3240618. URL: https://doi.org/10.1145/3240508.3240618.

[Lia17] J. Liao et al. “Visual Attribute Transfer through Deep Image Analogy”. In: *ACM Trans. Graph.* 36.4 (July 2017). ISSN: 0730-0301. DOI: 10.1145/3072959.

3073683. URL: https://doi.org/10.1145/3072959.3073683.

[Liu16] S. Liu et al. "Makeup like a Superstar: Deep Localized Makeup Transfer Network". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI'16. New York, New York, USA: AAAI Press, 2016, pp. 2568–2575. ISBN: 9781577357704.

[Ma21] X. Ma et al. "Deep learning method for makeup style transfer: A survey". In: *Cognitive Robotics* 1 (2021), pp. 182–187. ISSN: 2667-2413. DOI: https://doi.org/10.1016/j.cogr.2021.09.001. URL: https://www.sciencedirect.com/science/article/pii/S266724132100015X.

[Mir14] M. Mirza and S. Osindero. "Conditional Generative Adversarial Nets". In: *CoRR* abs/1411.1784 (2014). arXiv: 1411.1784. URL: http://arxiv.org/abs/1411.1784.

[Ngu21a] T. Nguyen, A. T. Tran, and M. Hoai. "Lipstick ain't enough: Beyond Color Matching for In-the-Wild Makeup Transfer". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 13300–13309. DOI: 10.1109/CVPR46437.2021.01310.

[Ngu21b] T. Nguyen, A. T. Tran, and M. Hoai. "Lipstick ain't enough: Beyond Color Matching for In-the-Wild Makeup Transfer". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 13300–13309. DOI: 10.1109/CVPR46437.2021.01310.

[Nir19] Y. Nirkin, Y. Keller, and T. Hassner. "FSGAN: Subject Agnostic Face Swapping and Reenactment". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 7183–7192. DOI: 10.1109/ICCV.2019.00728.

[Pon11] Nikolay Ponomarenko et al. "Modified image visual quality metrics for contrast change and mean shift accounting". In: *2011 11th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*. 2011, pp. 305–311.

[Sch11] K. Scherbaum et al. "Computer-suggested Facial Makeup". In: *Comp. Graph. Forum (Proc. Eurographics 2011)* 30.2 (2011).

[Tom98] 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.

[Ton07] W-S Tong et al. "Example-Based Cosmetic Transfer". In: *15th Pacific Conference on Computer Graphics and Applications (PG'07)*. 2007, pp. 211–218. DOI: 10.1109/PG.2007.31.

[Tsu03] N. Tsumura et al. "Image-Based Skin Color and Texture Analysis/Synthesis by Extracting Hemoglobin and Melanin Information in the Skin". In: *ACM SIGGRAPH 2003 Papers*. SIGGRAPH '03. San Diego, California: Association for Computing Machinery, 2003, pp. 770–779. ISBN: 1581137095. DOI: 10.1145/1201775.882344. URL: https://doi.org/10.1145/1201775.882344.

[Wan03] Z. Wang, E.P. Simoncelli, and A.C. Bovik. "Multiscale structural similarity for image quality assessment". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*. Vol. 2. 2003, 1398–1402 Vol.2. DOI: 10.1109/ACSSC.2003.1292216.

[Wan18] T-C Wang et al. "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8798–8807. DOI: 10.1109/CVPR.2018.00917.

[Xu13] L. Xu, Y. Du, and Y. Zhang. "An automatic framework for example-based virtual makeup". In: *2013 IEEE International Conference on Image Processing*. 2013, pp. 3206–3210. DOI: 10.1109/ICIP.2013.6738660.

[Yu18] Changqian Yu et al. "Bisenet: Bilateral segmentation network for real-time semantic segmentation". In: *European Conference on Computer Vision*. Springer. 2018, pp. 334–349.

[Zhu17] J-Y Zhu et al. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 2242–2251. DOI: 10.1109/ICCV.2017.244.

Mesh compression method with on-the-fly decompression during rasterization and streaming support

Anton Nikolaev
Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Moscow, 119991 , Russia
anton.nikolaev@graphics.cs.msu.ru

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

Vladimir Frolov
Keldysh Institute of Applied Mathematics RAS
Miusskaya sq., 4
Moscow, 125047, Russia
Lomonosov Moscow State University
vfrolov@graphics.cs.msu.ru

ABSTRACT

In this article we propose a method of mesh compression and streaming, that can be used for real-time rendering applications. While most of other existing compression methods require decompression on CPU before rendering and streaming methods use only non-compressed models, our approach allows to reduce memory consumption by applying decompression on GPU during rendering and streaming of only needed geometry data at the same time. Proposed approach requires pre-processing step, on which coarse 3D model with quad faces is build and resampling is done. Afterwards, each face of model is compressed and can be later rendered with tessellation shaders (decompression is done during rasterization). Also, we propose a way of adding streaming support to our compression method to further reduce memory consumption. Finally, we made a comparison with state of the art approach levels of detail (LODs) approach and found that proposed approach has much lower memory consumption without negative effects to rasterization performance and quality.

Keywords

Meshes, compression, streaming, parallel decompression.

1 INTRODUCTION

3D model compression is often necessary. With growth of mesh quality the amount of memory required to store these meshes increases too. Also, a GPU with better computation performance can be required to support real time rendering when using higher quality meshes. For now several solutions of this problem exist.

2 EXISTING SOLUTIONS

The first solution of the described problem is mesh compression. Almost all compression approaches use vertex data quantization (lossy compression technique, which compresses a range of values to a single quantum value) [Cho02] and try to reduce as much as possible the size of connectivity data or don't store it at all. First steps in this direction were done by applying triangle strips and triangle fans approaches, which allow storing only one index of new vertex per triangle instead of storing all three indices. Also some generalizations can be used [Dee95]. Triangle traversal approaches make another large group of mesh compression methods. Such approaches allow replacing indices with traversal history data. Several symbol codes can be used, specifying placement of new triangle vertices on the border of the compressed area [Gum99] [Ros99]. Sometimes additional data is also required to be stored with the history. Several such approaches also allow combining both quad and triangle faces in the same mesh [Lee02]. Another group of mesh compression techniques is formed by valence-driven methods. Vertex valences are written during traversal instead of traversal history, but sometimes additional data is also needed [All01]. Progressive compression is also possible and it is based on applying of some simplification operation (vertex or edge removal, etc) to the source mesh until the coarse mesh will be generated. During decompression, opposite operations are applied in the reversed order to the coarse mesh to restore the original one [Hop96] [Coh99]. However, all described approaches share one common problem. Parallel decompression on GPU is impossible due to data depen-

https://www.doi.org/10.24132/CSRN.3201.26

dency between iterations of the decompression process. So the decompressed model would be stored in VRAM and GPU memory usage won't change. Mesh compression method, allowing random access to mesh parts was also proposed [Cho08]. It is achieved by splitting a source mesh into parts (meshlets). By decreasing meshlet size parallel decompression can be applied [Zha12]. The authors used mesh segmentation and afterwards each part was compressed independently from all others. Decompression was done once on GPU with Cuda. Methods with parallel GPU decompression were also proposed [Jak17] [Mah21], however they use Cuda or compute shaders to decompress the mesh once before usage. It increases decompression speed comparing to CPU implementation, however GPU memory usage is still the same. Several compression methods were proposed for GPU decoding during rendering. However, they mostly apply only quantization for vertex attributes and possibly do some initial mesh simplification [Cho97] [Cal02] [Hao01]. Finally, there exist compression method suitable for multi-resolution meshes that uses hierarchy data to improve compression rate, but parallel decompression on GPU also is not supported [Käl09].

Levels of detail can reduce memory consumption allowing only the needed part of the model to be loaded into GPU memory [Lue03]. For complex scenes including a huge number of separate objects the HLODs approach [Eri01] also exist. It proposes to build LOD not only for separate objects, but also for groups allowing to increase LOD quality without negative memory usage impact. LODs generation in run-time was also proposed to be used for dynamic scenes.

Mesh streaming (loading of only necessary mesh data instead of loading the whole mesh) can also be used, but most existing solutions propose only mesh transfer over the network [Kim04]. Partial load of mesh data from external storage to RAM for more effective processing (including compression) is also described [Ise05], but in all cases only the decompressed mesh is used during rendering [Dou19].

In conclusion, the problem of mesh compression method not requiring decompression before rendering is still open. The aim of our work was to develop such method. In addition we decided to use some version of streaming or LODs with our proposed compression algorithm to further decrease memory consumption.

3 PROPOSED APPROACH

The proposed approach is based on generating a coarse mesh with quad faces (later named patches). Afterwards, the resampling (generation of an almost regular mesh using the original one and the coarse one) is done to achieve the required quality (chosen by the user). Finally compression is applied to the obtained data. With certain limitations on the resampling process the resulting data can be rendered with tessellation shaders, which are available and hardware accelerated on most of modern GPUs.

Coarse model generation and resampling

Coarse meshes are generated using the open-source tool Instant Meshes [Jak15]. This step requires certain user involvement to control several parameters, which cannot be computed automatically.

Afterwards the source mesh is resampled using GPU accelerated ray-tracing. For this process we use *rayQuery* from tessellation shaders. The acceleration structure is built from the source model. For each point in a uniformly tessellated patch a ray is traced along the interpolated normal to find the intersection point on the source mesh. The position of intersection becomes the position of this point. Finally this result is saved to buffer and moved to the GPU when all user-controlled setup is finished.

Such an approach achieves performance acceptable for interactive parameter tuning during the resampling process. The user can control mesh quality (controlled by the tessellation factor) and the maximum distance between source point and ray intersection point and see the resulting mesh in real-time with the ability to move the camera. Actually this resampling method can be replaced by some more complex approach. The only requirement is the same vertex and polygon placement pattern, that is used by the tessellation. Vertex positions can vary and are not required to be uniformly placed over the patch.

Compression

To compress all vertex data quantization is used. Bit count is selected separately for patch corners, inner vertices and patch borders depending on required precision (also specified by the user). For border vertices and inner vertices not absolute values but deltas (differences of values from some predicted ones) were quantized. Also, it should be noted, that quantization bit counts are selected for the whole mesh, not per patch or per border.

Deltas for borders are computed from points, appearing during subdivision of edges into equal parts (number of parts depends on number of border vertices). Transformation to new coordinate system, made by edge direction and two orthogonal axis, is also done for edge deltas. For inner vertices deltas are computed from positions appearing in the tessellation with uniform vertex placement. Coordinate system transformation is applied to inner vertex deltas too and uses interpolated normal and patch edge direction to form orthonormal basis. (See fig. 1)

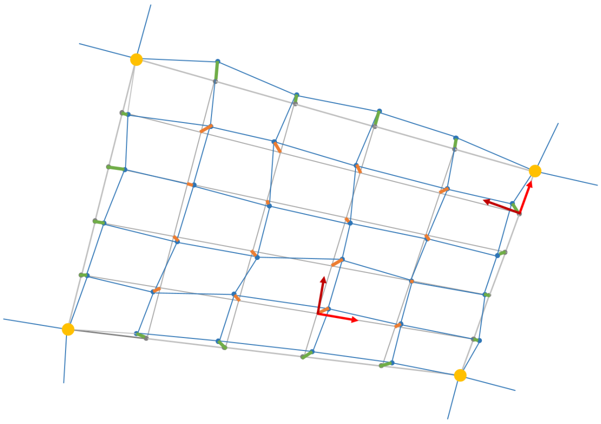


Figure 1: Visualization of saved data on one patch (projected along normal for simplicity). Corner vertices are shown yellow color, resampled grid is shown with blue lines and points. Deltas for border vertices are shown by green lines, inner vertex deltas are shown by orange lines. Grey grid shows a uniform subdivision of the patch. Axes for one border vertex and one inner vertex are shown with red arrows.

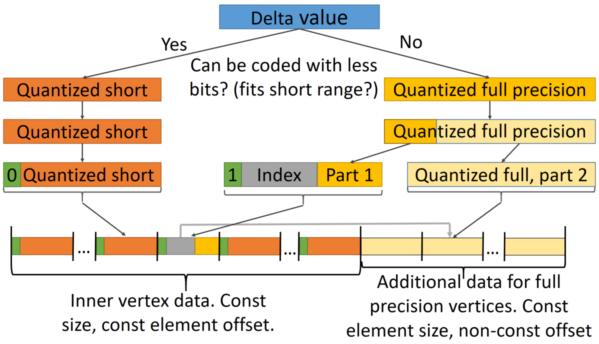

Delta value
Yes
No
Can be coded with less bits? (fits short range?)
Quantized short
Quantized full precision
Quantized short
Quantized full precision
0 Quantized short
1 Index
Part 1
Quantized full, part 2
...
...
...
Inner vertex data. Const size, const element offset.
Additional data for full precision vertices. Const element size, non-const offset

Figure 2: Visualization of proposed coding scheme.

For many patches inner vertex deltas are small especially along axes close to patch edges. To save some memory amount on heading zeroes appearing during small deltas quantization with constant bit count we propose a coding scheme with less number of bits for such cases. Supposing that m bits will be used for quantization following data representation is used. $(m+1)$ bits are stored for each inner vertex. One leading bit specifies value format. The following m bits contain either quantized delta with low bit count (if source delta value was small enough) or index of additional data and part of quantized value (if all indices in all patches fit less than m bits). For each patch additional data is placed after the main data. It allows saving some memory on heading zeroes for small deltas and still supports random access to data of each coordinate, so no decoding of previous vertices is required. Bit counts for both quantization formats are common for all patches of the mesh. The proposed coding scheme is shown on fig. 2. To obtain the final value one memory read from the precomputed address and possibly one more read operation from the address computed using the previously read value (required only for vertices quantized with full bit count) are required. The number of bits for short deltas is computed by iterating through all possible values (the optimal range is also found for each case) to use as small amount of memory as it is possible. As indices of additional data don't use much memory, while a lot of quantized deltas have heading zeroes, the proposed approach allows to save more than 10% of memory used by inner vertices data (see fig. 3).

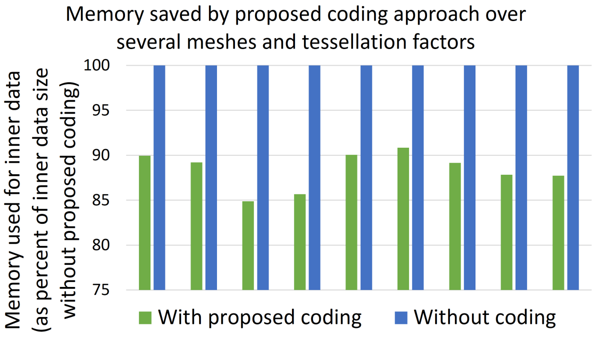


Figure 3: Memory saved by proposed coding scheme.

Per patch data is also needed and it stores corner vertex and border indices (because this data is shared between several patches) and offset of inner data in buffer, containing inner data (this value can't be simply computed during decompression, because each patch can have own inner data size).

Decompression

Decompression is done on the GPU during rasterization and uses tessellation shaders. Data is passed in 4 storage buffers (corner vertex data, border data, inner data and patch descriptions), each of them is presented as unsigned 32-bit integer array. The tessellation control shader decodes some common per patch data like corner vertex indices and positions and sets the tessellation factor. The tessellation evaluation shader restores either one inner vertex position or border vertex position. The total cost of one vertex decode includes a read of per-patch parameters. Each packed value with not-power-of-two bit count requires one or two memory reads (each value can either fully be placed in a 32-bit unsigned integer or be split between two subsequent integers and require two reads). So decoding cost also includes 1-2 memory reads for border vertex or 1-4 reads for inner vertex (one or two packed values) per each of 3 position components. Finally several MAD and bit-wise operations are required per component.

Streaming

As stated before, we store only the compressed model in GPU memory, but with streaming memory consumption can be improved even more, by storing only

https://www.doi.org/10.24132/CSRN.3201.26

needed parts of mesh. However, due to the specifics of the proposed compression approach (inner and border data is stored separately, quantized inner vertex deltas have their own packed format) streaming required some changes.

Each patch can use its own level of detail independent from other patches. We just introduce a limitation for total amount of memory used by all mesh data. Actually it is a configurable parameter and can be set depending on the application. The only data that should always be placed into GPU memory is patch descriptions, including border addresses, level of inner tessellation, and indices of corner vertices. Also corner vertices should always be placed in GPU memory. Actually part of this data is used to support streaming, while other part can be thought about as a minimal presented LOD of the mesh. During rendering the tessellation control shader can write requests for inner vertex data load into a request buffer if edge size gets greater than a defined value (it is also application controlled). In the next frame the CPU reads back these values and produces a buffer of processing requests for the compute shader. Also border tessellation factors are computed during this process. The compute shader decompresses inner vertex data and border data, does the interpolation of positions depending on the required tessellation factors and packs all this data with the described compression algorithm. However, due to data packing the inner data size can't be predicted before interpolation. So we use additional temporary buffer for processed data, where each patch has enough space to be placed not depending on interpolation results. Sizes of obtained data are also written to this buffer. Later we read this data on the CPU side and select where to copy these results. Such an approach doesn't noticeably increase memory usage as the size of all these buffers is constant not depending on mesh size and is also configurable. During streaming patch description data is also updated. Addresses of borders are selected on the CPU before compute shader dispatch and are written by this shader. For the inner data addresses are generated based on sizes from temporary buffer and written to patch descriptions during copying. Tessellation factors in patch descriptions (for inner factor) and in border data are also updated.

Memory management and defragmentation

The main problem of the proposed approach comes from data fragmentation. During camera movement some patches of the mesh can require better quality, so their tessellation factors should be increased and new data should be loaded. In most of cases new inner or border data can't be loaded in place of previous data, as its size increases, so a new place for it is found in buffer. And even after space used by previous patch data is marked as free it still can't be fully utilised, because with the proposed coding scheme its hardly possible to find several blocks with exactly matching size. So after several such iterations more and more memory is wasted, until there will be no space for required data in the buffer.

Use of standard solutions such as paging is hardly possible. Inner vertex position data (and border data that takes even less size) take too small an amount of memory to use this value as page size: such a choice will require a lot of additional memory to support a page table with huge number of pages. At the same time, large pages will introduce internal fragmentation. So we developed own method of memory management and defragmentation. It is supported on the CPU side and tries to fit as much data as possible into fixed size buffers. If there is not enough space to load required data the defragmentation process happens.

For defragmentation a temporary fixed-size buffer is used. The sequence of free and used blocks starting and ending with used blocks is searched with the requirement to maximize the size of free blocks inside. Another restriction is total size of used blocks in sequence, as all these blocks should fit the temporary buffer. When the sequence is found, all used blocks are shifted to the end of last used block placed before the sequence, producing one free block made from several ones met in sequence. If such sequence can't be found, the maximum sequence of used blocks placed between two free blocks and fitting into the fixed-size buffer is searched for and merged with another used sequence. Both operations are done by first copying all used blocks into the temporary buffer with the compute shader to avoid numerous buffer copy operations. Afterwards the filled contents of the temporary buffer are copied back by one operation. Only one such operation is applied each frame to reduce the performance impact. Such an approach has a fixed memory (and actually can have even zero memory cost as some currently unused buffer can be used as temporary too) and performance cost, but also controls the maximum amount of memory wasted due to fragmentation. It can be seen that this algorithm can't do anything only in one case: when we have a sequence of free and used blocks, where none of the free blocks is suitable for new data allocation, while each sequence of used blocks is larger than temporary buffer. The worst case appears when the buffer is filled by sequences of used blocks with a bit greater size than temporary buffer has, while free blocks between them have size a bit less than required for per patch inner vertex data. So even having temporary buffer with size of tens of patches would result in several percent of memory wasted due to fragmentation in worst case (at the same time full a mesh can include hundreds or thousands of patches, so size of temporary buffer is quite small). Finally, the worst case wouldn't generally be present for a long time since on camera movement some patches can request lower tessellation factor and will be replaced, leaving some free space.

If defragmentation fails, the memory management system tries to downgrade some blocks, that are not visible or don't require high tessellation factor any more. Afterwards defragmentation is applied again, if the block for new data still cannot be allocated.

Limitations

In its current version the proposed approach applies only to vertex positions. In general, it can be extended to compress also normals and texture coordinates. The only problem is the resampling process on texture coordinate seams.

Also, the described approach will show low compression efficiency on meshes containing a lot of separate parts each with a few triangles. This problem comes from the generation of the coarse model. Each such part will generate a patch, that will require to store a patch description for each such part.

Finally some compression steps involve user interaction, as we can't choose some parameters automatically. During coarse model generation we need to get as low patch count as possible, but still save certain mesh properties (new gaps shouldn't appear, patches should be placed to allow resampling of any visually important part of the source mesh, etc). A minimal amount of patches is desirable, because each patch always stores its description and corner vertices in GPU memory and this size is not dependant on the required tessellation factors of patches. Also, triangle and pentagon patch faces produced by instant meshes should be removed if possible. Each triangle face is presented as a quad with one fictive vertex. Pentagon faces are converted into 5 triangles and same procedure is applied afterwards. An alternative solution is to split such faces into 3 or 5 quads, but it produces T-junctions. And subdivision of all faces to remove these T-junctions is not acceptable, as it increases number of patches in the coarse model by 4 times, that will be even more inefficient.

4 EXPERIMENTAL RESULTS AND COMPARISON

To compare memory usage of proposed approach with LODs we used different tessellation factors the same across all model (actually its worst case for streaming) and compared memory consumption with an LOD, having same number of faces as the tessellated surface. From fig. 4 and fig. 5 it can be seen that with minimal level of detail our method uses a bit more memory comparing to an LOD, as it uses more memory for patch descriptions (4 indices, inner vertex data and border data addresses, tessellation factor) than LOD per face (just 4 indices). However, when the quality increases, our method shows much better memory usage, as we use delta coding with quantization, packing for internal data and don't need to store indices for border and inner vertices of each patch.

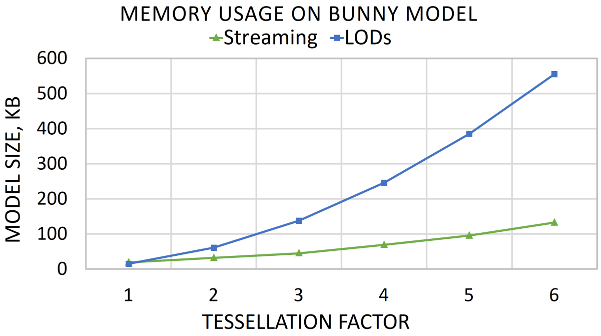


Figure 4: Comparison between the proposed compression method and an LOD with same number of polygons as the tessellated surface has (Stanford Bunny model).

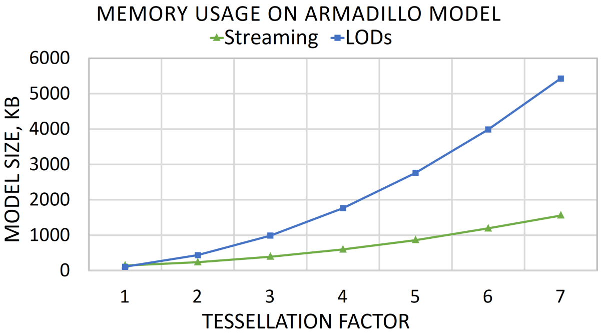


Figure 5: Comparison between the proposed compression method and an LOD with same number of polygons as the tessellated surface has (Armadillo model).

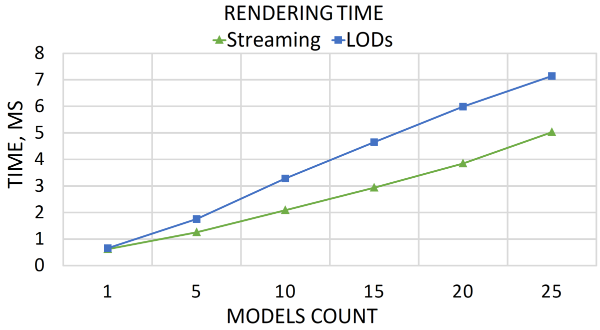


Figure 6: Comparison of rendering time with LODs and proposed approach.

Also, performance of rendering with both LODs and proposed approach was measured. We found that proposed approach can be used for real-time applications. Figure 6 shows that our method can work even faster

Edge length	Distance	LODs (kb)	Streaming (kb)
0.605	2.891	19.230469	18.90625
0.075	2.401	37.828125	18.90625
0.401	0.736	19.230469	19.105469
0.054	2.891	74.941406	19.929688
0.048	2.401	74.941406	20.839844
0.177	0.736	148.359375	21.539062
0.041	2.891	148.359375	23.367188
0.136	0.736	293.789062	24.050781
0.034	2.401	148.359375	27.671875
0.095	0.736	583.6875	29.210938
0.082	0.736	583.6875	32.234375
0.027	2.891	293.789062	35.550781
0.027	2.401	293.789062	35.5625
0.02	2.891	583.6875	52.527344
0.027	0.736	1154.882812	61.8125
0.0	0.736	1154.882812	70.742188
0.014	2.401	583.6875	81.511719
0.014	2.891	1154.882812	83.984375
0.0	2.401	1154.882812	120.609375

Table 1: Memory usage comparison with streaming and LODs depending on distance to mesh and required maximum edge length (in screen space coordinates). Zero edge length means to load all patches and LOD in maximum quality.

than LODs. In both cases the performance was actually limited by the primitive clipping and culling stage, not by memory access or SM (Streaming Multiprocessor) usage, showing that proposed approach has low computation cost and acceptable number of memory reads. Finally, some other work can be done on the GPU at the same time (for example pixel shaders or async compute queue tasks), as SM units are loaded by less than 33% according to Nvidia NSight.

Also, we compared our proposed approach with LODs on different camera positions with the Stanford Bunny model. For LODs we were supporting median will edge size not greater than a specified value. The same value was used as maximum edge length in each patch for streaming with compression. Results are shown in table 1. It can be seen that LODs use same amount of memory only when the minimal tessellation factor on all patches and minimal LOD are selected (lines 1 and 3 in table). In other cases LODs use much more memory.

Figures 7, 8, 9, 10 and 11 show visual comparison of LODs and the proposed approach with different camera positions and desired edge sizes. For all pictures in this section LODs are shown with a white mesh, while the proposed approach shows colored mesh (patches with different tessellation factors with different colors). Also, all comparisons are done with wireframe rendering mode to show polygon sizes. Regions of interest are shown with red rectangles. It can be seen that for proposed approach mesh is built in a more optimal manner. For mesh parts close to camera faces have similar size for both methods, while far parts require less polygons with the proposed approach, as they don't require such quality. Finally, fig. 12 shows sample render of Armadillo 3D model by proposed streaming and compression method.

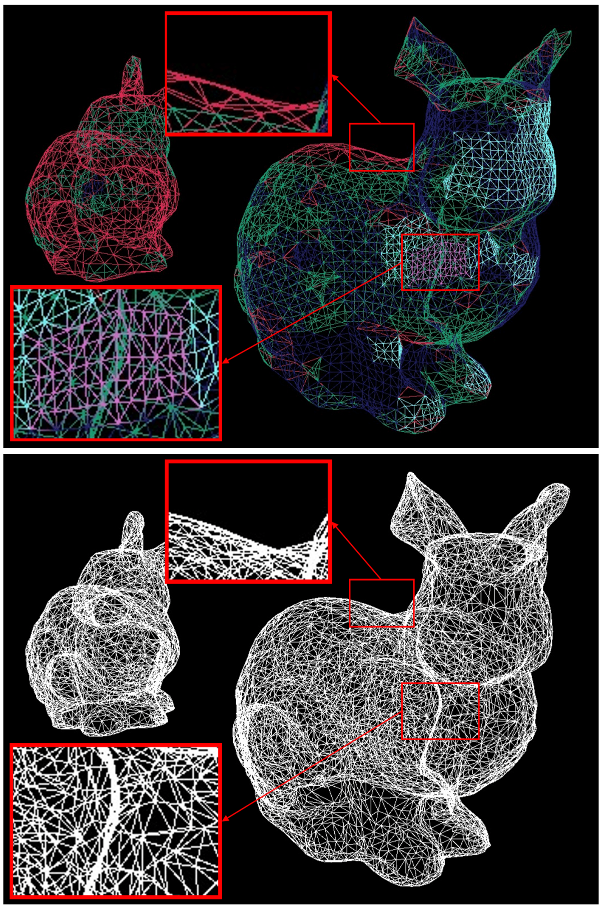


Figure 7: Visual comparison of rendering with LODs and the proposed approach.

5 CONCLUSION

In this paper we proposed a novel approach, that compresses models up to three times according to experimental results. Also, we describe a streaming and memory management systems, which can be used to further improve the memory usage. Finally, we made some comparisons, showing that our approach is fast enough and has effective VRAM usage at the same time and can be used in real-time applications. However, several limitations still exist, mostly caused by coarse mesh generation process difficulties. Also, some improvements can be done to extend the method for more general usage scenarios with compression of not only vertex positions, but also texture coordinates and normals.

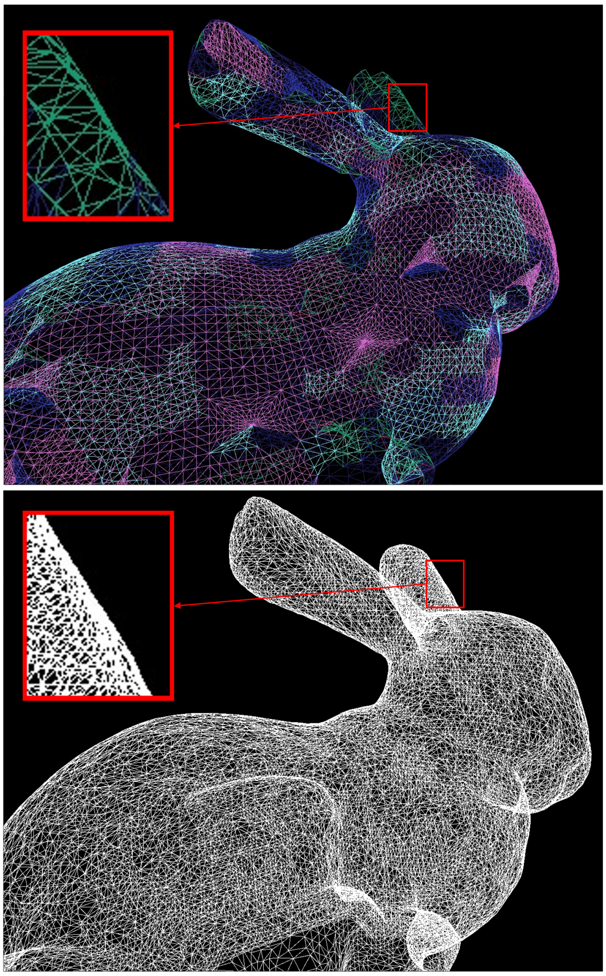


Figure 8: Visual comparison of rendering with LODs and the proposed approach.

REFERENCES

[All01] Pierre Alliez and Mathieu Desbrun. “Valence-driven connectivity encoding for 3D meshes”. In: *Computer graphics forum*. Vol. 20. 3. Wiley Online Library. 2001, pp. 480–489.

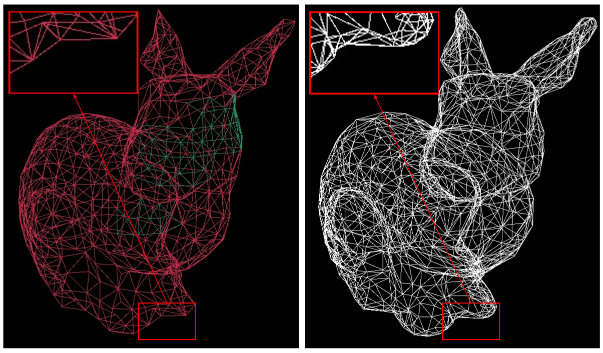


Figure 9: Visual comparison of rendering with LODs and the proposed approach.

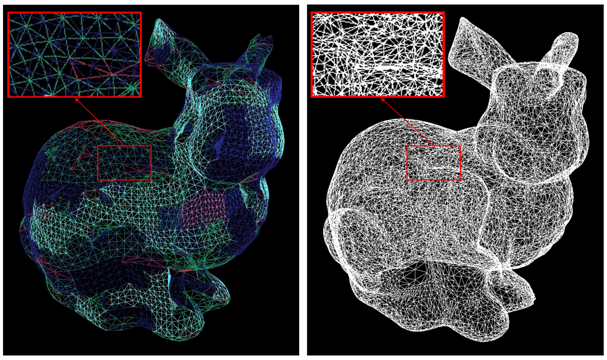


Figure 10: Visual comparison of rendering with LODs and the proposed approach.

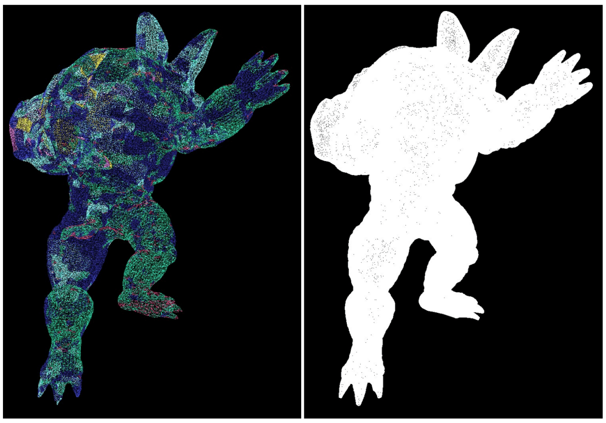


Figure 11: Visual comparison of rendering with LODs and the proposed approach. It can be seen that the proposed approach generates less number of sub-pixel triangles.

[Cal02] Dean Calver. “Vertex decompression in a shader”. In: *ShaderX: Vertex and Pixel Shader Tips and Tricks* (2002), pp. 172–187.

[Cho02] Peter H. Chou and Teresa H Meng. “Vertex data compression through vector quantization”. In: *IEEE Transactions on Visualization and Computer Graphics* 8.4 (2002), pp. 373–382.

[Cho08] Sungyul Choe, Junho Kim, Haeyoung Lee, and Seungyong Lee. “Random accessible mesh compression using mesh chartification”. In: *IEEE Transactions on Visualization and Computer Graphics* 15.1 (2008), pp. 160–173.

[Cho97] Mike M Chow. *Optimized geometry compression for real-time rendering*. IEEE, 1997.

[Coh99] Daniel Cohen-Or, David Levin, and Offir Remez. “Progressive compression of arbitrary triangular meshes”. In: *IEEE visualization*. Vol. 99. 1999, pp. 67–72.

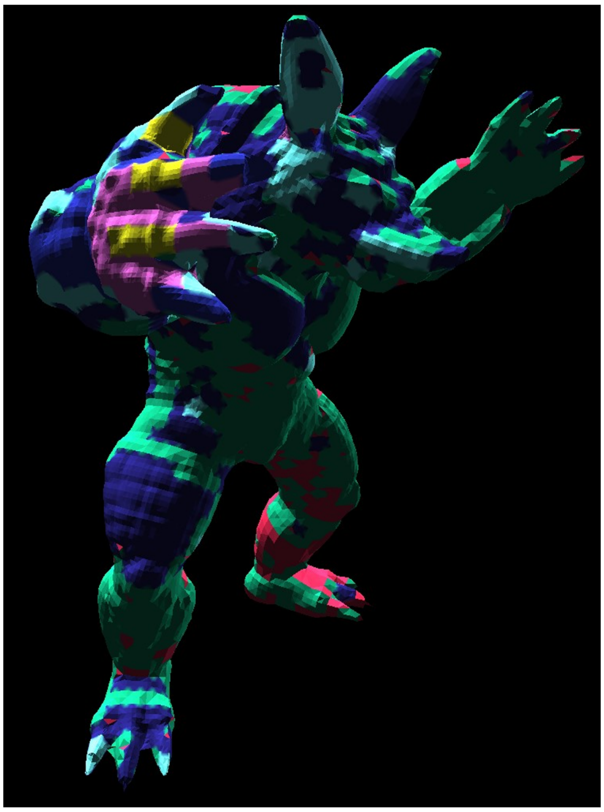


Figure 12: Sample of rendered 3D model with the proposed approach.

[Dee95] Michael Deering. “Geometry compression”. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 1995, pp. 13–20.

[Dou19] Alexandros Doumanoglou, Petros Drakoulis, Nikolaos Zioulis, Dimitrios Zarpalas, and Petros Daras. “Benchmarking open-source static 3D mesh codecs for immersive media interactive live streaming”. In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9.1 (2019), pp. 190–203.

[Eri01] Carl Erikson, Dinesh Manocha, and William V Baxter III. “HLODs for faster display of large static and dynamic environments”. In: *Proceedings of the 2001 symposium on Interactive 3D graphics*. 2001, pp. 111–120.

[Gum99] Stefan Gumhold. “Improved cut-border machine for triangle mesh compression”. In: *Erlangen Workshop*. Vol. 99. 1999, pp. 261–268.

[Hao01] Xuejun Hao and Amitabh Varshney. “Variable-precision rendering”. In: *Proceedings of the 2001 symposium on Interactive 3D graphics*. 2001, pp. 149–158.

[Hop96] Hugues Hoppe. “Progressive meshes”. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996, pp. 99–108.

[Ise05] Martin Isenburg, Peter Lindstrom, and Jack Snoeyink. “Streaming compression of triangle meshes”. In: *ACM SIGGRAPH 2005 Sketches*. 2005, 136–es.

[Jak15] Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. “Instant field-aligned meshes.” In: *ACM Trans. Graph.* 34.6 (2015), pp. 189–1.

[Jak17] Johannes Jakob, Christoph Buchenau, and Michael Guthe. “A Parallel Approach to Compression and Decompression of Triangle Meshes using the GPU”. In: *Computer Graphics Forum*. Vol. 36. 5. Wiley Online Library. 2017, pp. 71–80.

[Käl09] Felix Kälberer and Konrad Polthier. “Lossless compression of adaptive multiresolution meshes”. In: *2009 XXII Brazilian Symposium on Computer Graphics and Image Processing*. IEEE. 2009, pp. 80–87.

[Kim04] Junho Kim, Seungyong Lee, and Leif Kobbelt. “View-dependent streaming of progressive meshes”. In: *Proceedings Shape Modeling Applications, 2004.* IEEE. 2004, pp. 209–220.

[Lee02] Haeyoung Lee, Pierre Alliez, and Mathieu Desbrun. “Angle-analyzer: A triangle-quad mesh codec”. In: *Computer Graphics Forum*. Vol. 21. 3. Wiley Online Library. 2002, pp. 383–392.

[Lue03] David Luebke, Martin Reddy, Jonathan D Cohen, Amitabh Varshney, Benjamin Watson, and Robert Huebner. *Level of detail for 3D graphics*. Morgan Kaufmann, 2003.

[Mah21] Ahmed H Mahmoud, Serban D Porumbescu, and John D Owens. “RXMesh: a GPU mesh data structure”. In: *ACM Transactions on Graphics (TOG)* 40.4 (2021), pp. 1–16.

[Ros99] Jarek Rossignac. “Edgebreaker: Connectivity compression for triangle meshes”. In: *IEEE transactions on visualization and computer graphics* 5.1 (1999), pp. 47–61.

[Zha12] Jie-Yi Zhao, Min Tang, and Ruo-Feng Tong. “Connectivity-based segmentation for GPU-accelerated mesh decompression”. In: *Journal of Computer Science and Technology* 27.6 (2012), pp. 1110–1118.

Feature based CAVE software factory

Jacek Lebiedź
Faculty of ETI
Gdańsk University of Technology
ul. Gabriela Narutowicza 11/12
80-233 Gdańsk, Poland
jacekl@eti.pg.edu.pl

Bogdan Wiszniewski
Faculty of ETI
Gdańsk University of Technology
ul. Gabriela Narutowicza 11/12
80-233 Gdańsk, Poland
bowisz@eti.pg.edu.pl

ABSTRACT

In the paper we convey the lessons learned along the path we have gone through several years since establishing a room-sized CAVE installation at our university, from craft manufacturing and ad-hoc software reuse of VR software products to the robust feature driven software product line (SPL) implementing the Product Line Engineering (PLE) factory paradigm. With that we can serve all our departments and other entities from the region by rapidly instantiating different VR products based on a standard set of core assets and driven by a set of common features of VR applications destined to be deployed in the same target CAVE system – with the minimal budget and time to market requirements. A comprehensive survey of the most representative CAVE applications created in Gdansk Tech Immersive 3D Visualization Lab (I3DVL) according to PLE paradigm presented in the paper provides evidence supporting this claim.

Keywords
CAVE product portfolio, VR feature tree, Software Product Line

1 INTRODUCTION

Since its premiere in 1992, the stereoscopic video theater known since then for its recursive acronym CAVE (Cave Automatic Virtual Environment) has found many followers at universities and engineering companies reapplying the concept in a variety of fields.

Costs of a CAVE can reach a few million dollar level, especially when rear-projection is used and a significant amount of additional space for the equipment is required. All this severely limits the deployment of such installations in everyday workspaces and usually goes beyond the financial resources of typical educational institutions and small or moderate size businesses. For this reason, the total number of room-sized CAVEs exploited in the world today is low and serving a rather narrow niche market. In consequence, development of software applications dedicated for the particular CAVE would involve a different set of schematics, not necessarily making the overall software process cost-effective and of sufficient quality to justify the high investment and maintenance costs of the installation.

The motivation for writing this paper was to summarize our experience in manufacturing custom-made CAVE applications for research, education and training, commissioned by our university departments or SMEs from the region. Since 2014, we have been systematically expanding our infrastructure to its present form, including three CAVEs of different sizes integrated with large-size haptic devices, a theatre for 100+ viewers and a supercomputer, as characterized in Section 2.

Although I3DVL is basically a non-profit entity operating on a very specific market of recipients of CAVE products, all its related activities must meet the basic criteria of the commercial software process, including product assurance, cost effectiveness, time to market, productivity, quality and agility. In the paper we describe the path we have gone through all these years, from craft manufacturing and ad-hoc software reuse to the robust feature driven *Software Product Line* (SPL) [Nor12a] and convey the lessons learned in the process. Today, by reusing core assets described in Section 3 of a common set we can develop new CAVE applications based on a proven concept involving optional and variable features within the tight schedule, staffing and budgetary constraints. Their overview is given in Section 4. The existing core assets that evolve out of product development into new assets provide in turn a strong feedback loop for further development of the former keeping the product line up-to-date and reactive to the evolution of the underlying feature model.

https://www.doi.org/10.24132/CSRN.3201.27

2 THE CAVE INFRASTRUCTURE

Below we present briefly basic technical characteristics of the principal components of the infrastructure of our CAVEs and classify immersion levels its users may experience. We will refer to this classification further in the paper, when presenting the VR software factory paradigm implemented on them.

Graphics workstation onlooker experience

Four workstations, each one equipped with a graphics card driving a 27" stereo 3D display of the 2560×1440 resolution, can provide viewers wearing shutter glasses with the sense of depth when viewing images of the dynamic scene generated in real time. No immersion can be sensed for the lack of tracking devices.

The miniCAVE low-relief experience

A miniature open CAVE consisting of four displays as above and arranged as the floor and three surrounding walls allows for a miniature scale immersion of the viewer's face only, as shown in Fig. 1. Displays are driven by separate computers interconnected with the 1 Gb/s Ethernet or 40 Gb/s InfiniBand. The audio system consists of four speakers arranged above the wall displays and a subwoofer below the floor display. Shutter glasses of the viewer are equipped with the IR smart tracking system, whereas navigation is enabled by the flystick. One computer is a master for the other three slaves. The master reads data from the flystick and the tracking glasses, generates images for the central screen and broadcasts the relevant data to the slaves. Their task is to generate images synchronized with a central view.

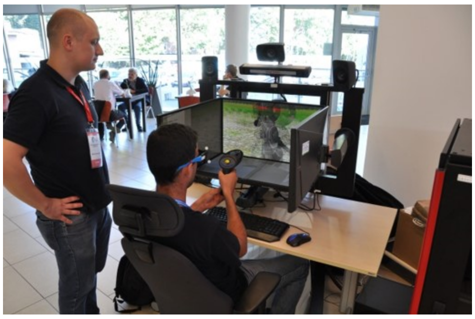


Figure 1: MiniCAVE for low-relief experience

The midiCAVE high-relief experience

A full scale immersion of the viewer in the straight-up (standing) position is provided by an open CAVE consisting of the square 2.12m×2.12m front screen wall, two side 2.12m×1.34m screen walls and the floor screen of the same size, as shown in Fig. 2. The stereoscopic image is displayed on each wall by a pair of projectors (1920×1200 for each eye) in passive technology with spectrum selection. Rear projection is used for the vertical screens, and front projection for the floor screen. Rectangular screens are operated by individual pairs of projectors, while the square front screen by two such pairs. Each pair of projectors is controlled by a single computer, interconnected alternatively with the 1 Gb/s Ethernet or 40 Gb/s InfiniBand. The audio system consists of four speakers in four upper corners of vertical screens and a subwoofer outside of the screens. The viewer is tracked in midiCAVE by eight IR cameras placed in all its corners and is enriched with a full body motion tracking system. For navigation a flystick is used.

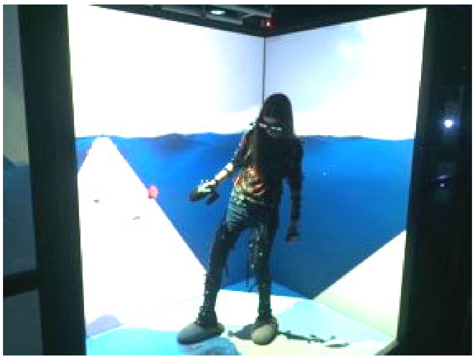


Figure 2: MidiCAVE for high-relief experience

The bigCAVE monument experience

Complete immersion is provided by the bigCAVE's six square 3.4m×3.4m walls surrounding the person inside the cube, creating impression of an unlimited space, beyond the reach of his/her arms, as shown in Fig. 3. Stereoscopic images are displayed on each screen by two 1920×1200 projectors in a rear projection mode, which gives after using the blending technique the 1920×1920 resolution of one screen in total. Stereoscopy is obtained by two alternative technologies: active Nvidia 3D Vision Pro or passive with spectrum selection. Each projector is driven by a separate server providing a rate of 60 fps for each eye. This performance level is well above the minimum requirements for a single viewer controlling the dynamic scene view reacting to his/her head movements. A fully successful attempt was therefore made to duplicate the number of images displayed on each of the six bigCAVE walls to allow two observers to control the scene view simultaneously without changing the number of projectors [Leb21a]. As a result, the 60 fps rate for each eye of each observer was reduced to 30 fps to allow for generating four frames instead of two; for most of the implemented application scenarios outlined in Section 4, these values turned out to be perfectly sufficient. Twelve computers plus two more for synchronizing the former, tracking the viewer inside and generating full dimensional sound constitute a server farm of 14 units, interconnected with 1Gb/s Ethernet and 40Gb/s InfiniBand. The audio system consists of eight speakers, two of them in each upper corner of the bigCAVE's cube and a subwoofer outside of it. The viewer is tracked by four IR cameras placed in the upper corners of the cube. One of the servers in the farm can also generate a stereo image perceived by the user inside to be transmitted (if needed) to the projector in the external large (100+ seats) 3D theatre.

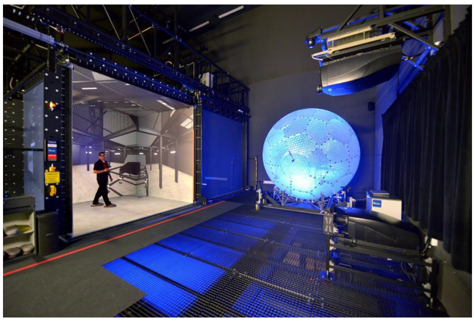


Figure 3: BigCAVE and the VirtuSphere

Large scale haptic devices

In addition to immersion perceived by the sense of sight and hearing, two haptic devices are currently being introduced in I3DVL to broaden the experience of immersion by affecting the participant's sense of balance:

- *CyberSphere* with force-feedback, founded on eight drive and measurement (DME) rolls with linear actuators driving the actual VirtuSphere with the user placed inside. Whereas the VirtuSphere is mounted on eight passive rollers evenly spaced on a round base and driven only by the body movements of user walking inside it, the actuators can measure angular velocity and adjust downforce (pressing the roller against the sphere) to provide additional feedback from the generated virtual worlds [Kow18a]. The sphere in either configuration fit in the bigCAVE cube and can be used to further extend the monument-like immersion perceived by the user, as shown in Fig. 4.

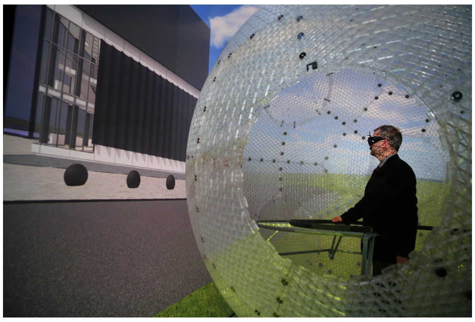


Figure 4: Virtu/CyberSphere in the bigCAVE

- *Stewart (4 DoF) platform* enabling simultaneous simulation of yaw, pitch, roll and lift with four actuators setting the platform in a sliding and rotating motion. One or more persons standing on its top can feel realistic movements of a deck of a simulated vessel (open sea experience) or the undercarriage of a land vehicle (off-road experience) [Laz94a].

HPC support

A cluster-based supercomputer composed of over 1600 computing server nodes provides nearly 1.5 PFLOPS of computational power of jointly 38500 cores and 48 GPUs. All nodes of the cluster are connected with InfiniBand FDR 56 Gb/s. Three 40Gb/s links between the supercomputer and the farm of computers driving our bigCAVE allow for sending data from it for processing as well as sending back to it voluminous data streams in real time. So far the HPC support was experimentally used by us for simulating complex weather phenomena (precipitation) and internal fires (flashover and backdraft). However, it is still the experimental facility planned for our future products.

3 THE CAVE PRODUCT PORTFOLIO

The requirements baseline for CAVE applications is typically focused on three basic attributes:

- *immersion* – high fidelity of images in terms of resolution as well as the sense of depth combined with the spatial sound should give viewers the impression of the fullest possible immersion in the generated dynamic scene of the virtual world;
- *interaction* – the generated dynamic scene views must exhibit an appropriate level of realism when reacting to the movement of viewers navigating in the related 3D space and respond promptly to any changes in their position;
- *performance* – calculation of the outcomes of viewers' actions affecting the dynamic scene content must be performed by the computing system in real-time, adequately to the dynamics of viewers.

Given the above a multitude of similar but separate VR software products could be considered – with the variety of optional features to be implemented to satisfy the end user needs. With the traditional approach to handling that developers would reuse pieces of code (small-grained reuse) [Mor02a] or larger parts of another product to build a new one (clone and own approach) [Gha18a]. However, it may be considered ineffective for CAVE products, as two applications built from the same base would be deployed and maintained in the same CAVE separately, with high duplication of effort to fix possible errors in multiple products multiple times and often implementing independently the same enhancement in different ways in similar products. If no systematic reuse of artifacts from previously implemented products is considered, or worse, no such artifacts have been collected yet, one extreme alternative would be craft manufacturing [McB02a]. This cut-and-try approach involving writing and modifying the application code by skilled programmers to make it fit in the CAVE system would dangerously reduce the latter to the role of an expensive toy for a small group of enthusiasts – unacceptable alternative to entities obliged to strictly follow the rules of financial discipline, such as our university.

When setting up the infrastructure described in Section 2 we took a closer look at common features that constitute requirements baseline for our CAVE products and the assets that constitute the core of its technical specification. Based on that our SPL adopts the underlying *Product Line Engineering* (PLE) factory paradigm and allows for optimizing effort, time and quality in product assurance of VR applications intended for use in our CAVEs. Further below we present this approach, which may provide a template solution to other universities struggling with the problem of maintaining such an expensive installation as room-sized CAVEs and its cost-effective use in everyday educational activities.

Requirements baseline – features, options

The basic tool for describing the desired product line functionality is the feature model [Fer02a], which can define the relationship of features in a family of products. It may be represented by a tree, which each vertex corresponds to one of many possible features, whereas different edges represent specific relationships of these features. In Fig. 5 we specify a tree of features of CAVE applications which can be derived from the three basic attributes listed before. There are four types of relationships between its nodes:

- *mandatory features* define the minimum functionality of a target CAVE application, i.e. each vertex, whose edge from the parent is marked with the ⊕ symbol must be selected. It may be seen that such an application should be able to provide simultaneously *immersion fidelity*, *interaction realism* and *real-time performance*, in terms of the dynamic scene *graphics* as well as *simulation* of its relevant phenomena. Moreover, *graphics* performance should concern both *static* and *dynamic* objects.
- *optional features* represent at least one option to be selected. It is assumed that from among all sibling vertices of a given parent node with the respective edges marked with the ◎ symbol, the leftmost one must be selected first, and next the other ones in the order of diminishing preference from left to right.
- *alternative features* concern exclusive choice from a set of equally preferred options. In other words, from among all sibling vertices of a given parent node with the respective edges marked with the ⊗ symbol, exactly one vertex must be selected.
- *prospective features* are planned for future applications, as new hardware and software assets will be available. Their respective edges are marked with the ∼⃝ symbol. In the future they will be merged with the ⊕ symbols marking the respective edges of their sibling vertices into the ◎ symbol, since their vertices will introduce new optional features.

Each our CAVE application allows viewers to experience immersion by attracting their senses with the spatial audio system *and* stereoscopic video solutions suitable for either a single viewer *or* multiple viewers. They enable simple navigation with a flystick (or alternatively with the inserted VirtuSphere) *and* tracking of a viewer's head based on IR with the additional *option* for tracing other body movements involving body markers. Real-time performance of the graphics is provided for *both* static objects (handling dynamic changes of each static object appearance due to the observer's head movements) and dynamic objects (moving independently across the wall screens). Simulation of the latter is supported by one computer of the CAVE farm with the *option* of additional support provided by the locally available HPC platform. Interesting extensions planned by us for future CAVE applications indicated in Fig. 5 concern the aspects of immersion which are not directly related to sensual impressions. Several applications in our product line already concentrate on *emotions* evoked by the content and meaning of the dynamic scene rather than its appearance only. Based on the research in affective computing and automatic monitoring of human emotions [Lan16a] we plan to introduce the *feedback* feature, to make applications more responsive and further deepen the immersion experience of viewers. Various combinations of the features in Fig. 5 specify the requirements baseline for our CAVE product portfolio and drives development of their respective code and technical specification. The latter constitutes a set of core assets that capture the underlying feature model and each new product is supposed to be built from that set in a prescribed way.

Technical specification – core assets

As mentioned before the primary capability of the software product line is instantiation of a final product derived from the common set of shared assets rather than crafting its code manually from scratch or reusing components of other products. The term "development" in this context means that not only the product code is being built, but many other assets that further support the overall engineering process toward the final product are also instantiated. They constitute a common platform

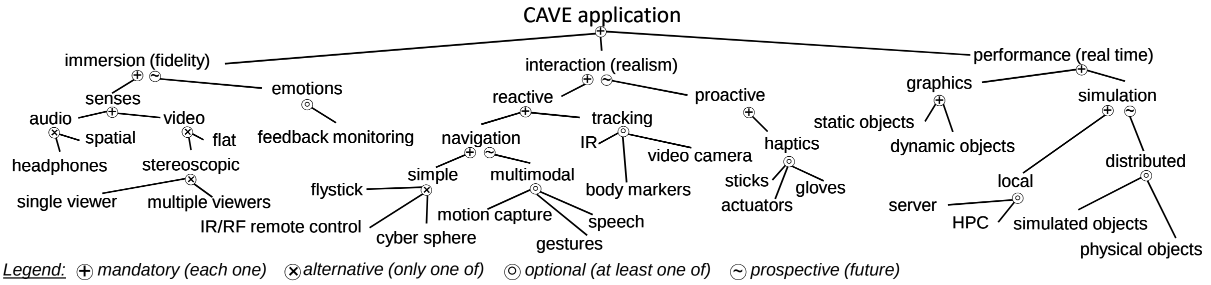


Figure 5: Feature model of CAVE applications

upon which many different products may be developed. The core assets that are shared across our SPL implementation for CAVE applications include the following categories [Nor12a]:

Requirements; incremental refinements of the basic feature model addressing technical details of its individual features (parameters and their variability, characteristics and attributes appropriate for the given software product instance, its target audience).

Architecture; a basic scheme for assembling the product from software components in the core asset base instantiated for each individual product.

Software components; basic building blocks for each individual product, reused without alteration or altered using inheritance or parametrized templates.

Performance models; hard and soft real time constraints, qualitative characteristics and methods for determining them, including image rendering, synchronization and switching.

Cost/schedule; generic framework for work schedule development and time and effort estimates for the entire product line.

Tools/processes; support for software product development and making changes, appropriate for the entire product line and production stages.

Test cases/data; generic testing artifacts suitable for all products in the product line and extensible to accommodate variation among them.

People/skills; evolving from programming toward relevant domain expertise and technology forecasting.

The key assets in the architecture and tools and processes categories of our SPL are the CAVE infrastructure specified in Section 2 and any modern game engine (GE) satisfying the criteria defined in [Pet12a]. Each such GE provides an IDE and resources enabling rapid development of high-fidelity virtual worlds by handling in the uniform way complex computations related to rendering of the dynamic scene, simulating physics of dynamic objects, detecting collisions, etc. Thanks to their modular structure, they can be repeatedly reused to instantiate a variety of VR products, very similar to what SPL provides. Currently our principal GE asset is Unity and Unreal Engine [Pet12a]. The initial content of the assets listed above are the following:

Requirements; Features selected from the baseline set (limited to the single 3D flat-screen display view, keyboard and mouse). Additional target domain-related requirements (external asset) added.

Architecture; Reference architecture selected (Unity or Unreal Engine). Development framework set up (non-stereoscopic 3D graphics display, keyboard and mouse).

Software components; The candidate application code and documentation (external asset) added.

Performance models; Scene rendering complexity (graphics performance, navigation reactivity) and dynamics of simulation objects assessed.

Cost/schedule; Additional candidate application cost and schedule (external asset) added.

Tools/processes; Toolset specific to the selected platform acquired (Unity, Unreal Engine or other).

Test cases/data; Preliminary acceptance criteria defined. The candidate application formally qualified as the product line asset (regular graphics workstation, single flat-screen display, keyboard and mouse).

People/skills; Candidate application domain specific skills (external asset) added.

Note that the core set is extended by external assets brought to the project by a candidate application that will be finally instantiated and deployed in the big-CAVE system as the stereoscopic 3D one. Besides this application's code and documentation other external assets include such domain-specific artifacts as the additional target domain requirements, specific performance characteristics and staff skills. The baseline requirements at this stage are defined by the feature model limited to the standard 2D user interface (display, keyboard and mouse). In the following production stages they are gradually refined, respective to the capabilities of the target CAVE system. The internal technical staff that can be assigned to any production activity in I3DVL varies from one to three persons, one more team leader person and a couple of external target domain experts. The internal staff roles and responsibilities require the skills of a graphics designer, programmer, tester and instructor – one person may simultaneously play several roles at various stages of the production process.

Production stages and product decisions

Although PLE can capitalize on commonality of final product instances destined for the same target system, the preliminary set of core assets listed before may exhibit variation due to inherent variability of the underlying feature model specified in Fig. 5. Note that depending on the particular application purpose and behavior individual products may be instantiated in different ways. Managing this variation in the production process requires a clear definition of internal variation points where appropriate product decisions are made. The point in the production timeline at which the decisions for a variation point are bound is referred to as the *binding time* [Ros11a]. Our SPL utilizes multiple binding times determining four production stages outlined in Fig. 6.

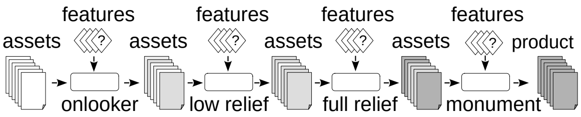


Figure 6: CAVE application SPL production stages

The respective production stages are marked with rounded rectangles and their relevant binding decisions are marked with diamonds. By starting with the preliminary set of assets listed before the product outputs from binding decisions at one production stage become partially instantiated asset inputs for binding decisions at the next production stage. The progress of instantiation of assets toward the final product is symbolically marked by darkening gray.

Workstations of the *onlooker* stage mimic configuration of the computers controlling projection on a single wall of bigCAVE. Owing to that developers can assess the quality of the depth and the overall design and logic of the content of stereoscopic images that the prospective application is supposed to deliver to each screen at its target installation. *Low-relief* stage utilizes the miniCAVE system for testing synchronization of images in side and floor screens with the central (front screen) view in the master-slave fashion, in particular simultaneous changes of views on all displays, joining adjacent screen views and verifying correctness of virtual cameras definitions on the master and slaves. *High-relief* stage utilizes midiCAVE to test synchronization of images in the application under development and verifying correctness of all virtual cameras definitions and assessment of blending of images on the central (front) screen generated by two slaves. Moreover, scenarios involving body motion capture may be implemented and multiple viewers may be involved if needed. *Monument* stage is the final one, as the bigCAVE system is the target environment for our VR applications. Blending of images on each screen wall is tested to qualify the application to the final acceptance testing of image synchronization.

For brevity we skip detailed description of the evolution of each core asset of the set into the final one, but remarks on some of them are in place (see Fig. 6): the application *architecture* asset evolves from a standard flat screen through a single stereoscopic 3D form up to the master-slave form of multiple screens, the *software components* asset expands from the initial application code to the form augmented with synchronization and interaction objects from our own CAVE library developed for that purpose, and the *performance models* asset expands from the basic quality (graphics performance, navigation reactivity and dynamics of simulation objects) characteristics to multiple screens synchronization quality characteristics, such as geometric continuity, 3D image stitching and consistency of adjacent images. Besides a GE platform for going from flat screen display to the 3D stereoscopy we use popular tools for unwrapping mono or stereo audio sources to surround formats [Tho07a].

The reason why our primary preliminary software component asset is mostly a regular non-stereoscopic 3D graphics flat-screen application with the mono or stereo sound is quite simple: most of the interesting domain applications are created in various departments of our university on ordinary flat-screen workstations and not necessarily with the intention of their subsequent implementation in CAVEs. The current ease of developing them thanks to the widely available and popular GE platforms (Unity in particular) means that several dozen of them are created at our university every year. They are systemically delivered by students as part of class projects or built by our faculty as 'proof of concept' in various R&D projects. Some the best of them are deployed over time in the university's didactics and become qualified candidates to convert them to the fully immersive, interactive and high-performance CAVE applications. There are also individual cases of building dedicated applications for research in the field of computer graphics, such as ray tracing in heterogeneous environments for example [Kacz21a]. I3DVL also receives ocassionally a small number of candidate applications developed by the third parties (mostly co-operating SMEs) with a request to deploy them in the bigCAVE.

Such a clear separation of the two phases of the bigCAVE application life cycle promotes agility and significantly reduces costs: for the initial flat-screen version phase the primary focus is on the semantic correctness of the scene content, its logic and realism, whereas for the phase described in the paper the proper development of the CAVE application takes place.

Assets of a bigCAVE product instantiated in the monument stage finally take the following form:

Requirements; synchronization of six screens and flystick functionality.

Architecture; multiplication of a single screen application to the master-slave (front vs. side, floor and ceiling screens) configuration.

Software components; master-slave synchronization and interaction objects (the own CAVE library).

Performance models; navigation/synchronization quality characteristics (geometric continuity, 3D image stitching, consistency of graphical effects in adjacent images).

Cost/schedule; fixed cost model (master-slave parallelization of six screens, two person-days, linking the single screen application to the bigCAVE interface one person-day).

Tools/processes; configuration and deployment of master-slave synchronization objects and interaction objects from the CAVE library with .tools from the relevant development framework.

Test cases/data; acceptance testing of the final application wrt. the feature model baseline requirements (immersion, interactivity, real-time performance).

People/skills; one person (tester and programmer).

4 CAVE APPLICATIONS OVERVIEW

Each CAVE application in our SPL has been instantiated in the production stages outlined in Fig. 6 out of some initial flat-screen application and the preliminary set of core assets mentioned before. For the sake of brevity, we will not place a complete bibliographic record for each of the applications discussed below, and only highlight their basic effort and time characteristics. Their purpose varied from teaching or training, through prototyping (devices, buildings, structures) or treatment of patients, up to pure or applied research.

For example, the Old Town application (Fig. 7) was aimed to synthesize visually new planned buildings in the existing architecture under variable day/night lighting and weather conditions (prototyping), but also involved teaching (objects can be designed by students). Viewers interact with the dynamic scene by selecting time of the day and weather conditions and navigate with a flystick along a predefined path or teleport to selected places [Leb16b]. Total instantiation time of this application was three months and involved two programmers or testers, supported by three teams of architects (nine persons in total).

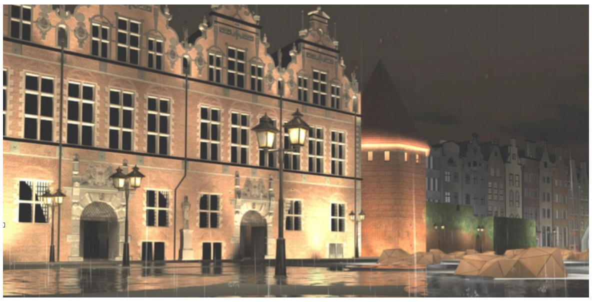


Figure 7: Old Town interactive visualization (drizzle)

The purpose of the interactive small UAV simulation application (Fig. 8) was also prototyping – a virtual drone may be reconfigured and reequipped with various simulated on-board devices like cameras or sensors. It can be controlled with a specialized controller in three different views: from the ground, from the cockpit, and from behind the drone's tail. Additionally, to enable monitoring of the UAVs in-flight state a standard instrument panel may be displayed on demand on the CAVE's wall currently pointed by the controller. Total development time of this application was three months and involved just one programmer/tester, supported by a single domain expert (pilot).

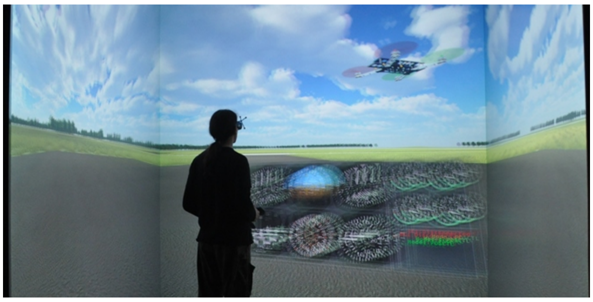


Figure 8: A small drone platform tester and trainer

These two applications are similar in the aspect of simulating weather phenomena, but totally different with regard to the dynamics of the simulation objects involved. Nevertheless, they both were instantiated out of the same set of core assets and, as indicated before, within the similar staffing and time-to-market limits.

Another applications in our SPL involving simulation of basically the same physical phenomena (gas flows), are the small fire simulator (Fig. 9) and a sail-boat in the virtual wind-tunnel (Fig. 10). The former may be used to train people how to operate a jet from a handheld fire extinguisher to quickly suppress a small fire in an office room, whereas the latter to make them understand settings of the sail for different courses of the boat in relation to the wind direction; individual bands and their colors show the direction and speed of the airflow [Leb16a]. Owing to the more complex simulation of the hot (flame) gases the instantiation time of the first application was four months, whereas for the latter it was just one month. In both cases only one programmer/tester was needed, supported by the relevant domain expert (respectively a fireman and a sailor).

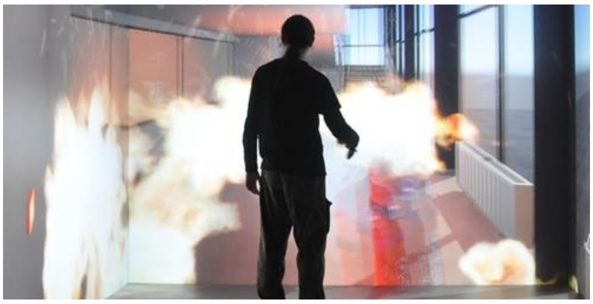


Figure 9: A small fire and extinguisher simulator

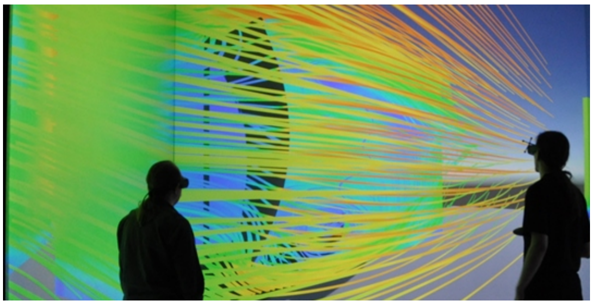


Figure 10: Simulation of an airflow around the sailboat

Another segment of our products are CAVE applications developed by our students in cooperation with the Polish Space Agency. The average instantiation time of each application in this segment was three months, with one programmer/tester and one domain expert (astronomer) involved. These applications address various issues related to understanding the Solar System [Mar22a], exploration of celestial bodies, detection of black holes with gravitational lensing, and visualization of constellations from various places on Earth, among others. They all take advantage of the possibility to directly immerse viewers in the visualized phenomena to help them understand its essence and course and are intended for primary and secondary school students. These applications use our bigCAVE with a few individuals inside, and a larger audience in the connected video theater outside. For example the application shown in Fig. 11 helps viewers to understand distances between planets of the Solar System, types of its celestial bodies (the Sun, planets, asteroids) and mechanics of their orbits. Users can navigate using teleportation and observe them from various places of the Solar System.

Figure 11: Interplanetary travel in the Solar System

Another application of the space segment in our portfolio is the Moon/Mars rover simulator (Fig. 12). The rovers are simulated according to the gravity of the respective celestial body and real hypsometric data of the explored terrain (Apollo LEM and Curiosity rovers).

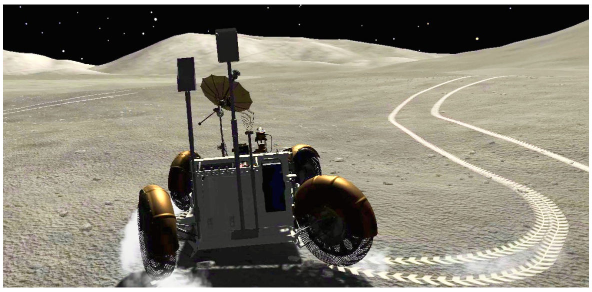


Figure 12: LEM lunar rover simulator

An important segment of our SPL are biomedical applications developed for research into therapies of various kinds based on immersion. One example is shown in Fig. 13; this application is intended to treat acrophobia (height anxiety) with the implosive therapy augmented with gamification. The task of the patient is to move from a safe room to various levels of a tall building and navigate along narrow gangways, suspended high between various buildings and collecting coins of different value along the way. A psychologist supervising the exercise can give advice to the patient inside, comment on decisions of the latter and encourage him/her to choose specific optional segments of the path. The instantiation time of this application was three months and involved three programmers/testers, including development of its initial version, as the preliminary software component asset. The supporting team of domain experts included five psychologists.

A biomedical application shown in Fig. 14 was developed for research into the diagnosis of dizziness. A patient is immersed in an unstable space of a tunnel built of rotating rings and with accompanying unpleasant sounds. Reactions of the patient can be measured in

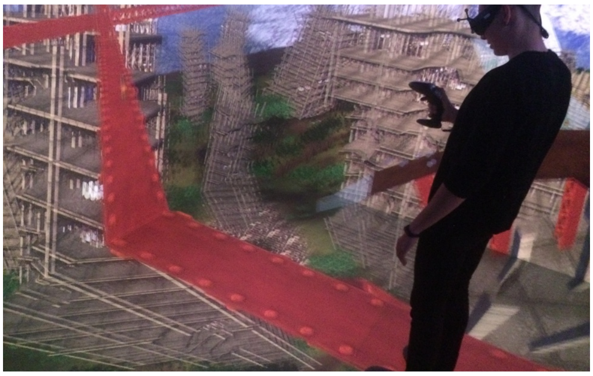


Figure 13: Acrophobia treatment game

a couple of ways: by recording the pressure exerted by each patient's foot on the ground, or recording electrical activity of the brain with a portable EEG, or both. The instantiation time of this product was three months, involved one programmer/tester and one domain expert (a biomedical engineer).

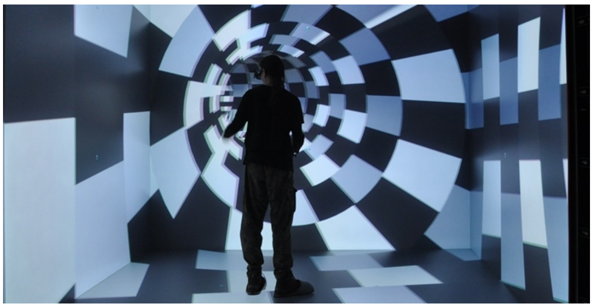


Figure 14: Stimulation for dizziness measurements

An interesting use of the bigCAVE in I3DVL is a virtual escape room in which puzzles play an educational role. The player's task is to guess the codes that open the door to leave the virtual room shown in Fig. 15. The codes to open individual locks can be obtained by solving chemical puzzles, assembling molecule models or performing virtual chemical reactions. The application was prepared by a team of three in two months, supported by a specialist chemist.

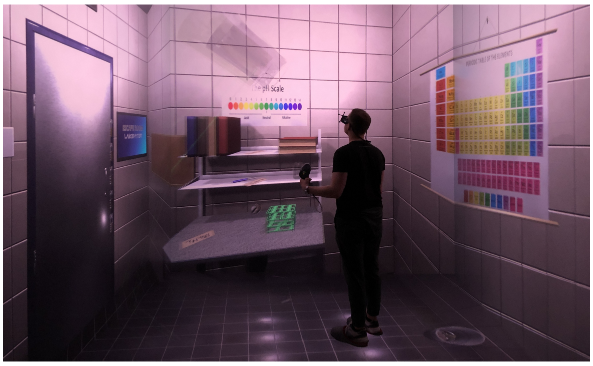


Figure 15: Chemical escape room

5 CONCLUSION

The survey of just a small fraction of our SPL portfolio of well over one hundred CAVE applications instantiated so far was intended to prove the point on PLE, which when based on a well defined set of core assets can make it possible to optimize development of CAVE applications in terms of both time and staffing requirements. The product delivery time (or so to speak time-to-market, although still within the university) for any of our applications rarely exceeded three months with at most three programmers/testers required to instantiate the product for the target bigCAVE system. We have demonstrated that the SPL paradigm can provide a cost-effective solution for rapid instantiation of CAVE applications. Thanks to the fact that it can exploit a common, managed set of features of software products destined to be deployed in the same target CAVE system, the otherwise different products can be developed as one. By defining the feature model for our product line, generating core assets from our first products (at that time crafted for bigCAVE) and setting up next the appropriate production stages supported by the specialized hardware, we were able to keep the delivery time within the limits imposed by the academic schedule and optimize development time and effort while keeping the size of the development team small. Adoption of the SPL approach also contributed greatly to improving quality of each new final product instance released, as functional and performance testing of the latter in various production stages helped a lot in fixing potential problems of the shared assets (adding a new variation point, refining the performance model, adding new test cases, etc.), thus improving the entire family of products and not just that one product instance.

6 REFERENCES

[Fer02a] Ferber, S. et al. Feature interaction and dependencies: Modeling features for reengineering a legacy product line, in Proc. 2nd Int. Conf. Software Product Lines SPLC 2, pp.235–256, San Diego, CA, USA, 2002.

[Gha18a] Ghabach, E. et al. Clone-and-Own software product derivation based on developer preferences and cost estimation, in Proc. 12th Int. Conf. on Research Challenges in Information Science RCIS 2018, pp.1–6, Nantes, France, 2018.

[Kacz21a] Kaczmarek, A., Lebiedź, J., Jaroszewicz, L., and Święszkowski, W. (2021). 3D scanning of semitransparent amber with and without inclusions. In *Proc. 29th Int. Conf. on Computer Graphics, Visualization and Computer Vision (WSCG 2021)*, pages 145–154, Plžen, Czech Republic, May 17-20 (on-line).

[Kow18a] Kowalczuk, Z. and Tatara, M. Sphere drive and control system for haptic interaction with physical, virtual, and augmented reality, IEEE Trans. Control Syst. Technol. 27, No.2, pp.588–602, 2018.

[Lan16a] Landowska, A., Szwoch, M., and Szwoch, W. Methodology of affective intervention design for intelligent systems, Interacting with Computers 28, No.6, pp.737–759, 2016.

[Laz94a] Lazard, D. and Merlet, J. The (true) Stewart platform has 12 configurations, in Proc. 1994 IEEE Int. Conf. on Robotics and Automation, pp.2160–2165, San Diego, CA, USA, 1994.

[Leb21a] Lebiedź, J. and Mazikowski, A. (2021). Multiuser stereoscopic projection techniques for CAVE-type virtual reality systems. *IEEE Trans. Human-Mach. Syst.*, 51(5):535–543.

[Leb16a] Lebiedź, J. and Redlarski, J. (2016). Applications of Immersive 3D Visualization Lab. In *Proc. 24th Int. Conf. on Computer Graphics, Visualization and Computer Vision (WSCG 2016)*, pages 69–74, Plžen, Czech Republic, May 30-June 3.

[Leb16b] Lebiedź, J. and Szwoch, M. (2016). Virtual sightseeing in Immersive 3D Visualization Lab. In *Proc. Fed. Conf. on Computer Science and Information Systems (FedCSIS 2016)*, pages 1641–1645, Gdańsk, Poland, Sept. 11-14.

[Mar22a] Martinez, K., Lebiedź, J., and Bustillo, A. (2022). Graphical interface adaption for children to explain astronomy proportions and distances. In *Proc. 30th Int. Conf. on Computer Graphics, Visualization and Computer Vision (WSCG 2022)*, pages 1–8, Plžen, Czech Republic, May 17-19 (on-line).

[McB02a] McBreen, P. Software craftsmanship: the new imperative, Addison-Wesley, 2002.

[Mor02a] Morisio, M., Ezran, M., and Tully, C. Success and failure factors in software reuse, IEEE Trans. on Soft. Eng. 28, No.4, pp.340–357, 2002.

[Nor12a] Northrop, L. and Clements, P. A framework for software product line practice, version 5.0, white paper REV-03.18.2016.0, Software Engineering Institute, Carnegie Mellon University, 2012.

[Pet12a] Petridis, P. et al. (2012). Game engines selection framework for high-fidelity serious applications, Int. J. of Interactive Worlds, art. ID 418638.

[Ros11a] Rosenmüller, M. et al. Flexible feature binding in software product lines, Automated Software Engineering 18, No.2, pp.163–197, 2011.

[Tho07a] Thorton, M. (2007). Surround sound from stereo. Upmixing plug-ins for Pro Tools. *Sound on Sound*, https://www.soundonsound.com/reviews/surround-sound-stereo.

Self-Organising Maps for Efficient Data Reduction and Visual Optimisation of Stereoscopic based Disparity Maps

Simone Müller
0000-0001-5830-8655
Leibniz Supercomputing Centre (LRZ)
Boltzmannstrasse 1
85748 Garching bei München
simone.mueller@lrz.de

Dieter Kranzlmüller
0000-0002-8319-0123
Ludwig-Maximilians-Universität (LMU)
MNM-Team
Oettingenstr. 67
80538 München
kranzlmueller@ifi.lmu.de

Abstract

Many modern autonomous systems use disparity maps for recognition and interpretation of their environment. The depth information of these disparity maps can be utilised for point cloud generation. Real-time and high-quality processing of point clouds is necessary for reliable detection of safety-relevant issues such as barriers or obstacles in road traffic. However, quality characteristics of point clouds are influenced by properties of depth sensors and environmental conditions such as illumination, surface and texture. Quality optimisation and real-time implementation can be resource intensive. Limiting the amount of data allows optimisation of real-time processing. We use Kohonen network existing self-organising maps to identify and segment salient objects in disparity maps. Kohonen networks use unsupervised learning to generate disparity maps abstracted by a small number of vectors instead of all pixels. The combination of object-specific segmentation and reduced pixel number decreases the memory and processing time towards real-time compatibility. Our results show that trained self-organising maps can be applied to disparity maps for improved runtime, reduced data volume and further processing of 3D reconstruction of salient objects.

Keywords

Kohonen Networks, Self-Organising Maps, Depth Image Segmentation, Disparity Maps, Computer Vision

1 INTRODUCTION

Modern fields such as autonomous driving or robotics are computationally intensive and expensive. Real-time based data processing is a mandatory requirement in autonomous vehicles [Mau15]. Spatial image data must be processed in time to detect objects and prevent possible collisions. The depth information of 3D reconstructions in space can be determined by disparity maps of stereoscopic image data. However, the quality of 3D reconstructions is affected by noise, distortion or blurring effects in the images. The use of volumetric data sets can optimise image effects, but real-time processing requires the limitation of processed data volume. The size of volumetric data causes high latency, time delays and additional occurrences of errors.

AI-based computer vision is an promising approach for real-time closed environmental perception. There are sophisticated AI algorithms for the recognition of objects in 2D images, such as YOLOv3 [Red18], Faster Region Based Convolutional Neural Networks (R-CNN) [Ren17] and Multi-Scale Convolutional Neural Network (MSCNN) [Cai16]. However, the previously trained 2D object recognition is limited to concrete object classes and visual interference effects of the images. In order to collect ambient information and make accurate decisions with high confidence, the AI usually needs to be trained extensively with many data sets. Some existing non-trained algorithms detect salient objects based on depth images. These types of algorithms process each pixel of a depth image, which affects the runtime [Hof19]. Even the merging of pixels into colour images is not suitable for a data-reduced and performant application [Ju14].

Our motivation is based on the challenges of AI-data reduction and visual optimisation. By using self-organising maps (SOM), we intend to abstract disparity maps to a reduced number of vectors since a reduced disparity map has a positive impact on computational and memory-based latency. The corresponding SOM algorithm reduces and accelerate the object detection

https://www.doi.org/10.24132/CSRN.3201.28

to a few vectors instead of the necessary total pixels. This allows the runtime optimisation.

Our contribution comprises the following aspects:

- Kohonen based depth image segmentation for efficient data reduction
- Concept and implementation of the SOM-algorithm for validation of our approach
- Analysis of data reduction, reliability, runtime and stability of the disparity based SOM-algorithm

Our evaluation reveals the stability and transferability of Kohonen based homogeneity detection of disparity distributions in form of self-organising maps. For this purpose, we use synthetically generated data from the Unreal Engine.

The paper is organised according to a fixed structure consisting of related work, concept and methodology of SOM optimised disparity matching, evaluation, conclusion as well as future work.

2 RELATED WORK

This section describes different related approaches and interrelationships of unsupervised AI technologies for object detection, segementation as well as performance optimisation.

Class-based Object Detection Object recognition is used across industries as a technology for image-based classification and localisation [Fri22]. Popular AI-algorithms such as YOLOv3, Faster R-CNN and MSCNN use a class-verifying diversity of object propositions for object recognition [Cai16, Mue18, Red18, Ren17]. The number of object proposals can be significantly reduced by using disparity images. General sliding window recognition based methods generate several object proposals of different sizes per sliding window position [Kri19].

Mueller et al. reduce the number of object proposals by creating only one object candidate per sliding window position [Mue18]. The object size is predicted by information such as object dimensions, disparity-based depth information or intrinsic sensor parameters. Most of the detected objects of [Mue18] are additionally upright so that the area of a candidate object has an approximately constant disparity distribution. A homogeneous disparity distribution is collected to remove inappropriate object proposals. The disparity values of all pixels within a candidate object are examined for homogeneity. In case of insufficient disparity homogeneity, the object proposal will not be considered further [Mue18]. The excessive number of object candidates precludes the suitability of this method for disparity based data reduction.

A further possibility for data reduction is the area-reduced creation of disparity map [Pon20]. Objects of certain classes are identified and localised as input defined stereoscopic images. The MS-CNN algorithm is used for this object recognition [Cai16, Pon20]. The pairwise matching object areas from the content bounding box of the respective stereoscopic images can be determined by a similarity check [Pon20]. The disparity is then determined only for the pixels which represent an object in a matching object area. Limitation resides in the recognition of previously trained object classes.

Superpixel Object Recognition The identification of visually prominent and conspicuous areas are a conducive skill for salient objects recognition [Bor19]. Ju et al. describe an algorithm that processes information from input images and depth maps to create a striking map for detecting salient objects [Ju14]. Thereby, the striking map shows how salient each individual pixel is in comparison to other pixels. Striking objects usually stand out from the surrounding background. Superpixels can be defined by grouping the pixels with same properties like pixel intensity [Jai19]. Ju et al. show that object recognition can be improved with additional depth information in relation to raw colour informations. Salient objects in colour can appear inconspicuous, whereas 3D perceptions are conspicuous. Until now saliency maps have not been optimised for reducing data. Additional steps, such as the creation of a binary map, are necessary to determine relevant sections of the disparity map [Jai19].

Colour and Depth Image Segmentation A division of the depth maps and its colour images into different segments allows the detection of salient objects [Hof19]. Through this approach, each pixel contains additional depth information after segmentation. Fig. 1 describes a search window method to distinguish the object affiliation from the foreground and background.

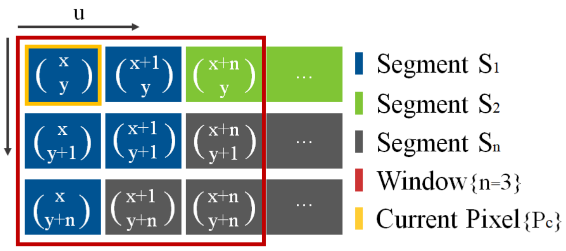


Figure 1: **Square Search Window** [Hof19]: Describes the method for distinguishing object affiliation between the foreground and background. The squares denote the individual segments of S_1, S_2 and S_n.

Each pixel of the depth map is analysed in a search window for the same segment and the greatest depth. The sum of depth gradients in a segment is subsequently used to allocate foreground or background objects [Hof19]. This methodology leads to a large number of false detections. A considerable number of depth differences is determined on the basis of neighbourhood observation. The segmentation in [Hof19] is based on the depth image as well as the colour image, but depth differences are still formed for all pixels during object recognition. This affects the running time.

SOM SOM offers promising properties for image segmentation through the use of image features such as colour, texture and pixel intensity [Kau14]. The Kohonen-referred SOM enables the visualisation of high-dimensional data by abstracting original colour sets of matrix-based input data sets [Koh01, Vet18]. The continuous colour set reduction of each pixel allows the following segmentation of the individual areas. Fig. 2 describes the individual processing steps of a SOM.

Initialisation of Reference Vectors
⇓
Iterative Learning Process
⇓
SOM Adjustment

Figure 2: **Pipeline of SOM [Koh01]**

Initialisation of Reference Vectors A classical SOM is represented by a n-dimensional array of neurons. Each neuron μ is connected to a reference vector m_{i} (Eq. 1).

$$m_{\mathrm{i}} = [\mu_{\mathrm{i}1}, \mu_{\mathrm{i}2}, \ldots, \mu_{\mathrm{in}}]^T \in R^n \qquad (1)$$

The arbitrary selectable values of m_{i} must be initialised at the start time t_0. [Koh01]. The reference vectors must have the same dimensions as the input data set. An iterative learning process is carried out after the reference vectors have been initialised.

Iterative Learning Process The input data set shall be defined as a colour-based [R_{n},B_{n},G_{n}] input vector x $\in R^n$. From the distance between input vector and all reference vectors, we can determine the neuron that most resembles x. A neuron with the greatest similarity to the input vector x is defined as winner neuron c. The Euclidean distance $d_{\mathrm{E(a,b)}}$ is suitable for calculating the distance between the vectors a and b (Eq. 2).

$$d_{\mathrm{E(a,b)}} = \sqrt{(\zeta_1 - \eta_1)^2 + (\zeta_n - \eta_{\mathrm{n}})^2} \quad \begin{cases} a = (\zeta_1, \zeta_{\mathrm{n}}) \\ b = (\eta_1, \eta_{\mathrm{n}}) \end{cases} \qquad (2)$$

The direct winner neuron connected reference vector as well as certain reference vectors are adjusted until they are similar to the red coloured input vector x of Fig. 3.

The time-dependent neighbourhood function $h_{\mathrm{ci(t)}}$ can be used to determine a neuron in the updating neighbourhood.

$$h_{\mathrm{ci(t)}} = \alpha_{\mathrm{(t)}} \cdot exp\left(-\frac{||r_{\mathrm{c}} - r_{\mathrm{i}}||^2}{2\sigma^2_{\mathrm{(t)}}}\right) \qquad (3)$$

$h_{\mathrm{ci(t)}}$ defines the learning degree of a corresponding reference vector from the input vector x as shown in Fig. 3.

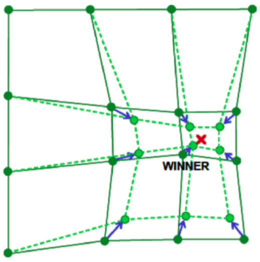


Figure 3: **Determined Winner Neuron** [Kah19]: Update of the selected reference vectors in x-direction

$\alpha_{\mathrm{(t)}}$ is described as the learning rate and indicates how strongly the reference vectors should learn from the input vector at time t. The integer value t in Eq. 3 denotes the discrete-time coordinate. By using the neighbourhood radius $\sigma_{\mathrm{(t)}}$, the neuron distances to the winning neuron can be determined.

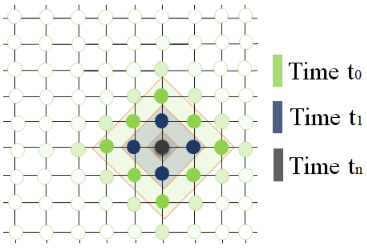
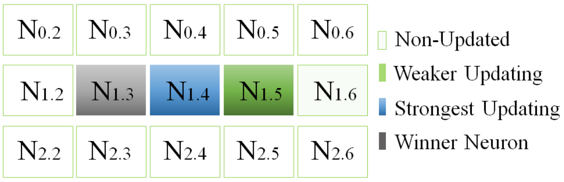


Figure 4: **Neighbourhood Building** [Kah19]: Temporal intensity decrease of the neural neighbourhood. The winner neuron is located in the centre of the coloured neurons.

The functional temporal and monotonic decrease of $\sigma_{\mathrm{(t)}}$ causes the equivalent decrease of neighbour neurons around the winning neuron (Fig. 4). Updating the location vector of the winning neuron c ($r_{\mathrm{c}} \in R^n$) and the further neurons i ($r_{\mathrm{i}} \in R^n$) decreases with increasing distance (Fig. 5).



N0.2	N0.3	N0.4	N0.5	N0.6
N1.2	N1.3	N1.4	N1.5	N1.6
N2.2	N2.3	N2.4	N2.5	N2.6

Figure 5: **Distance-Dependent Neuron Updating** [Kah19]: Representation of the location vector based distance influence on the map update. The neuron $N_{1.4}$ is updated more than $N_{1.5}$ due to its closer distance to the winning neuron $N_{1.3}$.

By applying the time-dependent and monotonically decreasing learning rate $\alpha_{\mathrm{(t)}}$, we can determine how fast the reference vectors learn from the input vector Eq. 4.

$$m_{\mathrm{i(t+1)}} = m_{\mathrm{i(1)}} + h_{\mathrm{ci(t)}}[x_{\mathrm{(t)}} - m_{\mathrm{i(t)}}] \qquad (4)$$

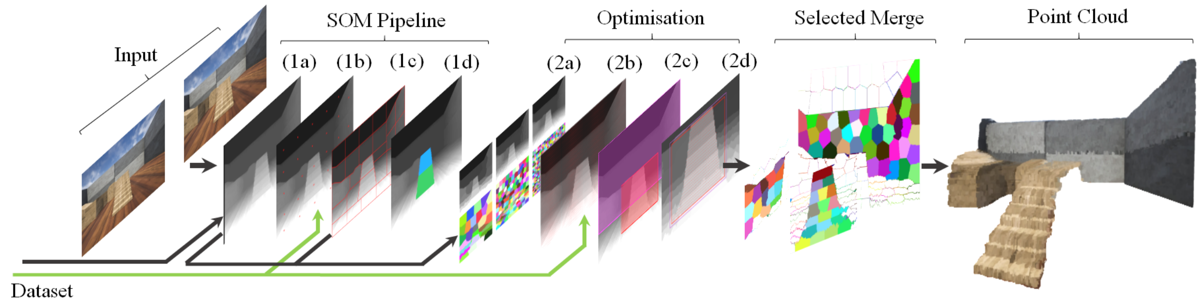


Figure 6: **Pipeline of SOM based Disparity Optimisation**: The pipeline is structured in 4 main steps consisting of actual SOM algorithm, optimisation, merging of object reduced SOM maps and the point cloud rendering.

A special feature of SOM is the dimensionally ordered adoption of reference vectors whereby the reference vectors are similar between neighbouring neurons. Ordering of reference vectors takes place within the initial phase of the iterative learning process. The global order of SOM requires an initial neighbourhood radius of sufficient size and high learning rate values.

The initialisation is followed by the final adjustment of SOM. For this purpose, the learning rate of nearest neighbour neurons should assume small values over a long period of time. The iterative process can be accelerated by initialising already ordered reference vectors [Koh01].

3 KOHONEN NET OPTIMISED DISPARITY MAPPING

In this section we present our SOM based concept for data reduction of disparity maps. Highlighted object areas are identified for this purpose within a disparity map by using kohonen-based SOM. The reduced disparity map has a positive impact on computational and memory-based latency.

Fig. 6 describes the main pipeline of SOM-based data reduction. The steps 1a-1d of the SOM pipeline indicate the following contents (Fig. 7):

1a: Disparity map replication on SOM
⇓
1b: SOM based segmentation of disparity Map
⇓
1c: Object area recognition based on SOM
and segmented disparity map
⇓
1d: Disparity map reduction through
identified object areas

Figure 7: **SOM Pipeline as shown in Fig. 6**

The optimisation of SOM (Fig. 6) occurs through the following parameter and function adjustment:

- SOM dimension
- Input dataset
- Training sequence
- Initial learning rate and neighbourhood radius
- Disintegration function

1a: Disparity Replication Disparity mapping to SOM consists of creating the input dataset, SOM initialisation and training with a used input dataset. The input data set for the SOM algorithm is defined as dimensional matrix (Fig. 8). The original calculation of the disparity map may include negative disparity values. To create the input data set, the origin disparity map must be reduced by the negative values.

$$
\begin{array}{c|ccccccc}
 & [d_{[0,0]} & d_{[0,1]} & d_{[0,2]} & \dots & d_{[0,x-3]} & d_{[0,x-2]} & d_{[0,x-1]}] \\
 & [d_{[1,0]} & d_{[1,1]} & d_{[1,2]} & \dots & d_{[1,x-3]} & d_{[1,x-2]} & d_{[1,x-1]}] \\
\text{Y-Dimension} & [\,\dots & & & \dots & & &] \\
 & [d_{[y-2,0]} & d_{[y-1,1]} & d_{[y-2,2]} & \dots & d_{[y-2,x-3]} & d_{[y-2,x-2]} & d_{[y-2,x-1]}] \\
 & [d_{[y-1,0]} & d_{[y-1,1]} & d_{[y-1,2]} & \dots & d_{[y-1,x-3]} & d_{[y-1,x-2]} & d_{[y-1,x-1]}] \\
\hline
 & \multicolumn{7}{c}{\text{X-Dimension}}
\end{array}
$$

Figure 8: **Disparity based Input Dataset**

The step for SOM initialisation includes the definition of map size and initialisation of reference vectors. The dimensions of the reference vectors must match the dimensions of the input data set (y, x, d). Continuing the iterative learning process in the form of a training, we obtain a SOM which might be used for further segmentation and detection of object areas. The training parameters and functions consist of the iteration number, initial learning rate and neighbourhood radius, distance and neighbourhood function as well as the function that defines current learning rate and current neighbourhood radius at time t (Eq. 1, Eq. 2, Eq. 3 and Eq. 4).

The vectors of the input data set can be processed sequentially or in a randomly selected order during training. In case of Fig. 9 we can see the sequentially training result.

A meaningful representation of the disparity map can be achieved by maximising distances between the reference vectors (Fig. 10).

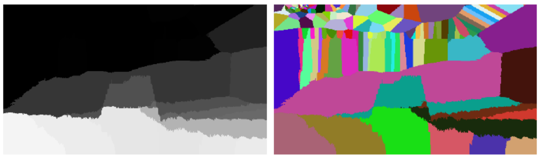
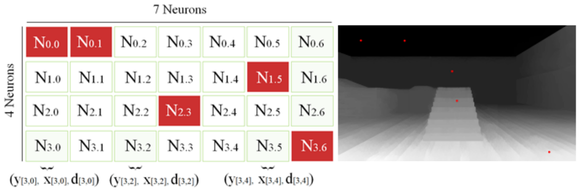


Figure 9: **Sequential Training**



N0.0	N0.1	N0.2	N0.3	N0.4	N0.5	N0.6
N1.0	N1.1	N1.2	N1.3	N1.4	N1.5	N1.6
N2.0	N2.1	N2.2	N2.3	N2.4	N2.5	N2.6
N3.0	N3.1	N3.2	N3.3	N3.4	N3.5	N3.6

Figure 10: **Selected Reference Vectors in a Disparity Map:** The red dots in the disparity map show the reference vectors. The dimension of the reference vectors consists of [4 × 7] neurons represented in a data set of (y,x,d).

1b: Disparity Map Segmentation The number of neurons within the SOM specifies the number of segments in the segmented disparity map. The following steps must occur for each pixel in the segmentation (Fig. 11):

Euclidean distance calculation
between pixels and reference vectors
⇓
Determine minimal reference vector
with smallest distance to the pixel
⇓
Pixel to neuron allocation

Figure 11: **Steps for SOM based Segmentation**

All pixels assigned to neurons represent a segment of the disparity map (see Fig. 12 and Fig. 6 in 1c).

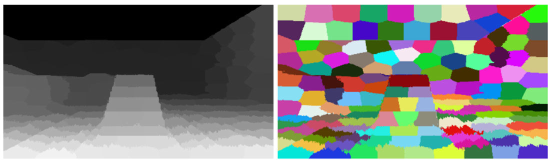


Figure 12: **Segmented Disparity Map:** The left image shows the result of a grey segmented disparity map and right image represents a colour segmented disparity map.

The grey value of the segments is based on the disparity of the associated reference vectors. The segments on the right side of Fig. 12 are coloured with randomly chosen colour values for the visible area delimitation.

The neuron count of SOM is equivalent to the segment count of a disparity map. As Fig. 13 shows, dimension expansion increases the precision of the segmented disparity map.

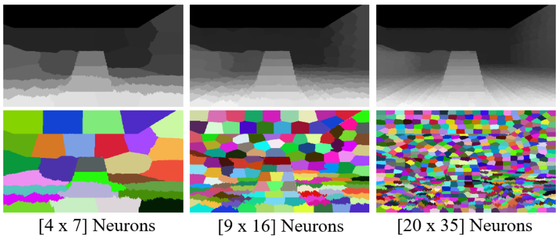


Figure 13: **Neuron-based Dimension Fitting of the Segmented Disparity Maps**

The dimension rise effects the direct growth of running time. Therefore, the requirements must be balanced during the optimisation process.

1c: Disparity based Object Area Recognition The purpose of our object recognition is to identify relevant disparity sections for further data processing. The following pipeline is to be used for this purpose (Fig. 14):

Presence check of object areas
⇓
Identification of object area and segment
allocated neurons
⇓
Associated pixel identification of
the selected object areas
⇓
Reduce the data volume
of the disparity map

Figure 14: **Object Area Identification for processing based Data Reduction**

An object area can be determined by $k \geq 1$ segments. A strong disparity difference is present when the segment designating neurons are not in close proximity to each other within the SOM. Similar neurons are determined in the neuron set for the object area identification. In case a disturbed neuron occurs between the surrounding neurons, the similar neuron cannot be part of the object area. Fig. 15 denotes the distance influence of similarly identified neurons on the segmented area regions.

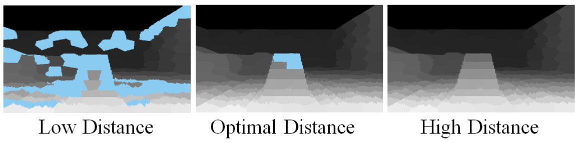


Figure 15: **Neuron Distances dependent Area Recognition**

Disparity based Data Reduction The data volume of a disparity map can be reduced by considering individual object sections. The object selection and the associated pixel reduction leads to a direct runtime reduction. A reasonable tolerance must be implemented, as not all pixels within a captured object are necessarily taken into account Fig. 16.

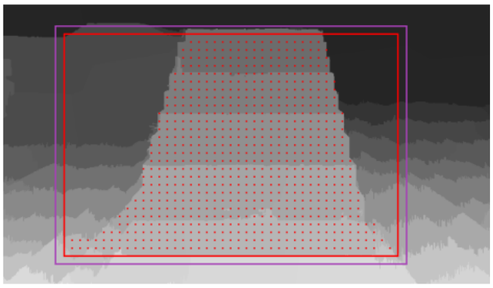


Figure 16: **Trimmed Object Area with and without Tolerance:** The red frame shows the tolerance-free neuron area. The purple frame contains a tolerance outside the stair contour.

4 EVALUATION

Our evaluation employs the SOM algorithm in several measurements to synthetically generated disparity maps. The performance analysis considers designs for runtime-optimised and non-runtime-optimised application cases. The used stair and garden images (Fig. 17 and Fig. 18) are synthetic data from Unreal Engine 4.

Quality Results We examine the number of executions necessary for complete identification of the object areas. Tab. 1 shows that the object areas are not fully identified in every execution.

	100% of Area	**95% of Area**
NOR_{Stair}	91	91
OR_{Stair}	84	86
NOR_{Garden}	79	91
OR_{Garden}	70	87

Table 1: **Iterations of SOM Algorithm:** Number of exports with compared 100% and 95% recognition of the object areas in 100 executions. The comparison is between the optimised runtime (OR) and the non-optimised runtime (NOR).

The pre-existence of an object area is incorrectly not re-established. In addition, only a partial area of the identified object is recognised. We examined how many executions had at least 95 % of the object areas identified. In areas with many objects and textures, the last 5% could not be fully recognised. Further optimisation of the SOM algorithm can increase the identification rate.

Data Reduction Considering the data reduction, we examine the percentage influence of data volumes with complete identification on average, minimum and maximum reduction.

Tab. 2 shows that the data quantity of the staircase and garden disparity maps is minimally reduced or even increased in the worst case. The unique pink segment in Fig. 17 shows incorrectly detected areas of other objects. In addition, the data reduction can be lower due to overlaps of relevant area sections (Fig. 18).

	$x_{\min}[\%]$	$\bar{x}[\%]$	$x_{\max}[\%]$	$x_{\text{opt}}[\%]$
NOR_{Stair}	12.56	79.13	86.35	90.04
OR_{Stair}	16.98	75.91	88.11	90.04
NOR_{Garden}	-6.94	39.42	47.81	56.55
OR_{Garden}	7.11	31.26	47.54	56.55

Table 2: **Data Reduction:** Results of the data reduced disparity maps Fig. 17 and Fig. 18 compared between the optimised runtime (OR) as well as the non-optimised runtime (NOR).

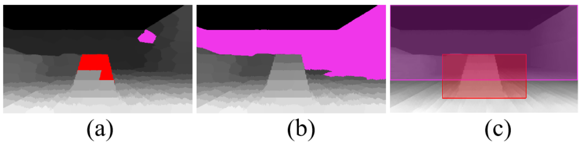


Figure 17: **Segmented Object Area of Stair Image**: Incorrectly identified object area and overlapping object areas. The figure indicates (a) unique segments, (b) faulty object areas and (c) relevant extracts.

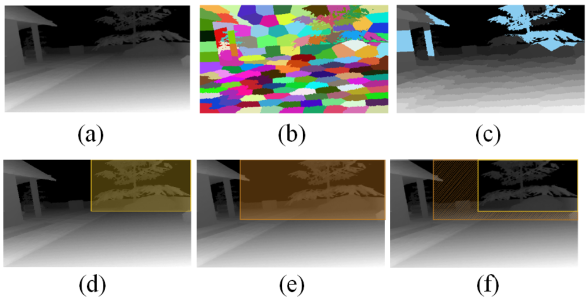


Figure 18: **Identified Object Area of Garden Image**: Description of the large-scale identified object area, which is structured into (a) original disparity map, (b) SOM map, (c) single object segmentation, (d) correct object area, (e) identified object area and (f) excess object area.

Runtime Behaviour The analysis of algorithmic stability behaviour is based on differences in runtime with repeated execution and a constant number of pixels.

A gradual increase from 9072 to 910080 pixels processed on a disparity map involves a maximum linear increase in runtime. Therefore, the number of pixels should be kept as low as possible with adequate result quality. It is noticeable that there is an almost constant runtime behaviour in the recognition of the object areas. Despite the neuron associated object areas, the number of pixels does not correlate with the number of neurons.

Stability Behaviour In the stability analysis, we set the processing pixel count to the runtime-optimised value of 9072. Our measurements show top and bottom outliers in the detected object areas. The bottom outliers indicate the runs in which no or only very small object areas were detected. The top outliers represent the runs of too large detected object areas. Top and bottom influence leads to short or long runtimes in terms of reduced data volume.

5 CONCLUSION AND FUTURE WORK

In this paper we present a concept for efficient data reduction and visual optimisation of 3D reconstutions by using SOM modified disparity maps. Our motivation is based on the challenges of real-time compatibility and 3D reconstructed quality improvements of visible scattering, distortion, noise and offsets effects. Trained Kohonen networks in form of SOM allow the detection, segmentation and further processed 3D reconstruction of protruding objects from the modified disparity map.

The amount of disparity data can be reduced by the optimised SOM application. Runtime dependency on kohonen training phase requires an optimisation of the SOM algorithm. The direct proportionality between the pixel numbers of a disparity map and the obtained results has effects on processing time. The SOM algorithm proves to be stable with different runtimes and randomness of initial reference vectors and training sequences. Due to the significant influence of randomness, different results can occur during data reduction. One advantage that should be exploited by using SOM is the arrangement of neurons. Neural arrays are suitable for the closer examination of identified object areas. However, a runtime advantage can only be achieved to a limited extent by ordering a SOM. The developed algorithm is particularly suitable for the requirements of a data reduction algorithm.

In our future work we will investigate random dropouts by using non-random sequences of qualitative and constant training results. Additional runtime optimisation will be achieved by parallelising the algorithm at various points. Computational operations in the creation of input data set, segmentation, object recognition, as well as data reduction can be distributed over several processes. We will analyse whether a divide and conquer strategy can be applied to SOM algorithm. For this purpose, the disparity map could be divided into several smaller disparity maps. Protruding detected objects could be reassembled before the data reduction. Further, we will examine the influence of Yig or YCbCr colour models. The recognition features due to striking colours of the Yig or YCbCr could lead to a reduced resolution of disparity map or neuron numbers. Future outsourcing of computational operations to the GPU could prove decently efficient in disparity map segmentation and training-conditional runtime optimisation. The simultaneous training of several pixels could also lead to promising runtimes. Finally, we will apply the SOM algorithm to real world based depth images.

6 ACKNOWLEDGEMENT

We thank Clara Schönberger for her support during concept development and data collection. We also thank Thomas Odaker, Elisabeth Mayer, Fabio Genz and Daniel Kolb who supported this work with helpful discussions and feedback.

7 REFERENCES

[Arb11] Arbelaez P., Maire M., Fowlkes C., Malik J., Contour Detection and Hierarchical Image Segmentation. 2011 IEEE Transactions on Pattern Analysis and Machine Intelligence, https://doi.org/10.1109/TPAMI.2010.161

[Cai16] Cai Z., Fan Q., Feris R., A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection., 2016 Computer Vision - ECCV 2016, https://doi.org/10.48550/arXiv.1607.07155

[Bor19] Borji A., Hou Q., Jiang H., Salient object detection: A survey. 2019 Computational Visual Media 5, https://doi.org/10.1007/s41095-019-0149-9

[Fri22] FRITZ LABS INCORPORATED, Object Detection Guide. 2022 Fritz, https://www.fritz.ai/object-detection/., last visited April 23th 2022

[Hof19] Hofmann C., Particke F., Hiller M., Thielecke J., Object Detection, Classification and Localization by Infrastructural Stereo Cameras. 2019 Proceedings of the 14th International Joint Conference on Computer Vision, https://doi.org/10.3390/electronics9020210

[Jai19] Jain D., Superpixels and SLIC. 2019 Medium, https://darshita1405.medium.com/superpixels-and-slic-6b2d8a6e4f08, last visited April 23th 2022

[Ju14] Ju R., Ge L., Geng W., Ren T., Wu G., Depth saliency based on anisotropic center-surround difference. 2014 IEEE International Conference on Image Processing (ICIP), https://doi.org/10.1109/ICIP.2014.7025222

[Kah19] Khazri A., Self Organizing Maps. 2019 IEEE Towards Data Science, https://towardsdatascience.com/self-organizing-maps-1b7d2a84e065, last visited April 23th 2022

[Kau14] Kaur D., Kaur, Y., Various image segmentation techniques: a review. 2014 International Journal of Computer Science and Mobile Computing 3, https://doi.org/10.1016/0031-3203(93)90135-J

[Koh01] Kohonen T., Self-Organizing Maps. 2001 Springer, https://doi.org/10.1007/978-3-642-56927-2

[Kri19] Krigan I., Introduction to Object Detection and Region Proposals. 2019 Brillio Data Science, https://medium.com/brillio-data-science/object-detection-part-1-introduction-to-object-detection-and-region-proposals-68f6624c98f5, last visited April 23th 2022

[Mau15] Maurer M., Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte. 2015 Springer, https://doi.org/10.1007/978-3-662-45854-9

[Mue18] Müller J., Fregin A., Dietmayer K., Disparity Sliding Window: Object Proposals from Disparity Images. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), https://doi.org/10.1109/IROS.2018.8593390

[Pon20] Pon A., Ku J., Li C., Waslander S., Object-Centric Stereo Matching for 3D Object Detection. 2020 IEEE International Conference on Robotics and Automation (ICRA), https://doi.org/10.1109/ICRA40945.2020.9196660

[Red18] Redmon J., Farhadi A., YOLOv3: An Incremental Improvement. 2018 Computer Vision and Pattern Recognition, https://doi.org/10.48550/arXiv.1804.02767

[Ren17] Ren S., He K., Girshick R., Sun J., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2017 IEEE transactions on pattern analysis and machine intelligence, https://doi.org/10.1109/TPAMI.2016.2577031

[Poz19] Pozo D., Jaramillo K., Ponce D., Torres A., Morales L., 3D reconstruction technologies for using in dangerous environments with lack of light: a comparative analysis. 2019 RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao, https://www.researchgate.net/publication/334836519

[Vac09] Skala V., Cross-talk measurement for 3D dispalays. 2009 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, https://doi.org/10.1109/3DTV.2009.5069676

[Vet18] Vettigli G., MiniSom: minimalistic and NumPy-based implementation of the Self Organizing Map. https://github.com/JustGlowing/minisom/, last visited April 23th 2022

Rate-Distortion Optimized Quantization in Motion JPEG

Tomasz Grajek
Institute of Multimedia Telecommunications
Poznań University of Technology
60-965, Poznań, Poland
tomasz.grajek@put.poznan.pl

Jakub Stankowski
Institute of Multimedia Telecommunications
Poznań University of Technology
60-965, Poznań, Poland
jakub.stankowski@put.poznan.pl

ABSTRACT

Rate-Distortion Optimized Quantization (RDOQ) is an encoding optimization technique that may be applied to any transform-based compression technique preserving bitstream compliance with the standard. In the paper, the application of the RDOQ to Motion JPEG is described and evaluated. The proposed solution includes block-level optimization with picture-level Lagrange multiplier estimation. Performed evaluation results in higher compression ratios as compared to typical Motion JPEG.

Keywords
Rate-Distortion Optimized Quantization (RDOQ), Motion JPEG, video compression, video streaming.

1. INTRODUCTION

In 2023 JPEG [ITU21] will celebrate its 30th anniversary. Despite the fact that many more efficient techniques were developed, just to mention a few: JPEG 2000 [ISO19], JPEG-XR [ITU20], JPEG-XL [ISO22], HEVC Intra (HEIF) [ISO17], AV1 Intra (AVIF) [Con21], it is still very popular and commonly used in many applications and products [Bor21, Hud18, W3T22].

Based on JPEG, Motion JEPG [RFC98] (sometimes abbreviated as MJPEG) was developed to handle video sequences. The general idea is to encode images from the sequence independently using traditional JPEG. The Motion JPEG is very commonly used in non-linear video editing systems allowing native random access to any frame. Motion JPEG is not so efficient as modern video compression techniques, e.g. HEVC [ISO20a] or VVC [ISO20b], however, it is by far a less complex and resources-demanding solution. Therefore, Motion JPEG is still an attractive technique.

The compression efficiency of a given technology strongly depends on the rate control of the encoders [Bea19, Ric02]. Therefore new control techniques may be added to encoders as long as conformance with the standard is preserved.

In literature, many different approaches to optimizing quantization for image or video encoders are described [Cro97, He14, Luo21, Ram94, Saf19, Wan22, Xu18]. Most of the solutions for JPEG assume that new quantization and Huffman tables have to be defined and transmitted to the decoder. However, such solutions cannot be applied to Motion JPEG as according to RFC 2435 [RFC98] does not allow transmitting custom quantization and Huffman tables.

Rate-distortion optimized quantization (RDOQ) [Kar08] is a non-normative technique allowing for compression efficiency increase by additional analysis of quantized transform coefficients before entropy encoding.

The authors of this paper in [Sta15] presented an extensive analysis of RDOQ application to HEVC.

The paper presents the adaptation and implementation of the RDOQ technique from [Sta15] in Motion JPEG.

2. Motion JPEG

Motion JPEG is a technique that uses a subset of JPEG [ITU21] to compress video sequences by independent coding of consecutive frames. JPEG [Pen93] is a very simple and straightforward approach utilizing transform coding (see Fig. 1).

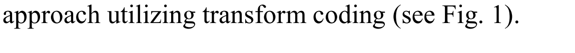


Figure 1. JPEG compression scheme. O - input data, T - transform coefficients, Q - quantized transform coefficients, S - quantized transform coefficients after zig-zag scanning, b - bitstream.

https://www.doi.org/10.24132/CSRN.3201.29

An encoded image is divided into 8x8 blocks. Then each block is independently processed in raster scan order. Firstly discrete cosine transformation is performed. Then quantization to transformed coefficients is applied. The strength of the quantization is adjusted by the scaling factor. The quantized transform coefficients are rearranged from an 8x8 matrix into a 64-element vector using a zig-zag scan algorithm. Finally, entropy encoding is performed. The entropy coding is mostly a Huffman coding, although an arithmetic one is also available. Prediction is applied only to DC coefficients. The AC coefficients contain direct image data. It should be stressed here, that in modern techniques a prediction error is transformed and quantized. Because there is no prediction in JPEG (besides prediction of DC coefficients) a huge number of non-zero quantized transform coefficients have to be entropy encoded. Therefore, bitstream (or file) contains almost only quantized transform coefficients.

Quantization in JPEG is defined as a simple division (uniform, scalar quantization) and rounding [Mia99, Pen93]. It is a very fast solution but may be suboptimal.

A huge number of non-zero quantized transform coefficients, especially for weak quantization, leave a space for further optimization, for example by using rate-distortion optimized quantization. As a result compression efficiency may be improved preserving conformance with the standard.

3. PROPOSED TECHNIQUE

In the paper, we propose to apply simplified rate-distortion optimized quantization based on that described in [Sta15] to Motion JPEG. The general scheme of the improved JPEG encoder is presented in Fig. 2.

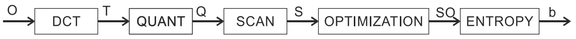


Figure 2. JPEG with RDOQ compression scheme. O - input data, T - transform coefficients, Q - quantized transform coefficients, S - quantized transform coefficients after zig-zag scanning, SO - quantized transform coefficients after RDOQ, b - bitstream.

The details of the proposed RDOQ are the following. After zig-zag scanning of a block of quantized transform coefficients, a vector is derived with DC coefficient at the beginning of the vector and AC coefficients with the highest frequency at the end. Only non-zero transform coefficients are analyzed starting from the last non-zero quantized transform coefficient in a given vector. For each non-zero coefficient, up to four cases are considered, namely: leaving the coefficient without any change, increasing the value of this quantized transform coefficient by 1, decreasing its value by 1, and finally setting it to 0. For each case, the RD cost is calculated using the Lagrange multipliers (λ) approach [Kar08] and the case with the lowest RD cost is chosen as the best one (see Fig. 3).

$$RD\_cost\,(S,\,n) = SSD(S,\,n) + \lambda \cdot R(S,\,n),$$

where:
n – a quantized transform coefficient identifier,
S – a value of the quantized transform coefficient n,
$RD\_cost(S,\,n)$ – the cost of quantization coefficient n to value S,
$SSD(S,\,n)$ – a sum of squared differences between the original and reconstructed block of samples,
$R(S,\,n)$ – number of bits needed to encode block with coefficient n quantized to value S,
λ – Lagrange multiplier.

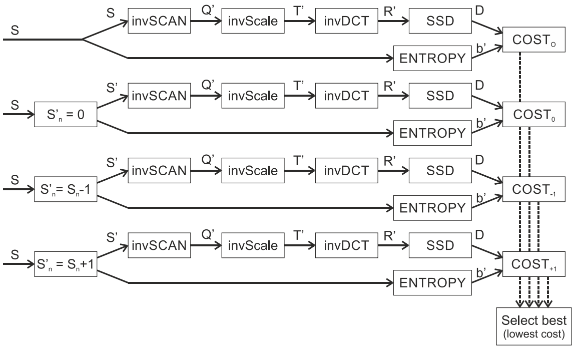


Figure 3. Flowchart of the RDOQ. *S* - quantized transform coefficients after zig-zag scanning, S'_n - modified quantized transform coefficients.

The described procedure is repeated for the rest of the non-zero quantized transform coefficients in the vector towards the beginning of the vector. DC coefficient is excluded from the analysis as it is predictively encoded between blocks. The reason behind this decision is to allow unrestricted parallelization of RDOQ, as no other data is encoded with dependencies between blocks.

Lagrange multiplier calculation

Application of the Lagrange multiplier optimization requires a calculation of mentioned λ multiplier. In the analyzed case, the λ multiplier expresses how to balance distortions introduced by the given technique and the number of bits needed to encode quantized transform coefficients and establishes the operational tradeoff for the encoder rate-distortion optimization stage. Lower λ values encourage the encoder to prefer bitrate reduction over quality and higher λ values strengthen the importance of quality over bitrate.

In typical applications, the JPEG encoder does not use any optimization as it works with constant quality (constant scaling factor), very often set by the

https://www.doi.org/10.24132/CSRN.3201.29

user. The authors are not aware of any general formula to calculate the λ multiplier for JPEG similarly as it was derived for HEVC or VVC. Therefore, the λ multiplier is calculated locally by encoding the given picture with Q and Q-1 values. After each encoding information about SSD and bitrate (R) are gathered. Finally, the λ multiplier is derived as:

$$\lambda = -\frac{SSD_Q - SSD_{Q-1}}{R_Q - R_{Q-1}},$$

where:

SSD_x – the sum of squared differences between the original and reconstructed image for a given scale factor setting (Q or Q-1),
R_x – number of bits needed to encode coefficients from the whole image for a given scale factor setting (Q or Q-1).

The local calculation of λ is performed at the picture level. This allows for the adaptation of the λ parameter to the characteristic of every encoded image.

4. IMPLEMENTATION

In order to evaluate the proposed technique, it was necessary to prepare and test its implementation. Authors considered using one of the existing JPEG/MJPEG implementations, for example, libjpeg-turbo [LJT22] which can be considered the fastest software-based implementation of JPEG encoder and decoder. Unfortunately, the complexity of highly optimized implementations and some design choices made by its authors made the usage of existing JPEG encoders very burdensome. Therefore, the authors decided to develop their own implementation of the encoder.

The implementation created for RDOQ-related experiments is restricted to a subset of features used in Motion JPEG and is highly modular with each processing stage (i.e. transform, quantization, scan, entropy) clearly separated from each other. Some algorithmic techniques used in modern JPEG implementations (like fast integer-based DCT transform and fast reciprocal-based integer quantization) were included to match the behaviour of production-quality JPEG encoders. In order to speed up experiments, some of the encoder stages (transform and quantization) were implemented using vector instructions (SSE4.1, AVX2) [Lom11].

As described in Section 3, the RDOQ optimization step requires the calculation of block distortion (SSD) and the number of bits required to represent the currently processed block. SSD calculation is quite straightforward as it corresponds to decoding of JPEG compressed block of pixels, but with entropy decoding omitted. Therefore, the SSD calculation steps (inverse scan, inverse scaling, and inverse transform) are inherited from the JPEG decoder and followed by distortion calculation.

In the case of calculation of the number of bits required to represent the currently processed block, one can use an already existing Huffman encoder. Nonetheless, this is a very inefficient approach. Since the Huffman encoder in JPEG is designed to create a valid bitstream, most of the work done by the Huffman encoder is useless when this block is used to calculate the number of bits only. To avoid unnecessary computations, the authors developed the so-called Huffman counter module. The Huffman counter is a simplified version of the Huffman encoder and it is responsible for fast and accurate calculation of the number of bits required to represent a block of transform coefficients.

The developed implementation is designed to be easily parallelized, although in this paper authors concentrated on the proposed algorithm and compression efficiency to avoid distracting the reader with parallelization-related details.

As mentioned in section 1, the authors concentrated on the Motion JPEG use case, therefore some of the coding techniques and coding tools are out of the scope of this paper. The optimization of Huffman tables is not studied nor implemented since RFC2435 [RFC98] forces the encoder to use default Huffman tables. Similarly, the JPEG standard [ITU21] includes the possibility to use an arithmetic encoder instead of a Huffman one. However, due to patent issues, the adoption of arithmetic encoding in JPEG is negligible and it is almost impossible to find supporting implementation.

It should be emphasized that all developed implementations of MJPEG encoder (both with and without RDOQ) are fully conformant to JPEG standard [ITU21] which means they produce correct and decodable bitstreams. This bitstream can be transmitted as described in RFC2435 [RFC98] but also could be embedded into JFIF files [ITU11].

5. EVALUATION

Methodology

As mentioned before, the authors of the paper concentrate on the Motion JPEG use case. Therefore, to evaluate the compression efficiency of the proposed solution a wide range of video sequences recommended by MPEG Committee experts of the International Organization for Standardization was used. These video test sequences are commonly used for video compression techniques development and evaluation as they cover a wide range of content characteristics. Experiments were conducted on the following 16 sequences with 1920x1080 resolution: *BQTerrace*, *BasketballDrive*, *Cactus*, *Kimono1*, *ParkScene*, *blue_sky*, *pedestrian_area*, *riverbed*, *rush_hour*, *station2*, *sunflower*, *tennis*, *toys_and_calendar*, *tractor*, *vintage_car*, *walking_couple* (see Fig.4). For evaluation, the first 100 frames from each sequence were used.

Figure 4. Single images selected from video test sequences used in the experiments in order from top-left: *tractor*, *sunflower*, *station2*, *soccer*, *rushhour*, *riverbed*, *pedestrian_area*, *ice*, *harbour*, *crew*, *city*, *bluesky*.

The experimental evaluation was performed for all quality point values (Q) in the range 1-100. It is worth noting that in JPEG nomenclature the Q parameter means “quality”, while in modern techniques (e.g. AVC, HEVC, or VVC) the q or rather qp parameter corresponds to the quantization step size. Therefore, Q=1 means the strongest quantization, so the lowest possible quality and the highest compression ratio; while Q=100 means the weakest quantization thus the highest possible quality and the lowest compression ratio.

During evaluation three coding scenarios were considered:

- 1st – using a base MJPEG encoder without RDOQ;
- 2nd – using an MJPEG encoder with RDOQ enabled only for luminance component;
- 3rd – using an MJPEG encoder with RDOQ enabled for all available components (luminance and chrominances).

Both 2nd and 3rd scenarios were compared against 1st (base). According to [ISO20c], results are presented as Bjøntegaard-Delta bitrate and Bjøntegaard-Delta PSNR. Results are presented for luminance component only (ΔPSNR-Y and ΔBitrate-Y) as well as for all components averaged (ΔPSNR-YCbCr and ΔBitrate-YCbCr) with 6:1:1 weights for Y:Cb:Cr respectively.

Each test sequence was encoded, decoded and PSNR values were calculated. Taking into account that experiments were performed for 16 test sequences, 100 quality points, and 3 scenarios, the total number of test points is equal to 4800 which corresponds to 480000 processed pictures.

Results

In Figures 5 and 6 examples of two test sequences were presented for a useful range of bitrates that guarantee very good quality (above 40dB) of reconstructed data.

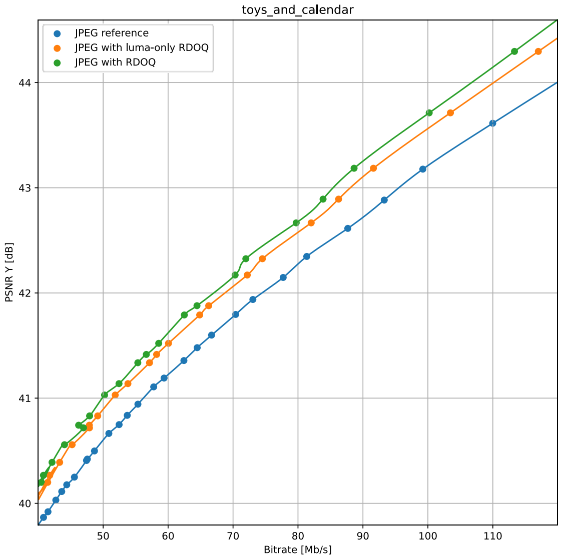


Figure 5. Examples of Rate-distortion curves for the *toys_and_calendar* sequence.

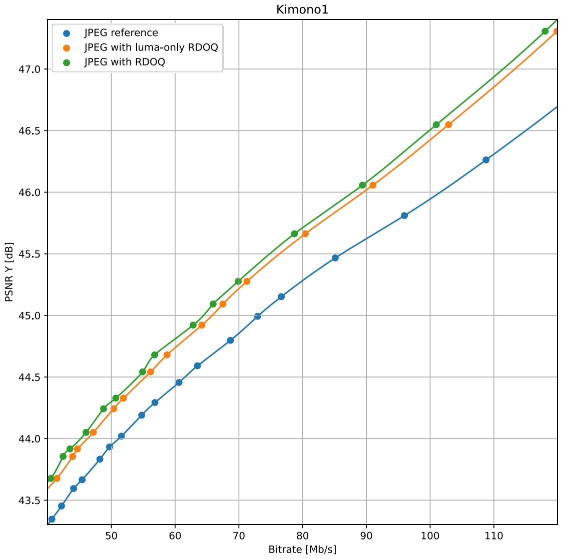


Figure 6. Examples of Rate-distortion curves for the *Kimono1* sequence.

In mentioned figures three R-D curves are presented corresponding to three considered scenarios: using a base MJPEG encoder without RDOQ (blue line), using an MJPEG encoder with RDOQ enabled only for the luminance component (orange line), and using an MJPEG encoder with RDOQ enabled for all available components (green line). It is clear that enabling RDOQ results in compression efficiency. The R-D curves for other test sequences are very similar.

More detailed data are gathered in Tables 1 and 2 where Bjøntegaard-Delta bitrate and Bjøntegaard-Delta PSNR measures are presented for all test sequences. To calculate them four *Q* values were selected i.e. 70, 75, 80, and 85 that correspond to a useful range of bitrates. In Table 1 results for the 2nd scenario are presented whereas in Table 2 results for the 3rd scenario are gathered. Depending on the characteristic of video content 2nd scenario offers from 0.34 to 0.86 dB gain in quality, 0.42dB on average for the same bitrate. On the other hand, scenario 2 results in 6.2 to 12.1% (8.3% on average) bitrate reduction preserving the same quality of reconstructed data. Scenario 3 results in quality gain from 0.40 to 1.10dB, 0.54dB on average for the same bitrate, and from 8.4 to 13.0% (10.38% on average) bitrate reduction.

Sequence	ΔPSNR Y [dB]	ΔPSNR YCbCr [dB]	ΔBitrate Y [%]	ΔBitrate YCbCr [%]
BQTerrace	0.859	0.735	-6.52	-6.80
BasketballDrive	0.480	0.446	-8.52	-8.69
Cactus	0.397	0.368	-7.26	-7.40
Kimono1	0.341	0.307	-10.46	-9.70
ParkScene	0.421	0.393	-6.71	-6.76
blue_sky	0.536	0.561	-9.39	-9.15
pedestrian_area	0.412	0.391	-9.22	-8.97
riverbed	0.348	0.315	-6.54	-6.22
rush_hour	0.390	0.361	-12.16	-11.59
station2	0.383	0.357	-8.55	-8.21
sunflower	0.411	0.393	-10.97	-9.94
tennis	0.384	0.354	-9.38	-9.22
toys_and_calendar	0.344	0.318	-7.85	-7.69
tractor	0.366	0.347	-6.17	-5.82
vintage_car	0.376	0.318	-6.58	-6.62
walking_couple	0.352	0.305	-7.08	-7.07
Average	**0.425**	**0.392**	**-8.337**	**-8.116**

Table 1. Experimental results for luma-only RDOQ (2nd scenario) presented as Bjøntegaard Delta (BD) for bitrate and PSNR.

Sequence	ΔPSNR Y [dB]	ΔPSNR YCbCr [dB]	ΔBitrate Y [%]	ΔBitrate YCbCr [%]
BQTerrace	1.101	0.914	-8.32	-8.37
BasketballDrive	0.604	0.543	-10.53	-10.36
Cactus	0.534	0.471	-9.64	-9.36
Kimono1	0.405	0.371	-12.42	-11.56
ParkScene	0.582	0.510	-9.14	-8.68
blue_sky	0.683	0.656	-11.82	-10.67
pedestrian_area	0.491	0.463	-10.88	-10.51
riverbed	0.441	0.387	-8.25	-7.61
rush_hour	0.443	0.428	-13.72	-13.53
station2	0.473	0.427	-10.50	-9.76
sunflower	0.488	0.462	-12.98	-11.61
tennis	0.476	0.435	-11.48	-11.15
toys_and_calendar	0.467	0.418	-10.50	-9.95
tractor	0.503	0.447	-8.40	-7.46
vintage_car	0.488	0.413	-8.47	-8.49
walking_couple	0.459	0.396	-9.16	-9.07
Average	**0.540**	**0.484**	**-10.387**	**-9.883**

Table 2. Experimental results for RDOQ performed on both luma and chroma (3rd scenario) presented as Bjøntegaard Delta (BD) for bitrate and PSNR.

6. CONCLUSIONS

In the paper adaptation and implementation of the RDOQ technique to Motion JPEG was presented. The proposed solution includes block-level optimization with picture-level Lagrange multiplier estimation. Moreover, some possible ways to parallelize the RDOQ in Motion JPEG were highlighted. Extensive experiments with a wide range of test video sequences proved that this simple technique offers about 10% bitrate reduction when preserving the same quality of reconstructed videos.

7. ACKNOWLEDGMENTS

The research was supported by the Ministry of Education and Science of the Republic of Poland under the subvention for research SBAD.

8. REFERENCES

[Bea18] Beach, A., Owen, A., “Video Compression Handbook”, 2nd Edition, Peachpit Press, 2018.

[Bor21] Borkowski, D., Janczak-Borkowska, K., “Reduction of JPEG Artifacts using BSDEs”, Computer Science Research Notes, vol. 3101, 29th WSCG, Plzen, Czech Republic, 2021

[Con21] Concolato, C., Klemets, A., (eds.), AV1 Image File Format (AVIF), Alliance for Open

Media, https://aomediacodec.github.io/av1-avif/, 2021.

[Cro97] Crouse, M., Ramchandran, K., "Joint Thresholding and Quantizer Selection for Transform Image Coding: Entropy-Constrained Analysis and Applications to Baseline JPEG", IEEE Transactions on Image Processing, vol. 6, no. 2, pp. 285-297, 1997.

[He14] He, D., Wang, J., "Rate distortion optimized quantization based on weighted mean squared error for lossy image coding", 2014 IEEE International Conference on Image Processing (ICIP), pp. 5606-5610, Paris, France, 2014.

[Hud18] Hudson, G., Léger, A., Niss, B., Sebestyén, I., Vaaben, J., "JPEG-1 standard 25 years: past, present, and future reasons for a success", Journal of Electronic Imaging, vol. 27, no. 4, 2018.

[ISO17] ISO/IEC 23008-12, "Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format", 2017.

[ISO19] ISO/IEC IS 15444, "Information technology - JPEG 2000 image coding system", Edition 4, 2019 also ITU-T Rec. T.800, Edition 06/2019. 2019

[ISO20a] ISO/IEC 23008-2, "MPEG-H Part 2, High-Efficiency Video Coding", Edition 4, 2020, also: ITU-T Rec. H.265 Edition 7.0, 2019

[ISO20b] ISO/IEC 23090-3, "MPEG-I Part 3, Versatile Video Coding", Edition 1, 2020 also: ITU-T Rec. H.266 Edition 1.0, 2020

[ISO20c] ISO/IEC TR 23002-8, "Working Practices Using Objective Metrics for Evaluation of Video Coding Efficiency Experiments", 2020.

[ISO22] ISO/IEC 18181:2022, "Information technology - JPEG XL image coding system", 2022.

[ITU11] Recommendation ITU-T T.871, "Information technology – Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF)", International Telecommunication Union - Standardization Sector (ITU-T), 2011.

[ITU20] Recommendation T.832, "Information technology - JPEG XR image coding system", International Telecommunication Union - Standardization Sector (ITU-T), 2020.

[ITU21] Recommendation H.222.0, "Information technology - Generic coding of moving pictures and associated audio information: Systems", International Telecommunication Union - Standardization Sector (ITU-T), 2021

[Kar08] Karczewicz, M., Ye, Y., Chong, I., "Rate Distortion Optimized Quantization", document ITU-T SG16 Q.6, VCEG-AH21, 2008

[LJT22] libjpeg-turbo software webpage, https://libjpeg-turbo.org/, retrieved 2022.

[Lom11] Lomont, C., "Introduction to Intel:registered: Advanced Vector Extensions", Intel White Paper, 2011.

[Luo21] Luo, X., Talebi, H., Yang, F., Elad, M., Milanfar, P., "The Rate-Distortion-Accuracy Tradeoff: JPEG Case Study", 2021 Data Compression Conference (DCC), Snowbird, USA, 2021

[Mia99] Miano, J., "Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP", Addison-Wesley Professional, 1999.

[Pen93] Pennebaker, W.B., Mitchell, J.L., ”JPEG: Still Image Data Compression Standard (Digital Multimedia Standards S), Springer, 1993.

[Ram94] Ramchandran, K., Vetterli, M., "Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility", IEEE Transactions on Image Processing, vol. 3, no. 5, pp. 700-704, 1994.

[RFC98] RFC 2435, RTP Payload Format for JPEG-compressed video, 1998.

[Ric02] Richardson, I., "Video Codec Design: Developing Image and Video Compression Systems", Wiley, 2002

[Saf19] Safonov, I.V., Kurilin, I.V., Rychagov, M.N., Tolstaya, E.V., "Fast Control of JPEG Compression Rate", In: Document Image Processing for Scanning and Printing. Signals and Communication Technology. Springer, 2019

[Sta15] Stankowski, J., Korzeniewski, C., Domański, M., Grajek, T., „Rate-distortion optimized quantization in HEVC: performance limitations", 31st Picture Coding Symposium, PCS 2015, pp. 85-89, Cairns, Australia, 2015.

[W3T22] W3Techs, Usage statistics of JPEG for websites, https://w3techs.com/technologies/details/im-jpeg, 2022

[Wan22] Wang, Q., Liu, P., Zhang, L., Cheng, F., Qiu, J., Zhang, X., "Rate–distortion optimal evolutionary algorithm for JPEG quantization with multiple rates", Knowledge-Based Systems, vol. 244, 108500, 2022.

[Xu18] Xu, M., Canh, T.N., Jeon, B., "Rate-Distortion Optimized Quantization: A Deep Learning Approach", IEEE High Performance Extreme Computing Conference, Waltham, USA, 2018.

Real-time CPU-based View Synthesis for Omnidirectional Video

Jakub Stankowski
Institute of Multimedia Telecommunications
Poznań University of Technology
Polanka 3
61-131 Poznań, Poland
jakub.stankowski@put.poznan.pl

Adrian Dziembowski
Institute of Multimedia Telecommunications
Poznań University of Technology
Polanka 3
61-131 Poznań, Poland
adrian.dziembowski@put.poznan.pl

ABSTRACT

In this paper, the authors describe the real-time CPU-based implementation of the virtual view synthesis algorithm for high-resolution omnidirectional content. The proposed method allows a user of the immersive video virtually navigating within the scene captured by a multiview system comprised of 360-degree or 180-degree cameras. The proposed method does not require using powerful graphic cards as other state-of-the-art real-time synthesis methods. Instead, the emerge of consumer-grade multithreaded CPUs and CPU-based virtual view synthesis, allows further development of cheap, consumer immersive video systems. The proposed method was compared with the state-of-the-art view synthesis algorithm – RVS, both in terms of quality of synthesized views and computational time required for the synthesis, presenting the usefulness of the proposed method.

Keywords

Virtual view synthesis, omnidirectional video, immersive video systems, real-time video processing.

1. INTRODUCTION

The virtual view synthesis is a crucial step of processing of immersive video [Isg14], virtual navigation of free-viewpoint television systems [Tan12]. It allows a user of such kind of the video system to virtually navigate within scene captured with a multicamera system (e.g., [Goo12], [Sta18], [Zit04]) by producing artificial viewports between actual positions of the cameras [Dzi19], [Fac18].

In order to provide a proper feeling of immersion of the user into the scene, the view synthesis has to be performed in the real-time.

There are multiple state-of-the-art methods of the real-time view synthesis. However, most of them require using dedicated FPGA devices (e.g. [Aki15], [Li19]), powerful GPUs (e.g., [Non18], [Zha17]) or even VLSI devices [Hua19]. The necessity of using such devices significantly limits the possibility of developing a cheap, consumer immersive video system.

In this paper, we present a real-time synthesis method implemented on the CPU, which is much harder to efficiently implement [Dzi18], [Sta20]. Moreover, existing algorithms of the CPU-based real-time view synthesis described in [Dzi18] and [Sta20] handle only the typical, perspective views thus cannot be used in modern immersive video systems with omnidirectional, 360-degrees video.

2. VIRTUAL VIEW SYNTHESIS FOR OMNIDIRECTIONAL CONTENT

Omnidirectional vs. perspective synthesis

Regardless of the content type, the virtual view synthesis is based on reprojecting information from input views to the virtual view. However, the math behind the reprojection differs for different types of content.

For perspective content, the reprojection uses homography matrices, combining extrinsic and intrinsic parameters of input camera and virtual camera [Sta20].

https://www.doi.org/10.24132/CSRN.3201.30

For omnidirectional content, reprojection equations depend on the representation type, e.g., equirectangular projection (ERP) or cube map projection (CMP). In this paper, we deal with the ERP, as it is the most commonly used representation for omnidirectional video.

Algorithm of view synthesis

The proposed algorithm is made up from four major stages (Fig. 1): (1) depth reprojection, (2) depth merging and filtering of the depth map of the virtual view, (3) texture reprojection, and (4) inpainting of holes in the virtual view.

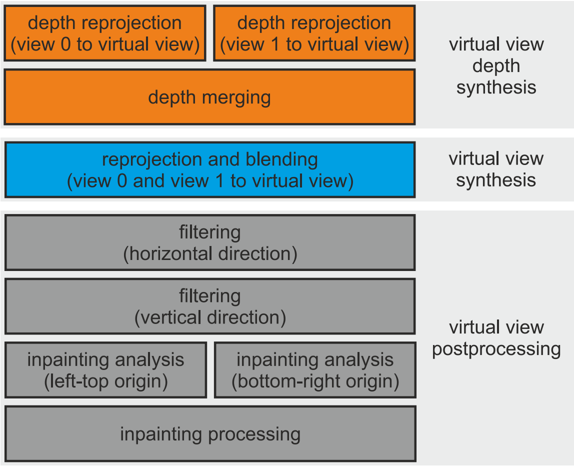

depth reprojection (view 0 to virtual view)
depth reprojection (view 1 to virtual view)
depth merging
virtual view depth synthesis
reprojection and blending (view 0 and view 1 to virtual view)
virtual view synthesis
filtering (horizontal direction)
filtering (vertical direction)
inpainting analysis (left-top origin)
inpainting analysis (bottom-right origin)
inpainting processing
virtual view postprocessing

Figure 1. Overview of described synthesis algorithm.

In order to meet the requirement of the computational time and make the algorithm real-time, only two input views are used for the synthesis. However, in the omnidirectional scenario, where each input view contains much more information than for perspective cameras, such a limitation does not significantly reduce the quality of the synthesized views.

Of course, in case of using only two views, these two views have to be carefully chosen among all available input views in order to provide the best possible quality. However, fast and efficient view selection algorithms exist (e.g. [Dzi18b] or the method used in [MPEG21b]) and can be used for this purpose.

2.1.1 Depth reprojection

At first, only depth maps are processed. Each pixel of both input depth maps is processed in the same way. Firstly, the position of the point in the 3D space is calculated:

$$\begin{aligned} X &= z_R \cdot cos(\theta_R) \cdot cos(\varphi_R) - t_X^R\,, \\ Y &= z_R \cdot sin(\theta_R) - t_Y^R\,, \\ Z &= -z_R \cdot cos(\theta_R) \cdot sin(\varphi_R) - t_Z^R\,, \end{aligned}$$

where z_R is the depth of the pixel, $[t_X^R, t_Y^R, t_Z^R]'$ is the translation vector of the input camera R, and (θ_R, φ_R) is the angular position of the pixel within input view R.

In the second step of depth reprojection, a 2D position of the pixel within virtual view (θ_R, φ_R) and its depth z_V are calculated:

$$\begin{aligned} \varphi_V &= tan^{-1}\left(-\frac{Z - t_Z^V}{X - t_X^V}\right), \\ z_V &= \sqrt{(X - t_X^V)^2 + (Y - t_Y^V)^2 + (Z - t_Z^V)^2}\,, \\ \theta_V &= sin^{-1}\left(\frac{Y - t_Y^V}{z_V}\right). \end{aligned}$$

If the virtual view is a perspective one, the projection from 3D space into the image plane is performed with multiplication by projection matrix of the virtual camera:

$$\begin{bmatrix} x_V \\ y_V \\ z_V \end{bmatrix} = \boldsymbol{P_V} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.$$

2.1.2 Depth merging and filtering

In the second stage, depth maps reprojected from both input depth maps are merged to create a single depth map of the virtual view.

For each pixel of this depth map, the merging algorithm chooses the depth candidate, which is closer to the virtual camera. If a pixel was reprojected from one input view only, no merging is necessary, and this single candidate is chosen. If no depth candidates are available (i.e., pixel was not visible in any input view), the pixel of the merged depth map will be empty.

After merging, the depth map of the virtual view is being filtered to eliminate small depth artifacts visible as single pixels surrounded by much smaller or much greater depth values. Such wrong pixels in the depth map may significantly reduce both the subjective and objective quality of the synthesized view, and are caused mostly by object discontinuities and blurred edges within input depth maps.

2.1.3 Texture reprojection

After creation of the depth map of the virtual view, texture information is reprojected. The fast reprojection of texture is performed using look-up tables, which for each pixel of the virtual view store its initial position in the input view.

The color of each pixel is calculated by averaging of its colors in both input views (if possible) or copying a color from one input view if it was not visible in another view.

2.1.4 Inpainting

After the depth and texture reprojection, the virtual view has holes – areas not visible in any input views. These areas have to be inpainted [Ber00], [Dar10].

In order to perform possibly fastest inpainting, a color of each empty pixel is set by copying of the color from one of its four neighbors (top, left, right or bottom) – the neighbor which has the farthest depth.

3. ALGORITHM IMPLEMENTATION

In order to evaluate the synthesis performance several implementations have been prepared. The basic one described as “Reference” is written without any optimizations. The second one is algorithmically optimized including reduction of time-consuming operations, buffering of pre-calculated immediate values and memory load/stores reduction.

The second set of implementations includes vectorization and parallelization by using techniques developed for previously described synthesis algorithm for perspective video [Sta20].

The vectorized implementations use AVX2 or AVX512 extensions [Dem13]. Unfortunately, due to significantly higher number of calculations and usage of more complex functions (including many trigonometric transformations) the vectorized implementation for omnidirectional view synthesis is much more difficult. Both vectorized implementation uses a specially crafted routines for $sqrt$, tan^{-1} and sin^{-1} calculation. The implementations used to calculate abovementioned functions prioritize performance over precision and can be considered as seasonable compromise between speed and distortion introduced by computation errors.

The multithreaded implementation uses previously developed Independent Projection Targets (IPT) [Sta20] in order parallelize the depth reprojection. Depth reprojection is the most compute heavy step of synthesis algorithm and its parallelization allows to significant reduction of computation time.

4. EXPERIMENTS

Test sequences

The proposed view synthesis algorithm was tested on a test set containing four miscellaneous omnidirectional test sequences (Fig. 2):

1. ClassroomVideo, 4K×2K resolution, 16 full-360° cameras [Kro18],
2. Chess, 2K×2K resolution, 10 semispherical cameras placed on the sphere [Ilo19],
3. Cyberpunk, 2K×2K resolution, 10 parallel semispherical cameras [Jeo22],
4. Hijack, 4K×2K resolution, 10 parallel cameras with angle of view 180°×90° [Dor18].

The sequences are commonly used in immersive video applications, e.g., within ISO/IEC JTC1/SC29/WG04 MPEG Video Coding group [MPEG21].

Experiment setup

In the experiment, 9 implementations of proposed virtual view synthesis method were evaluated. The results are compared with the state-of-the-art method for omnidirectional view synthesis: RVS (Reference View Synthesizer) [Fac18], commonly used by individual researchers and the ISO/IEC MPEG group [MPEG18].

The implementations differ in usage of AVX2 and AVX512 instruction sets, multi-threading (MT), and Independent Projection Targets (IPT) technique which allows to use separate buffers for each thread during depth projection.

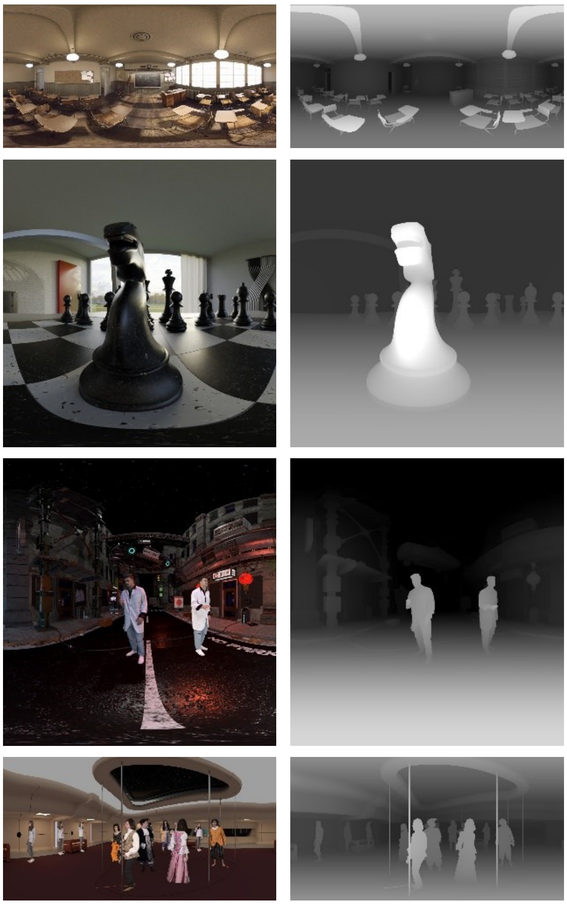


Figure 2. Input views and corresponding depth maps for (from top): ClassroomVideo, Chess, Cyberpunk, and Hijack.

CPU model	Implementation	Processing time [ms/frame]				
		DP	DM	VP	PP	Entire frame
i7-8700K	RVS	n/a	n/a	n/a	n/a	21583.300
	Reference	778.356	1.869	72.967	70.264	923.458
	Optimized	736.679	1.864	73.216	74.240	886.000
	Optimized + MT	375.859	1.889	12.184	14.664	404.597
	Optimized + MT + IPT	197.117	11.830	12.054	14.571	235.573
	Optimized + AVX2	116.948	1.846	50.350	73.066	242.210
	Optimized + AVX2 + MT	62.580	1.908	11.313	14.626	90.428
	Optimized + AVX2 + MT + IPT	38.271	13.289	11.246	15.395	78.203
R9-3900X	RVS	n/a	n/a	n/a	n/a	19718.100
	Optimized	659.097	2.145	84.709	69.443	815.396
	Optimized + MT + IRT	183.148	12.162	11.764	12.162	219.238
	Optimized + AVX2	146.462	2.239	62.278	68.305	279.284
	Optimized + AVX2 + MT + IPT	44.242	13.849	11.360	12.644	82.097
i9-7900X	Optimized + MT + IRT	233.397	6.940	13.021	11.587	264.946
	Optimized + AVX2	144.286	2.957	65.467	80.655	293.366
	Optimized + AVX2 + MT + IPT	52.648	6.907	9.187	12.392	81.135
	Optimized + AVX512	103.010	3.081	67.077	83.265	256.434
	Optimized + AVX512 + MT + IPT	36.605	6.812	9.232	9.865	62.515
	Optimized + AVX512 + MT + IPT, synthesis 4K -> 2K	37.547	1.727	3.892	4.360	47.527

Table 1. Computation time comparison of the state-of-the-art view synthesis method RVS [Fac18] and all tested implementations of proposed synthesis method on 4K×2K sequence (ClassroomVideo) Processing stages: DP – depth projection, DM – depth merging, VP – view projection, PP – postprocessing.

CPU model	Implementation	Processing time [ms/frame]				
		DP	DM	VP	PP	Entire frame
i7-8700K	RVS	n/a	n/a	n/a	n/a	11383.700
	Reference	424.173	0.849	34.664	38.501	498.187
	Optimized	379.340	0.804	32.197	39.616	451.957
	Optimized + MT	141.300	0.844	6.739	7.160	156.043
	Optimized + MT + IPT	85.406	5.365	5.965	7.976	104.712
	Optimized + AVX2	47.482	0.947	21.619	38.537	108.585
	Optimized + AVX2 + MT	27.340	0.864	5.859	6.844	40.907
	Optimized + AVX2 + MT + IPT	17.201	5.053	4.700	6.970	33.924
R9-3900X	RVS	n/a	n/a	n/a	n/a	9476.200
	Optimized	248.529	0.977	39.807	28.696	318.009
	Optimized + MT + IRT	71.964	6.039	4.979	4.871	87.853
	Optimized + AVX2	67.775	1.030	29.170	36.822	134.797
	Optimized + AVX2 + MT + IPT	16.752	6.890	5.526	6.613	35.781
i9-7900X	Optimized + MT + IRT	101.038	3.417	7.151	4.304	115.91
	Optimized + AVX2	59.672	1.223	33.816	39.869	134.58
	Optimized + AVX2 + MT + IPT	21.937	2.638	3.495	6.471	34.541
	Optimized + AVX512	46.590	1.432	35.491	35.875	119.388
	Optimized + AVX512 + MT + IPT	16.349	3.063	4.844	3.800	28.056

Table 2. Computation time comparison of the state-of-the-art view synthesis method RVS [Fac18] and all tested implementations of proposed synthesis method on 2K×2K sequence (Cyberpunk). Processing stages: DP – depth projection, DM – depth merging, VP – view projection, PP – postprocessing.

Quality metric	ClassroomVideo		Hijack		Cyberpunk		Average		
	RVS	Proposed	RVS	Proposed	RVS	Proposed	RVS	Proposed	Difference
WS-PSNR [Sun17]	31.76	31.53	38.36	38.17	28.64	28.76	32.92	32.82	-0.10
IV-PSNR [MPEG20]	44.79	44.23	46.01	46.33	37.47	37.57	42.76	42.71	-0.04
VMAF [Li16]	38.32	40.27	71.18	67.30	39.73	37.49	49.74	48.35	-1.38
SSIM [Wan04]	0.927	0.921	0.987	0.986	0.861	0.869	0.925	0.925	0.001
MS-SSIM [Wan03]	0.705	0.656	0.947	0.944	0.658	0.673	0.770	0.758	-0.012
VIF [She06]	0.366	0.346	0.793	0.773	0.341	0.326	0.500	0.482	-0.018

Table 3. Average quality of synthesized virtual views.

In order to present the efficiency of the proposed view synthesis algorithm, the computational time needed for view synthesis was evaluated on three different CPUs: Intel i7-8700K, AMD R9-3900X and Intel i9-7900X. Processors used for evaluation differs both in architecture and in number of available cores. The i9-7900X is the only one being capable of executing AVX512 instructions which were available for performance evaluation.

The complexity of each tested implementation was evaluated as an average processing time needed for synthesis of one frame of a virtual view. The processing time was measured using precision time stamps according to [MDNL20]. For implementations developed by paper authors the processing times for each processing stage (depth projection, depth merging, view projection and postprocessing) was also gathered.

The quality of synthesized views was assessed using 6 state-of-the-art full-reference objective quality metrics: Weighted-to-Spherically-Uniform PSNR (WS-PSNR) [Sun17], Structural Similarity Index Measure (SSIM) [Wan04], Multi-Scale SSIM (MS-SSIM) [Wan03], Visual Information Fidelity (VIF) [She06], Video Multimethod Assessment Fusion (VMAF) [Li16], and ISO/IEC MPEG's metric for immersive video – IV-PSNR [MPEG20].

The quality of the synthesis was estimated by comparing input views with virtual views synthesized at the same position.

Evaluation results

The results of performed experiments are presented in Tables 1 – 3. Tables 1 and 2 show the average computational time required for synthesis of one frame of the virtual view, for 4K×2K and 2K×2K sequence, respectively.

The performance of proposed synthesis technique has been measured as average computation time required to synthesize one video frame. Independently of the platform, even the unoptimized implementation of proposed technique was at least order of magnitude faster than RVS. The algorithmic and implementation related optimizations allows for ~5% reduction in computational time. The higher gain could be achieved by using AVX2 vectorized implementation (up to 87%). The change from AVX2 to AVX512 leads only to small improvements since the used processor (i9-7900X) combines two 256-bit execution units into one 512-bit. The most gain in AVX512 implementation comes from more efficient EVEX encoding, reduced processor front-end burden and usage of mask registers.

The parallelization techniques allow for significant improvements in synthesis performance but is strongly correlated with number of available CPU cores. The combined gain from parallelization techniques (typical multithreaded implementation + IPT) allows to speedup computations by 4 times.

Fortunately, both vectorization and parallelization can be constructively combined leading to almost 14× better performance when compared to optimized implementation.

For the 4K×2K test sequence the best measured performance is ~16 FPS (with ~21 FPS with reduced output resolution). This cannot be treated as real-time, but the value is close to 25 FPS and further improvements in CPU performance and some tuning of implementation could allow for real time processing.

For 2K×2K resolution the framerate of ~38 FPS was achieved implying, that the proposed virtual view synthesis algorithm can operate in the real-time for high-resolution immersive content.

In Table 3, average objective quality metrics for each sequence are presented. It has to be noted, that the quality of the virtual view does not depend on the implementation of the proposed synthesis method.

Fig. 3 presents the subjective comparison between fragments of views synthesized using RVS (left) and proposed method (right). The characteristics of synthesis artifacts are different because of different inpainting methods and the general rule of reprojection (triangle-based projection in RVS and fast pixel-based projection in the proposed algorithm). However, it can be stated that the overall subjective quality of views synthesized using both tested methods is similar.

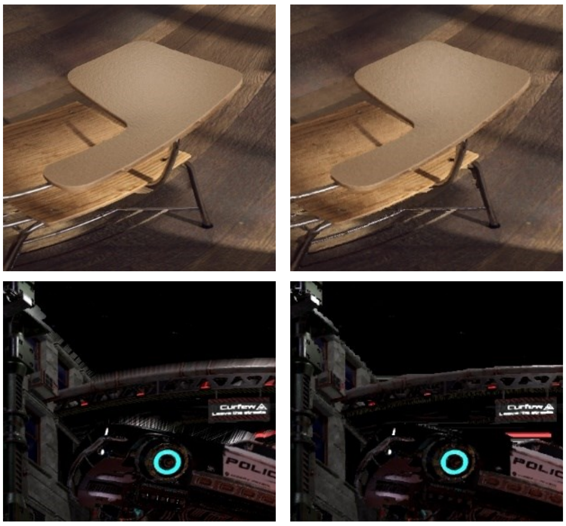


Figure 3. Fragments of virtual views synthesized using RVS (left) and proposed method (right).

5. CONCLUSIONS

The virtual view synthesis for omnidirectional views requires more calculations and is less susceptible to reprojection simplifications than for typical, perspective views. However, the paper shows, that the development of the CPU-based implementation of the real-time virtual view synthesis method is possible also for such kind of content.

The experimental results show that good-quality virtual views can be synthesized in the real-time, providing the possibility of development of cheap immersive video systems in the near future.

6. ACKNOWLEDGMENTS

This work was supported by the Ministry of Education and Science of Republic of Poland.

7. REFERENCES

[Aki15] Akin, A., Capoccia, R., Narinx, J., Masur, J., Schmid, A., and Leblebici, Y. Real-time free viewpoint synthesis using three-camera disparity estimation hardware. 2015 IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, pp. 2525-2528, 2015.

[Ber00] Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. Image inpainting. SIGGRAPH 2000, New Orlean, USA, 2000.

[Dar10] Daribo, I., and Pesquet-Popescu, B. Depth-aided image inpainting for novel view synthesis. 2010 IEEE International Workshop on Multimedia Signal Processing, Saint Malo, France, 2010.

[Dor18] Doré, R. Technicolor 3DoF+ test materials. ISO/IEC JTC1/SC29/WG11 MPEG, M42349, San Diego, CA, USA, 04.2018.

[Dem13] Demikhovsky, E. Intel® AVX-512 Architecture. Comprehensive vector extension for HPC and enterprise, LLVM Developers' Meeting, San Francisco, USA, 2013.

[Dzi18] Dziembowski, A., and Stankowski, J. Real-time CPU-based virtual view synthesis. 2018 International Conference on Signals and Electronic Systems (ICSES), Kraków, Poland, 2018.

[Dzi18b] Dziembowski, A., Samelak, J., Domański, M., "View selection for virtual view synthesis in free navigation systems," International Conference on Signals and Electronic Systems, ICSES 2018, Kraków, Poland, 10-12.09.2018.

[Dzi19] Dziembowski, A., Mieloch, D., Stankiewicz, O., Domański, M., Lee, G., and Seo, J. Virtual view synthesis for 3DoF+ video. 2019 Picture Coding Symposium (PCS), Ningbo, China, 2019.

[Fac18] Fachada, S., Bonatto, D., Schenkel, A., and Lafruit, G. Depth image based view synthesis with multiple reference views for virtual reality. 3DTV-Conference: The True Vision – Capture, Transmission and Display of 3D Video (3DTV-CON), Helsinki, Finland, 2018.

[Goo12] Goorts, P., Dumont, M., Rogmans, S., and Bekaert, P. An end-to-end system for free viewpoint video for smooth camera transitions. 2012 International Conference on 3D Imaging (IC3D). Liege, Belgium, 2012.

[Hua19] Huang, H., Wang, Y., Chen, W., Lin, P. and Huang, C. System and VLSI implementation of phase-based view synthesis. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, pp. 1428-1432, 2019.

[Ilo19] Ilola, L., Vadakital, V.K.M., Roimela, K., and Keraenen, J. New test content for immersive video – Nokia Chess. ISO/IEC JTC1/SC29/WG11 MPEG, M50787, Geneva, Switzerland, 10.2019.

[Isg14] F. Isgro et al. Three-dimensional image processing in the future of immersive media. IEEE Tr. on Circuits and Systems for Video Tech., 2014.

[Jeo22] Jeong, J.Y., Yun, K.J., Lee, G., Cheong, W.S., Yoo, S. "[MIV] ERP Content Proposal for MIV ver.1 Verification Test," ISO/IEC JTC1/SC29/WG04 MPEG VC, M58433, Online, Jan. 2022.

[Kro18] Kroon, B. 3DoF+ test sequence ClassroomVideo. ISO/IEC JTC1/SC29/WG11 MPEG, M42415, San Diego, CA, USA, 04.2018.

[Li16] Li, Z., Aaron, A., Katsavounidis, I., Moorthy, A., and Manohara, M. Toward a practical perceptual video quality metric. Netflix Technology Blog, 2016.

[Li19] Li, Y., Claesen, L., Huang, K., and Zhao, M. A real-time high-quality complete system for depth image-based rendering on FPGA. IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 4, pp. 1179-1193, 2019.

[MDNL20] Microsoft Developer Network Library. Acquiring high-resolution time stamps. https://msdn.microsoft.com/enus/library/windows/desktop/dn553408, 2020.

[MPEG18] "Reference View Synthesizer (RVS) manual," Doc. ISO/IEC JTC1/SC29/WG11 MPEG, N18068, Macao, Oct. 2018.

[MPEG20] Software manual of IV-PSNR for Immersive Video. ISO/IEC JTC1/SC29/WG04 MPEG VC, N0013, Online, Oct. 2020.

[MPEG21] Common Test Conditions for MPEG Immersive Video. ISO/IEC JTC1/SC29/WG04 MPEG VC, N0051, Online, Jan. 2021.

[MPEG21b] Test Model 11 for MPEG Immersive video. Document ISO/IEC JTC1/SC29/WG04 MPEG VC, N0142, Online, Oct. 2021.

[Non18] Nonaka, K., Watanabe, R., Chen, J., Sabirin, H., and Naito, S. Fast plane-based free-viewpoint synthesis for real-time live streaming. 2018 IEEE Visual Communications and Image Processing (VCIP), Taichung, Taiwan, pp. 1-4, 2018.

[She06] Sheikh, H.R., and Bovik, A.C. Image information and visual quality. IEEE Transactions on Image Processing, vol. 15, no. 2, pp. 430-444, 2006.

[Sta18] Stankiewicz, O., Domański, M., Dziembowski, A., Grzelka, A., Mieloch, D., Samelak, and J. A Free-viewpoint Television system for horizontal virtual navigation. IEEE Transactions on Multimedia, vol. 20, no. 8, pp. 2182-2195, 2018.

[Sta20] Stankowski, J., and Dziembowski, A. Fast view synthesis for immersive video systems. Proceedings of the 28. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG'2020, Plzen, Czech Republic, 05.2020.

[Sun17] Sun, Y., Lu, A., and Yu, L. Weighted-to-Spherically-Uniform Quality Evaluation for Omnidirectional Video. IEEE Signal Processing Letters 24.9(2017):1408-1412.

[Tan12] Tanimoto M. et al. FTV for 3-D Spatial Communication. 2012 Proceedings of the IEEE, vol. 100, no. 4, pp. 905-917, 2012.

[Wan03] Wang, Z., Simoncelli, E.P., and Bovik, A.C. Multiscale structural similarity for image quality assessment. The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, vol. 2, pp. 1398-1402, 2003.

[Wan04] Wang, Z., Bovik, A.C., Sheikh, H.R., and Simoncelli, E.P. "Image quality assessment: From error measurement to structural similarity," IEEE Transactions on Image Processing, vol. 13, Jan. 2004.

[Zit04] Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., and Szeliski, R. High-quality video view interpolation using a layered representation. ACM Transactions on Graphics, vol. 3, pp. 600-608, 2004.

[Zha17] Zhang, L., Li, Y., Zhu, Q., and Li, M. Generating virtual images for multi-view video. Chinese Journal of Electronics, vol. 26, no. 4, pp. 810-813, 2017.

The Study of the Video Encoder Efficiency in Decoder-Side Depth Estimation Applications

Adam Grzelka
adam.grzelka

Adrian Dziembowski
adrian.dziembowski

Dawid Mieloch
dawid.mieloch

Marek Domański
marek.domanski

@put.poznan.pl
Institute of Multimedia Telecommunications
Poznań University of Technology
Polanka 3
61-131 Poznań
Poland

ABSTRACT

The paper presents a study of a lossy compression impact on depth estimation and virtual view quality. Two scenarios were considered: the approach based on ISO/IEC 23090-12 coder-agnostic MPEG Immersive video standard, and the more general approach based on simulcast video coding. The commonly used compression techniques were tested: VVC (MPEG-I Part 3 / H.266), HEVC (MPEG H part 2 / H.265), AVC (MPEG 4 part 10 / H.264), MPEG-2 (MPEG 2 part 2 / H.262), AV1 (AOMedia Video 1), VP9 (AOMedia VP9). The quality of virtual views generated from the encoded stream was assessed by the IV-PSNR metric which is adapted to synthesized images. The results were presented as a relationship between virtual view quality and the quality of decoded real views. The main conclusion from performed experiments is that encoding quality and virtual view quality are encoder-dependent, therefore, the used video encoder should be carefully chosen to achieve the best quality in decoder-side depth estimation.

Keywords

Multiview Video, Immersive Video Encoding, Depth Estimation, Virtual View Synthesis

1 INTRODUCTION

In the immersive video, a viewer has an opportunity to change his/her position and orientation in a three-dimensional scene. It enables fully immersive virtual navigation using head-mounted displays or a more simple change of viewpoint displayed on a traditional screen. In order to provide virtual views to the final user, it is required to acquire a scene from a number of views and estimate its three-dimensional geometry. As these views and geometry (usually represented in the form of depth maps) have to be sent to the renderer which generates the requested viewpoint, they usually are compressed using dedicated immersive video codecs, or simply using versatile video codecs. Lossless encoding has limited applications because even after compressing these data, the sufficient bitrate required to send it is usually in the range between 5 and 50 Mbps [Boy21][Fis20].

One of the possible solutions for decreasing the bitrate of immersive video is the estimation of geometry (depth) in the decoder, using the decoded views. This scheme of compression was already proved to be efficient in many applications [Gar21] and was included as one of the profiles of the new MPEG Immersive video (MIV) coding standard [Boy21], called MIV Geometry Absent (GA) [Mie22]. All of the profiles are codec-agnostic, i.e., after the initial pre-processing of input data, they are utilizing the traditional video encoders to encode the MIV representation.

While the MIV standard makes it possible to use any available video encoder, during the works of ISO/IEC MPEG it was mainly tested and tuned using other newest codecs from this group. The works presented in this paper were performed to find the answer to two questions related to the codec-agnosticism of MIV. First of all, what is the performance of MIV GA with other video codecs not related to MPEG standards? Secondly, how does using these different encoders impact the efficiency of different implementations of a decoder-side depth estimation scheme?

https://www.doi.org/10.24132/CSRN.3201.31

The paper is organized as follows: Section 2 describes the overview of the decoder-side depth estimation scheme and includes a description of its individual parts. Section 3 shows the methodology of experiments proposed to evaluate the DSDE in order to answer the abovementioned questions. The results of the experiments and their discussion are presented in Section 4, while the final conclusions and summary are presented in the last Section 5.

2 DECODER-SIDE DEPTH ESTIMATION

The decoder-side depth estimation (DSDE) approach shifts some of the video processing steps from the encoder to the decoder, making the decoding process more sophisticated and time-consuming. The video processing performed in the decoder operating in the DSDE approach comprises three major steps:

1. video decoding,
2. depth estimation,
3. virtual view synthesis.

When analyzing the entire data flow (not only the video sub-bitstreams), an additional step should be considered – metadata decoding. These metadata include camera parameters and other crucial information about views, or parameters used in depth maps estimation, e.g. bit depth [Gar21].

The first step of the video processing is a simple video decoding, performed by a typical 2D video decoder, e.g. VVC or HEVC. This step is crucial as it restores source views from the bitstream, but in this paper it is not considered and treated as trivial.

In the second step, the most time-consuming process is performed, allowing to estimate the geometry of the scene based on information sent within input views [Gar21] and decoded metadata of the multiview video.

There are numerous depth estimation methods described in the literature, including recent high-quality methods, e.g. graph-optimization-based methods described in [Rog19] and [Nam21], or methods based on using neural networks, e.g. GANet [Zha19] or GWCNet [Guo19]. However, as was presented in [Mie22], the most suitable method for the DSDE and overall immersive video applications is IVDE (Immersive Video Depth Estimation) [Mie20], developed by the ISO/IEC MPEG Video Coding group with its tools allowing proper depth estimation even for highly compressed input views [Mie21].

The last step of the decoding in any immersive video system, including the DSDE approach, is the rendering of viewports requested by the viewer. Such a rendering requires input views, corresponding depth maps, and camera parameters as input data, and outputs any view, created by reprojection of pixels [Dzi19a], [Fac18] followed by operations increasing the quality of rendered views such as filtering or inpainting [Jia21].

3 OVERVIEW OF THE EXPERIMENT

In order to properly assess the efficiency of different video encoders in the decoder-side depth estimation applications, two scenarios were tested. In both scenarios, the virtual views are generated from a lossy compressed multiview sequence.

The first scenario is based on using the newest ISO/IEC standard for immersive video compression: MPEG Immersive video (MIV). In the second one, a more general approach is considered, in which all the source views are separately encoded and used as the input for the standalone depth estimator and view synthesizer at the decoder side.

MPEG Immersive Video

The block diagram of the multiview video processing in the first experiment is presented in Fig. 1. As the MIV standard is codec-agnostic, any video encoder and decoder (blue blocks in Fig. 1) can be used to encode and decode the "atlases" produced by the MIV encoder (using the MIV Geometry Absent profile [Mie22]).

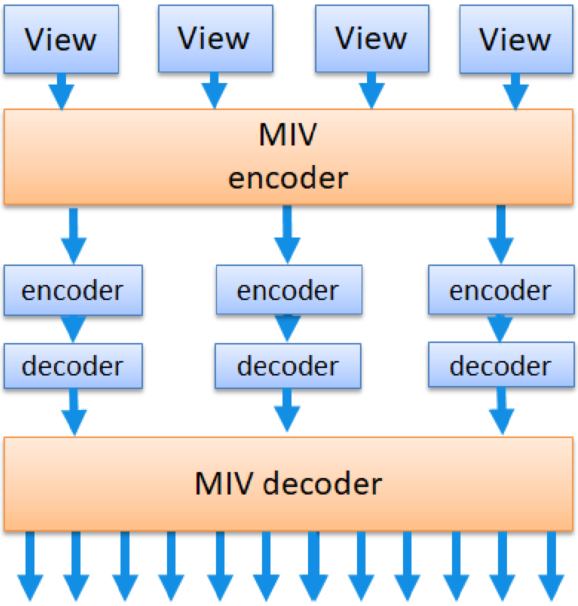


Figure 1: Block diagram of the MIV experiment.

This experiment was performed under the MIV Common Test Conditions (MIV CTC) [MPEG21b] developed by the ISO/IEC MPEG Video Coding group, which defines the entire pipeline for immersive video encoding, including detailed rules for encoding, processing, and quality assessment of the immersive video, as well as a set of 15 miscellaneous sequences (Table 1), including both omnidirectional and perspective sequences, computer-generated content and natural sequences captured by real multicamera systems. According to the MIV CTC, for each test sequence, 17 frames were encoded.

Sequence	Resolution	Frames	Views
Carpark	1920 × 1088	250	9
Chess	2048 × 2048	300	10
ChessPieces	2048 × 2048	300	10
ClassroomVideo	4096 × 2048	120	15
Fan	1920 × 1080	97	15
Fencing	1920 × 1080	250	10
Frog	1920 × 1080	300	13
Group	1920 × 1080	99	21
Hall	1920 × 1088	500	9
Hijack	4096 × 2048	300	10
Kitchen	1920 × 1080	97	25
Mirror	1920 × 1080	100	15
Museum	2048 × 2048	300	24
Painter	2048 × 1088	300	16
Street	1920 × 1088	250	9

Table 1: Parameters of MIV CTC squenceces.

In the experiment, the effectiveness of four different video encoders was assessed, including two encoders developed by ISO/IEC MPEG: VVC [Bro21] in the optimized implementation VVenC [Wie21] and fast implementation of HEVC [Sul12]: x265 [x265]; as well as two royalty-free encoders: AV1 and VP9, both implemented in FFmpeg 4.4.1 [ffmpeg].

At the decoder side, the synthesized input views (virtual views synthesized at the position of input ones) were generated using the MIV decoder, which includes the decoder-side depth estimation implemented in IVDE software [Mie20] [MPEG21c] and the renderer implemented in the TMIV 9 software (Test Model 9 for MPEG Immersive video) [MPEG21a].

The objective quality was measured as IV-PSNR [MPEG20] measured between input views and synthesized input views. The IV-PSNR was calculated for all input views and is presented as a mean value, averaged over all views and all 17 frames.

General approach

This scenario is an extension of the experiment performed by the authors of this paper and presented in [Dzi16], presenting an influence of the newest coding techniques on top of previously tested encoders.

The multiview video processing pipeline used for the second experiment is presented in Fig. 2. In this experiment, all the input views are separately encoded using four simulcast encoders, including two encoders tested in the first scenario: VVC and HEVC, and two older techniques: AVC in the x264 implementation [x264] and MPEG-2 implemented within the FFmpeg 4.4.1 [ffmpeg]. All used encoders are optimized and publicly available, increasing the reproducibility of presented experimental results.

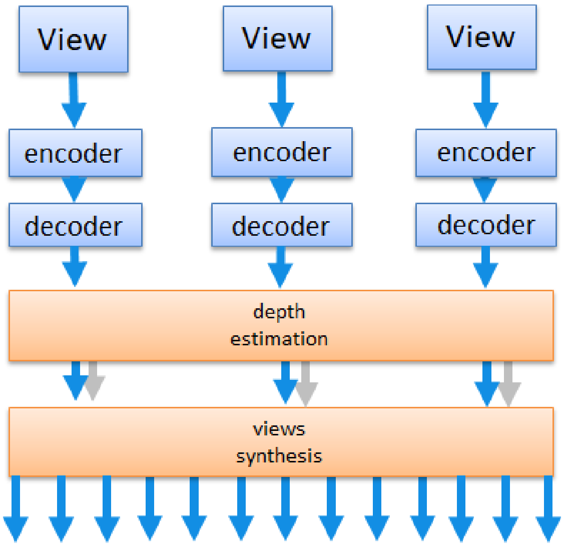


Figure 2: Block diagram of the general approach experiment.

On the decoder side, two multiview video processing algorithms were used. For depth estimation, the same IVDE algorithm [Mie20] was used to ensure, that the results of both performed experiments are not influenced by introducing different depth artifacts. For virtual view synthesis, the Advanced View Synthesizer described in [Dzi19a] was used. The advantage of this synthesizer is the possibility of easy optimization and fast implementation what was presented in [Sta20].

In this experiment, 8 multiview test sequences (Table 2) were used, including sequences captured by linear and circular multicamera systems [MPEG08], [MPEG15]. For all the sequences, more than 30 input views were used. For each sequence, four views were used as input ones for the entire processing (Fig. 2), while the rest was used for the quality assessment purposes, allowing proper and accurate objective quality assessment.

Sequence	Resolution	Frames	Views
BBB Butterfly Arc	1280 × 768	120	91
BBB Butterfly Lin.	1280 × 768	120	91
BBB Flowers Arc	1280 × 768	120	91
BBB Flowers Lin.	1280 × 768	120	91
BBB Rabbit Arc	1280 × 768	120	91
BBB Rabbit Lin.	1280 × 768	120	91
Dog	1280 × 960	300	80
Pantomime	1280 × 960	500	80

Table 2: Parameters of multiview sequences.

To be compliant with the first experiment, 17 consecutive frames were processed for each test sequence.

4 EXPERIMENTAL RESULTS

MPEG Immersive Video

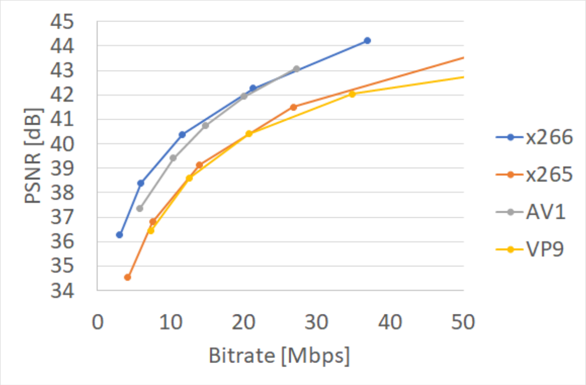


Figure 3: PSNR rate-distortion curves for decoded atlases.

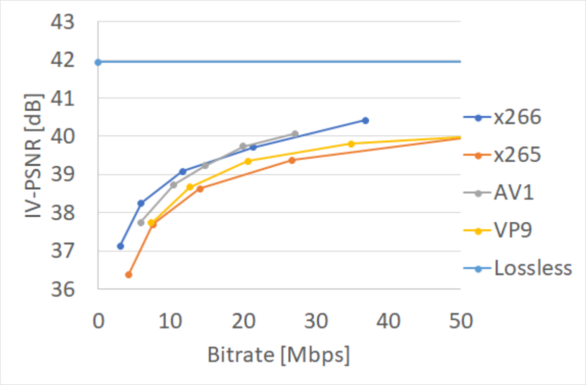


Figure 4: IV-PSNR rate-distortion curves for synthesized virtual views.

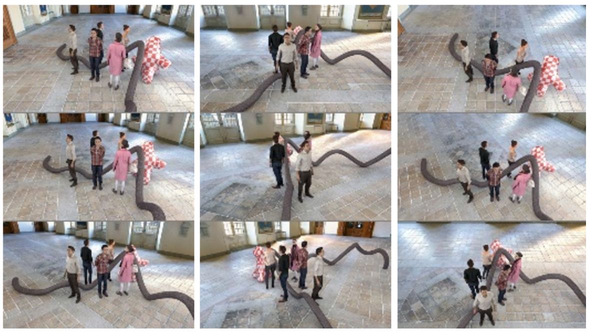


Figure 5: Atlases generated by the MIV encoder, sequence Group.

Figures 3 and 4 show the efficiency of four tested video encoders. In Fig. 3, the efficiency is presented in terms of the PSNR rate-distortion curves for decoded video (i.e., atlases, see Fig. 5). Fig. 4 presents the dependency between the total bitrate required for transmission of the immersive video encoded with different encoders and the mean quality of synthesized views (IV-PSNR averaged over 15 sequences, 17 frames, and all synthesized input views).

It should be noted that the bitrates presented in Figs. 3 and 4 are exactly the same, as they correspond to the same immersive video bitstreams.

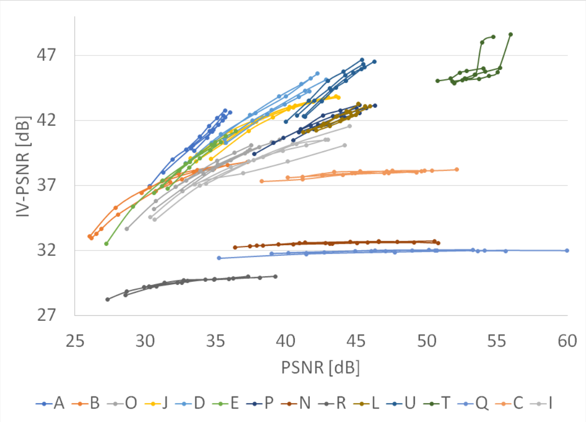


Figure 6: Dependency between decoding quality and synthesis quality for all MIV CTC sequences.

When comparing the results obtained using different encoders, some general observations can be stated. Firstly, MIV is indeed a codec-agnostic standard, and the RD-curves for all the encoders look similarly. Secondly, VVC and AV1 encoders in used fast implementations perform similarly both in terms of quality of decoded atlases and synthesized virtual views. On the other hand, the results for HEVC and VP9 seem to be more interesting. In terms of the quality of the decoded atlases, both encoders provide similar results. However, when comparing the IV-PSNR of the synthesized virtual views, a slight but noticeable advantage of VP9 can be found.

The possibility of quality assessment for two points of the decoder (before view synthesis – Fig. 3 and after view synthesis – Fig. 4) allows drawing a dependency between these two qualities, which is presented in Fig. 6. The results shown in Fig. 6 are drawn separately for each test sequence and each tested video encoder, presenting a dependency between the average PSNR of the decoded atlases and the average IV-PSNR of synthesized views. Curves for each sequence are colored differently.

As shown in Fig. 6, the majority of the curves are grouped. The only outliers can be found for sequences Q (ChessPieces), N (Chess), C (Hijack), and R (Group), for which the curves are almost horizontal. It means, that for these sequences the quality of the synthesized views does not depend on the quality of decoded atlas, thus increasing the total bitrate does not improve the user's experience.

In general, all the curves can be approximated by the linear equation:

$$\text{IV-PSNR}(view) \approx a \cdot \text{PSNR}(atlas) + b \qquad (1)$$

Sequence	ID	a	b
ChessPieces	Q	0.02	31.06
Chess	N	0.03	31.43
Hijack	C	0.06	35.35
Group	R	0.13	25.15
Carpark	P	0.43	23.25
Mirror	I	0.44	21.67
Fencing	L	0.45	22.32
Kitchen	J	0.50	22.73
Hall	T	0.55	16.47
Museum	B	0.55	19.27
Fan	O	0.61	16.80
Painter	D	0.67	16.97
Street	U	0.83	8.70
Frog	E	0.90	8.88
ClassroomVideo	A	0.99	7.08

Table 3: Linear approximation results for MIV CTC sequences.

Values of parameters a and b estimated for all test sequences can be found in Table 3. An example of curves for two sequences is presented in Figs. 7 and 8. Values highlighted in red correspond to the outliers in Fig. 6. For all these sequences the correlation between the quality of decoded atlases and synthesized virtual views is extremely low. It is caused by the appearance of strong synthesis artifacts in the virtual views (Fig. 9).

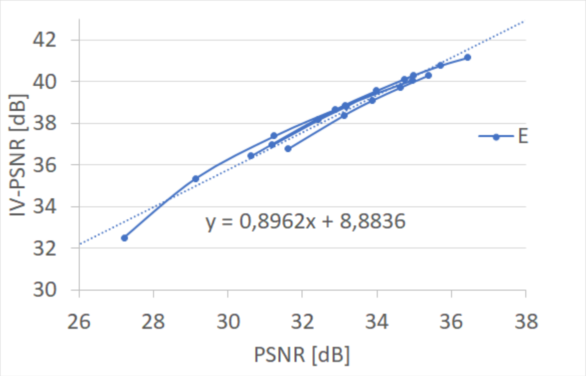


Figure 7: Linear approximation for Frog sequence.

General approach

Figs. 10 and 11 gather the results of the second experiment, presented in the same way, as for the experiment using the MPEG Immersive Video coding standard presented in the previous subsection. Fig. 10 contains the dependency between the total bitrate needed for transmission of all (four) input views and the quality of decoded views. Fig. 11 presents the results of the virtual

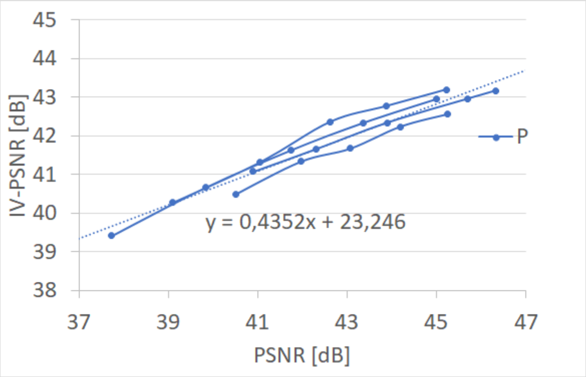


Figure 8: Linear approximation for Carpark sequence.

view synthesis, which was performed using these decoded views.

Similarly to the previous subsection, also the dependency between both qualities was measured and reported. Calculated values of parameters a and b of the linear equation bonding the IV-PSNR of the virtual view with PSNR of the decoded input views are

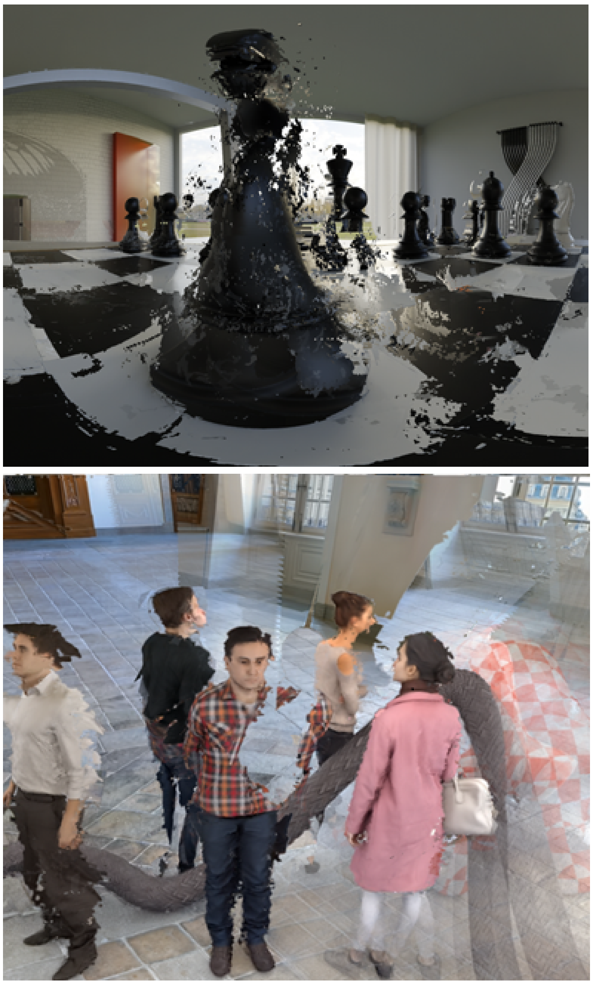


Figure 9: Synthesized views with strong artifacts, sequences ChessPieces and Group.

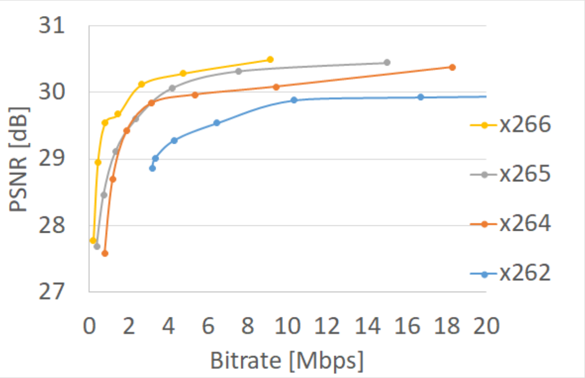


Figure 10: PSNR rate-distortion curves for decoded source views.

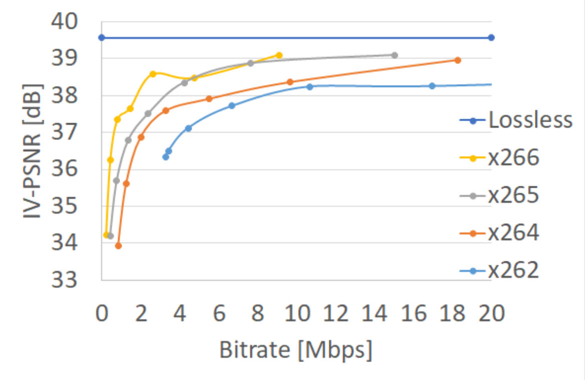


Figure 11: IV-PSNR rate-distortion curves for synthesized virtual views.

reported in Table 4. PSNR/IV-PSNR curves obtained for two test sequences are presented in Figs. 13 and 14.

Four sequences were highlighted in red in Table 4. For these sequences, the correlation between synthesis IV-PSNR and decoding PSNR is very low. For these sequences, many disturbing artifacts can be found in the synthesized virtual views, as presented in Fig. 12. Such a dependency is consistent with the observations taken for the first experiment, showing the relevance of both tested scenarios.

Sequence	a	b
BBB Flowers Lin.	0.1	23.04
BBB Flowers Arc	0.1	22.48
BBB Butterfly Lin.	0.15	29.92
Pantomime	0.16	32.29
BBB Butterfly Arc	0.22	30.89
Dog	0.25	26.31
BBB Rabbit Lin.	0.32	25.71
BBB Rabbit Arc	0.34	23.98

Table 4: Linear approximation results for 8 multiview sequences.

Obtained results follow the expectations, as newer and more advanced encoding standards perform better than the older ones, both in terms of the decoding quality and the quality of synthesized virtual views.

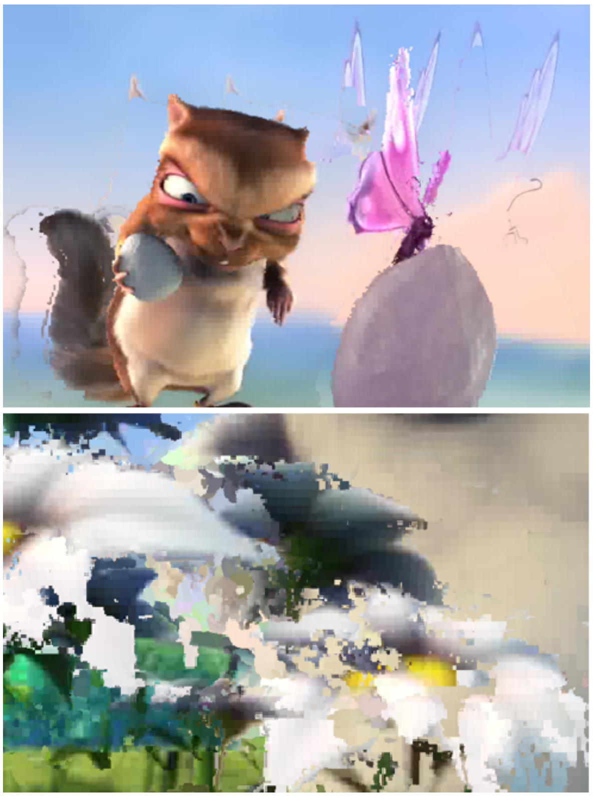


Figure 12: Synthesized views with strong artifacts, sequences BBB Flowers Lin. and BBB Butterfly Lin.

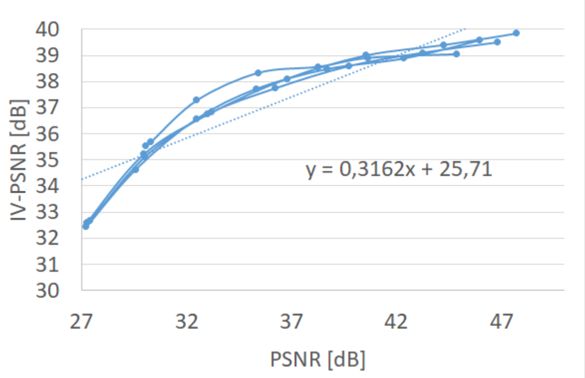


Figure 13: Linear approximation for BBB Rabbit Lin.

https://www.doi.org/10.24132/CSRN.3201.31

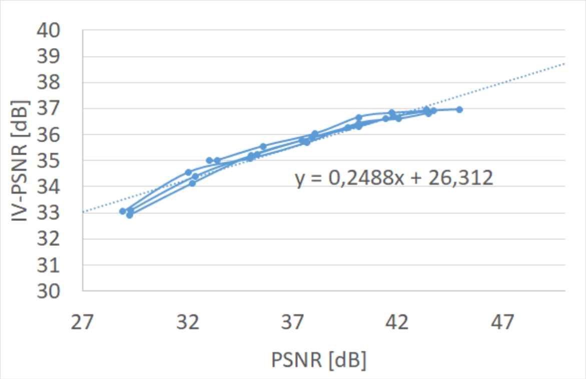


Figure 14: Linear approximation for Dog sequence.

5 CONCLUSIONS

In the paper, we have analyzed the influence of the lossy compression introduced by various video encoders on the depth map estimation process. Such research is very important and allows us to make some crucial observations.

At first, there is a strong correlation between the quality of the decoded input views and the quality of virtual views synthesized based on them.

Secondly, the newest ISO/IEC standard for immersive video compression – the results prove that MPEG Immersive Video (MIV) is indeed a "codec-agnostic" technique and any video codec can be used with it, nevertheless, the used codec significantly impacts the quality of synthesized virtual views thus the viewer's experience.

The third observation is that VP9 and the optimized implementation of the HEVC encoder (x265) provide similar quality. However, when comparing the IV-PSNR of the synthesized virtual views, a slight but noticeable increase the final quality for VP9 can be found.

At last, the dependency between the quality of the decoded input views and the quality of the synthesized views can be expressed by a linear approximation. The slope of this linear approximation can suggest if a sequence is easy to be properly synthesized. The steep trend line suggests, that the virtual view is visually consistent; if the trend line is almost horizontal, the virtual view has noticeable rendering artifacts.

All of the presented observations and conclusions suggest that efficient decoder-side depth estimation is possible.

6 ACKNOWLEDGMENTS

This work was supported by the Ministry of Education and Science of Republic of Poland.

7 REFERENCES

[Boy21] Boyce, J. et al. "MPEG Immersive Video Coding Standard," in Proceedings of the IEEE, vol. 109, no. 9, pp. 1521-1536, Sept. 2021, doi: 10.1109/JPROC.2021.3062590.

[Bro21] Bross, B. et al. "Overview of the Versatile Video Coding (VVC) standard and its applications," IEEE Tr. on Circ. and Syst. for Vid. Tech., 2021, doi: 10.1109/TCSVT.2021.3101953.

[Dzi16] Dziembowski, A. et al. "The influence of a lossy compression on the quality of estimated depth maps," 2016 International Conference on Systems, Signals and Image Processing (IWSSIP), 2016, pp. 1-4, doi: 10.1109/IWSSIP.2016.7502730.

[Dzi19a] Dziembowski, A. et al. "Virtual View Synthesis for 3DoF+ Video," 2019 Picture Coding Symposium (PCS), 2019, pp. 1-5, doi: 10.1109/PCS48520.2019.8954502.

[Dzi19b] Dziembowski, A. and Domański, M. "Objective quality metric for immersive video", ISO/IEC JTC1/SC29/WG11 MPEG2019/M48093, July 2019, Göteborg, Sweden

[Fac18] Fachada, S. et al. "Depth image based view synthesis with multiple reference views for virtual reality," 2018 - 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2018, pp. 1-4, doi: 10.1109/3DTV.2018.8478484.

[ffmpeg] FFmpeg encoder available at: www.ffmpeg.org.

[Fis20] Fischer, R., et al. "Improved Lossless Depth Image Compression", Journal of WSCG 28, 2020, 168-176, doi: 10.24132/JWSCG.2020.28.21.

[Gar21] Garus, P. et al. "Immersive Video Coding: Should Geometry Information be Transmitted as Depth Maps?," IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 5, pp. 3250-3264, May 2022, doi: 10.1109/TCSVT.2021.3100006.

[Guo19] Guo, X. et al. "Group-Wise Correlation Stereo Network," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3268-3277, doi: 10.1109/CVPR.2019.00339.

[Jeo21] Jeong, J.Y. et al. "DWS-BEAM: Decoder-Wise Subpicture Bitstream Extracting and Merging for MPEG Immersive Video," 2021 International Conference on Visual Communications and Image Processing (VCIP), 2021, pp. 1-5, doi: 10.1109/VCIP53242.2021.9675419.

[Jia21] Jia, B. et al. "Virtual view synthesis for the nonuniform illuminated between views in surgical video." Multim. Tools Appl. 80 (2021): 20619-20639, doi: 10.1007/s11042-021-10732-3.

[Mie20] Mieloch, D. et al. "Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation," in IEEE Access, vol. 8, pp. 5760-5776, 2020, doi: 10.1109/ACCESS.2019.2963487.

[Mie21] Mieloch, D. et al. "Point-to-Block Matching in Depth Estimation," International Conference on Computer Graphics, Visualization and Computer Vision WSCG 2021, doi: 10.24132/CSRN.2021.3002.15.

[Mie22] Mieloch, D. et al. "Overview and Efficiency of Decoder-Side Depth Estimation in MPEG Immersive Video," in IEEE Transactions on Circuits and Systems for Video Technology, doi: 10.1109/TCSVT.2022.3162916.

[MPEG08] "1D Parallel Test Sequences for MPEG-FTV," ISO/IEC JTC 1/SC 29/WG11 M15378, Apr. 2008.

[MPEG15] "[FTV AHG] Big Buck Bunny light-field test sequences," ISO/IEC JTC 1/SC 29/WG11 M35721, Feb. 2015.

[MPEG20] "Software manual of IV-PSNR for Immersive Video," ISO/IEC JTC 1/SC 29/WG04 N0013, Oct. 2020.

[MPEG21a] "Test Model 9 for MPEG Immersive Video," ISO/IEC JTC 1/SC 29/WG04 N0084, May 2021, Online.

[MPEG21b] "Common Test Conditions for MPEG Immersive Video," ISO/IEC JTC 1/SC 29/WG04 N0085, May 2021, Online.

[MPEG21c] "Manual of IVDE 3.0," ISO/IEC JTC1/SC29/ WG04 N0058, Jan. 2021.

[Nam21] Nam, D.Y. and Han, J.K. "Improved Depth Estimation Algorithm via Superpixel Segmentation and Graph-cut," 2021 IEEE International Conference on Consumer Electronics (ICCE), 2021, pp. 1-7, doi: 10.1109/ICCE50685.2021.9427631.

[Rog19] Rogge, S. et al. "MPEG-I Depth Estimation Reference Software," in 2019 International Conference on 3D Immersion (IC3D), 2019, pp. 1-6, doi: 10.1109/IC3D48390.2019.8975995.

[Sta20] Stankowski, J., Dziembowski, A., "Fast View Synthesis for Immersive Video Systems", Journal of WSCG 28, 2020, 137-144, doi: 10.24132/CSRN.2020.3001.16.

[Sul12] Sullivan, G. et al. "Overview of the High Efficiency Video Coding (HEVC) standard," IEEE Tr. on Circ. and Syst. for Vid. Tech., vol. 22, 2012, doi: 10.1109/TCSVT.2012.2221191.

[Wie21] Wieckowski, A. et al. "VVenC: An Open And Optimized VVC Encoder Implementation," Proc. IEEE International Conference on Multimedia Expo Workshops (ICMEW), 2021, doi: 10.1109/ICMEW53276.2021.9455944.

[x264] "x264 encoder" available at: https://www.videolan.org/developers/x264.html.

[x265] "x265 encoder" available at: https://x265.com/.

[Xie21] Xie, Y. et al. "Performance analysis of DIBR-based view synthesis with kinect azure," 2021 International Conference on 3D Immersion (IC3D), 2021, pp. 1-6, doi: 10.1109/IC3D53758.2021.9687195.

[Zha19] Zhang, F. et al. "GA-Net: Guided Aggregation Net for End-To-End Stereo Matching," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 185-194, doi: 10.48550/arXiv.1904.06587.

Parallel YOLO-based Model for Real-time Mitosis Counting

Robin Yancey
Department of Computer Science
University of California, Davis
USA (95616) Davis, CA
reyancey@ucdavis.edu

ABSTRACT

It is estimated that breast cancer incidences will increase by more than 50% by 2030 from 2011. Mitosis counting is one of the most commonly used methods of assessing the level of progression, and is a routine task for every patient diagnosed with invasive cancer. Although mitotic count is the strongest prognostic value, it is a tedious and subjective task with poor reproducibility, especially for non-experts. Object detection networks such as Faster RCNN have recently been adapted to medical applications to automatically localize regions of interest better than a CNN alone. However, the speed and accuracy of newer state-of-the-art models such as YOLO are now leaders in object detection, which had yet be applied to mitosis counting. Moreover, combining results of multiple YOLO versions run in parallel and increasing the size of the data in a way that is appropriate for the specific task are some of the other methods can be used to further improve the score overall. Using these techniques the highest F-scores of 0.95 and 0.96 on the MITOS-ATYPIA 2014 challenge and MITOS-ATYPIA 2012 challenge mitosis counting datasets are achieved, respectively.

Keywords

YOLO, deep learning, mitosis counting, breast cancer, histopathology, machine learning, real-time detection

1 Introduction

1.1 Mitotic Count & Issues

The Nottingham Grading System (NGS) is recommended by various professional bodies internationally (World Health Organization [WHO], American Joint Committee on Cancer [AJCC], European Union [EU], and the Royal College of Pathologists (UK RCPath) [17]. It says that tubule formation, nuclear pleomorphism, and mitotic index should each be rated from 1 to 3, with the final score ranging between 3 and 9. This is divided into three grades: Grade 1, score 3-5, well differentiated; Grade 2, score 6-7, moderately differentiated; and Grade 3, score 8-9, poorly differentiated [1].

When pathologists need to make this assessment of the tumor for mitotic count, they start by finding the region with the highest proliferative activity. The mitotic count is used to predict the aggressiveness of a tumor and is defined in a region from ten consecutive high-power fields (HPF) within a space of $2mm^2$. Variation in phase and slide preparation techniques make it possible to misdiagnose. They also often have a low density and can look different depending on whether the mitosis is in one of the four main phases: prophase, metaphase, anaphase, and telophase.

The shape of the cell itself differs significantly for each phase. For example, when in telophase it is split into to separate regions even though they are still one connected mitotic cell. Apoptotic cells (or cells going through preprogrammed cell death) and other scattered pieces of waste on the slides can also easily be confused with mitoses, having a similar dark spotty appearance. Further, mitotic nuclei often resemble many other hyperchromatic cellular bodies such as necrotic and non-dividing dense nuclei, making detection of mitosis more difficult on tissue [27]. The variation in the process of obtaining the slides using different scanners and different preparation techniques may also make distinguishing cells more exhausting. Worse yet, pathologists can get tired and it can make it harder to make proper judgement on slides when trained pathologists need to examine hundreds of high power fields (HPF) of histology images, in a short amount of time. Biopsies can take up to ten days before the patient receives results [18].

The increasing numbers of breast cancer incidences calls for a more time-and cost-efficient method of prognosis, which could later even help to provide care to impoverished regions. Automatic image analysis

https://www.doi.org/10.24132/CSRN.3201.32

has recently proven to be a possible solution, with inter-observer agreement when tested against the human judgement [28].

2 Related Work

2.1 Automatic and Machine Learning Methods

The use and development of automatic detection methods of mitosis counting have gradually been increasing since the end of the 20th century in order to make doctors' jobs easier and more efficient [16]. Due to the recent progress in digital medication, a large amount of of data has became available for use in the medical studies. Machine learning has helped to discover new characteristics of cancer mutations by sorting through more image data than humanly possible and simultaneously analyzing all of the millions of image pixels undetectable to the human eye. For example, in the field of histopathology, machine learning algorithms have been used for analysis of scanned slides to assist in tasks including diagnosis [9]. The use of computing in image analysis may reduce variability in interpretation, improve classification accuracy, and provide clinicians (or those in training) with a second opinion [9]. Existing methods use either handcrafted features captured by specific morphological, statistical, or textural attributes determined by a pathologist or features are automatically learned through the use of convolutional neural networks (CNN).

2.2 Deep Learning Methods

With the help of their strong self-learning qualities, deep learning networks, especially neural networks have also been heavily investigated in medical image processing [26]. CNN's have made a significant impact in machine learning for image classification, segmentation, object detection, and computer vision tasks [5]. Medical applications in particular, such as mitosis detection, cell nucleus segmentation and tissue classification tasks have also been popular tasks for CNN's. This is because pathological images are texture-like in nature, making them ideal task to learn with their shift invariance and pooling operations. Deep learning methods often outperform traditional methods such as use the of handcrafted features alone since feature extractors and can be classifiers simultaneously optimized [23] [33].

CNNs are well-suited to learn high level features such as mitotic figures, which is likely what made these methods winners of the ICPR2012 [22], ICPR 2014 [21], and AMIDA 2013 challenges [29] [18]. These well-known mitosis counting competitions were held at conferences and now are publicly available datasets commonly used for research, further discussed in 3.3.

Ciresan et al., won ICPR 2012 using deep max-pooling convolutional neural networks to classify each image pixel using a patch centered on the pixel as context. The simple CNN consisted of five convolutional layers with max pooling layer, and two fully connected layers [3]. A similar model has also been successfully used in detecting mitoses from the AMIDA13 challenge [12], where a multi-column neural network is used to classify image patches and generate the precise image descriptors.

2.3 Object Detection for Histopathological Image Analysis

On the other hand, it has now become well-known that a basic CNN alone lacks cell level supervision and often requires limiting the size of the input image. This is done so that sub-image features can be learned from localized regions of the image rather than the full image context consisting of multiple objects as well as non-objects (regions of interest). Moreover, object detection or precise localization is actually a more common task than full-image classification in medical applications. Consequently, deep learning methods designed originally for object detection such as R-CNN, Faster R-CNN, and Mask R-CNN have been applied to this to target specific frames from within the image which have been deduced from a ROI (Region of Interest).

For example, Lu et al. cascade detection algorithm based on segmentation and classification and reached 0.83 on the ICPR 2012 data set and 0.58 on the ICPR 2014 data set. It used a cascaded convolutional neural network based on UNet, which consisted of three parts: semantic segmentation and classification to detect mitosis. First UNet is used for segmentation to locate the candidate set of mitotic targets. Second, the cell nucleus is located by means of semantic segmentation to obtain accurate image blocks of mitotic and non-mitotic cells via a Vnet. Third, the cell image output block is used to train a CNN to do binary classification and this area is checked for mitosis [14]. Sebai et al. developed a multi-task deep learning framework for both object detection and instance segmentation tasks using Mask RCNN. First, it is used for segmentation to estimate the mitosis mask labels for the weakly annotated mitosis dataset. This produces the mitosis mask and bounding box labels for training another mitosis detection and instance segmentation model for mitosis detection on the other dataset [25]. They obtained an F-score of 0.86 on the 2012 ICPR dataset and an F-score of 0.48 on the 2014 ICPR dataset. Rao used Faster-RCNN to achieve the highest F-score of 0.96 when their model was trained and tested on all three challenge datasets above combined [18]. This is 6.22% more accurate than

https://www.doi.org/10.24132/CSRN.3201.32

the previous high score of 0.90 achieved by the model proposed by [24].

3 Materials & Methods

The first goal was to test the newer and more advanced object detection networks such as YOLOv3, YOLOv4-scaled, YOLOv5, and YOLOR for the mitosis counting task. The second goal was to try a number of different methods of increasing the size of the training data to further improve prediction accuracy. This included adding images from multiple scanners, combining the two different contest datasets, and multiple forms of image augmentation. Image augmentation helps to reduce overfitting while artificially enlarging the dataset [10]. The third goal was to try running the best YOLO models in parallel for improved accuracy since the training and inference times were exceptionally short compared to former methods.

3.1 Finding the Optimal Model & Configuration

Once the best augmentation combination was found, different numbers of epochs and versions were tested for each YOLO version model to find the optimal setup for speed and accuracy for this specific application. Once this was found, it was used for the further testing.

3.1.1 Combining YOLOv5m-p5 with YOLOR

Both YOLOv5m-p5 and YOLOR consistently produced the highest F-scores, but with different predictions. Also, YOLOR predicted its highest scores at quicker runtimes. Therefore, when the bounding box and confidence scores of each of the predictions made by YOLOv5m-p5 and YOLOR were averaged, the overall results and runtimes could be optimized. This helps to refine the results without loss in efficiency because each of the models can be trained and tested in parallel on a separate cloud GPU.

3.2 Increasing the Size of Training Data

3.2.1 Alternate Data Augmentation

Data augmentation can help add more samples, while increasing variability and diversity in the appearance of each mitotic region. This makes the model more robust towards new examples that show up in the test set with similar characteristics.

The types of augmentation tested included none, blur, noise, rotation, mosaic, brightness, and exposure, and each was compared to when no augmentation was applied. In each case a new training image was added to the dataset for each image augmentation and the unfiltered image was still included in the training set.

3.2.2 Multiple Scanners

To test for the potential change in accuracy by adding data of multiple scanners, training was done with the model on the images from each scanner alone and testing on images from the same scanner on which it was trained. It was then compared that to the results of training on both scanner data combined to see if adding data from the other scanner helps predict. Next, testing was done by alternating the training and testing data to test on data from another scanner besides the one of which it was trained on. These tests are also interesting or useful for realistic situations in which similar training data from the same scanner for the image is missing.

3.2.3 Multiple Databases

Then, to assess the effect of combining data from multiple databases to the training set, the ICPR 2012 training set was combined with the ICPR 2014 training set. If the predictions on the test images from one database alone are better when the model is trained with data from both then this helps us determine how overall useful this could be in real life cancer detection, as well. For example, we could continue to add data to the training set and keep updating the weights to get better predictions on any test set from a new patient.

3.3 Datasets & Preparation

3.3.1 Contest Datasets

The models were trained with two different open datasets from the International Conference on Pattern Recognition (ICPR) of breast cancer histopathology in 2012 [22] and 2014 [21] developed to address this challenging issue. The data is for mitosis counting in images stained with standard hematoxylin and eosin (H&E) dyes obtained from breast biopsies. The hematoxylin stains cell nuclei a purplish blue, while eosin stains the extracellular matrix and cytoplasm pink (and blood cells in red). The Aperio Scanscope XT and the Hamamatsu Nanozoomer 2.0-HT slide scanners have different resolution and are used to produce RGB high-power fields (HPFs). Annotations for the image coordinates of each mitosis are made by two senior pathologists, where if one disagrees a third will give the final say. The ICPR 2012 X40 resolution training dataset consists of 35 images with 226 mitotic cells. The original HPFs are of size 2084 $\times$ 2084 pixels. The ICPR 2014 X40 resolution training set provided consists of 1,136 frames containing a total of 749 labeled mitotic cells. Aperio images are sized 1539 $\times$

1376 pixels, and Hamamatsu images are 1663 × 1485 pixels.

Since the time of the contests both have been very commonly re-used among the research in this area thus far, so testing with these datasets will help compare the results to other published works.

Preprocessing The annotations of *official* test set for the ICPR contests is unavailable to the public so part of this training set was used for the test set. This is similar to what most other research groups (such as those referenced) have done, in order to be able to check the correctness of predictions by their developed framework. Here, the test set was selected randomly by extracting a set of images containing approximately the average number of mitosis in one slide from the provided training set. Also, similar to other groups referenced, the training set was artificially augmented to increase the density of mitosis and avoid class imbalance. Images needed to be cropped due to the small size and number of the mitosis compared to the very large size of the original HPF images.They were then expanded in order for the mitosis to be large enough for the smallest detectable size of the network aspect scales. These patches in mitotic regions were cropped into approximately 64 equal sized subsections from each HPF after being converted to JPEG. The bounding box coordinates were then created by adding 25 pixels in the upper left and lower right directions from the derived provided centroid coordinates.

Finally, each image is expanded to 416×416 and its coordinates are scaled upward accordingly. This is the appropriate input image size and scale for the YOLO network setup, which does internal data augmentation to rotate and resize the images internally. In order to provide consistency and better prediction accuracy (of both large and small objects), it is best for each image to have the same height and width. For the annotations, the first two coordinates are the centroid, while the second two coordinates are the width and height. So the new second coordinates were modified from the cropped images from the original training set provided and calculated by subtracting $x_2 - x_1$ and $y_2 - y_1$ and the first by adding half of that to the original, then they are divided by the height and width of the image.

The following calculation were used to generate the new coordinates for x and y:

- $x = (x_1 + (x_2 - x_1) \cdot 1/2) \cdot 1/w$
- $y = (y_1 + (y_2 - y_1) \cdot 1/2) \cdot 1/h$

3.3.2 Accuracy Calculation

The score for the tests here was calculated using the F-score, in the same way as the contestants. According to the contest evaluation criteria, a correct detection (true positive) is the one that lies within 32 pixels from this centroid of the ground truth mitosis. This is a harmonic mean of precision and recall (sensitivity), as described below.

$$F - score = \frac{2 \cdot (precision \cdot sensitivity)}{(precision + sensitivity)} \quad (1)$$

The precision measures how accurate the predictions are using the percentage of the correct predictions out of the total. It is calculated using the FP which represents the number of false positive predictions, and TP which is the number of true positive predictions, as shown below:

$$Precision = \frac{TP}{FP + TP} \quad (2)$$

The recall measures how well all the positives are found in the test set, where FN is the number of false negatives (those ground truths which were not detected), as shown below

$$Recall = \frac{TP}{FN + TP} \quad (3)$$

3.4 Software & Hardware

Google Colab cloud was used for the GPU access. The architecture is limited to NVIDIA P100 or T4, with RAM to 25 GB.

3.4.1 YOLOv5 and YOLOv3

The YOLOv5 [7] implementation used is written in the Ultralytics framework [6]. The repository also contains the model parameters and layers for the YOLOv3 network.

3.4.2 YOLOv4-Scaled

The official implementation of YOLOv4-Scaled [30] makes use of the Pytorch framework. Yolov4-csp from the yolov4-large branch for cloud GPU was used.

3.4.3 YOLOR

The official implementation of YOLOR [31] is on Github. The *yolor-p6.cfg* was used.

4 Results

4.0.1 Results of Data Augmentation

The Table 1 below shows some of the different augmentation techniques which were applied to the dataset and compared to the test without augmentation. The *Augmentation Type* column is the type of augmentation applied described above.

Augmentation Type	ICPR 2014 F-score	ICPR 2012 F-score
None	0.77	0.67
Exposure	0.94	0.96
Brightness	0.94	0.94
Blur	0.92	0.93

Table 1: Tests with YOLOR with Different Data Augmentation Techniques Applied to each Dataset

For each of the tests, the YOLOR model was trained for 120 epochs and a batch size of 8. There is a significant increase in the score for the dataset with any image augmentation that was tested. Over 3 trials on ICPR 2014, each type, mosaic blur, rotation, noise produced an F-score of 0.92, while a combination of techniques around 0.94. Exposure (and brightness) (changes of +/- 25 %) were consistently the highest on both datasets. For further testing only one augmentation technique was applied to the datasets since the combination of multiple augmentations did not significantly effect the results, besides increasing the training time.

Further, when no augmentation was applied and the training time was increased to the same amount as all of the other tests (the number of epochs were doubled), the F-score was still not as high as it was with augmentation; it only increased to 0.87 and 0.86 for ICPR 2012 and ICPR 2014, respectively. The training time for ICPR 2012 was around 0.32 hours for each test with augmented data, while the training time was around 1.4 hours for ICPR 2014.

4.0.2 Models & Versions

The YOLO version/model, and combined training and testing runtimes (in hours), are shown in the Model and Time columns, respectively, of the tables below. 120 epochs were evaluated in order to maximize the F-score.

YOLOv5 Each preset model scale and size of YOLOv5 was tested with a batch size of 16 for both the p5 and p6 versions, but the p5 version performed better. The resulting scores with each of the different model scales are shown in Tables 2 and 3 for ICPR 2014 and 2012 datasets, respectively.

Overall the *m* model consistently had a slightly higher F-score. For ICPR 2014 and 2012, F-scores of up to 0.95 and 0.96, respectively, were achieved using YOLOv5m with the augmented training data when trained for 120 epochs. The version *l* and *x* required far longer runtime without an increase in F-score for both datasets.

YOLOv4-Scaled Table 4 shows the tests with the scaled YOLOv4-csp with a batch size of 16 and a range of numbers of epochs. For the ICPR 2014 dataset it takes around 60 epochs for the F-score to reach its highest F-score of around 0.93. The runtime was similar than YOLOv5m for a lower F-score. However, it produced faster speed and higher accuracy than YOLOv3.

Model	Time (hrs)	Precision	Recall	F-score
YOLOv5s	1.52	0.93	0.95	0.94
YOLOv5m	1.53	0.95	0.95	0.95
YOLOv5l	2.29	0.90	0.94	0.92
YOLOv5x	4.09	0.95	0.94	0.94

Table 2: Tests with YOLOv5-p5 for 120 Epochs with ICPR 2014

Model	Time (hrs)	Precision	Recall	F-score
YOLOv5s	0.18	0.9	1.0	0.94
YOLOv5m	0.26	0.96	0.96	0.96
YOLOv5l	0.50	0.96	0.96	0.96
YOLOv5x	0.79	0.92	0.96	0.94

Table 3: Tests with YOLOv5-p5 for 120 Epochs with ICPR 2012

Epochs	Time (hrs)	Precision	Recall	F-score
20	0.49	0.90	0.84	0.87
40	0.95	0.89	0.93	0.91
60	1.44	0.90	0.95	0.93
80	1.91	0.89	0.95	0.92

Table 4: Tests with YOLOv4-Scaled with ICPR 2014

Epochs	Time (hrs)	Precision	Recall	F-score
20	1.35	0.91	0.77	0.84
60	1.63	0.90	0.92	0.91
120	2.69	0.90	0.92	0.91

Table 5: Tests with YOLOv3 with ICPR 2014

Epochs	Time (hrs)	Precision	Recall	F-score
20	0.09	0.82	1.0	0.90
60	0.25	0.82	1.0	0.92
120	0.50	0.81	1.0	0.91

Table 6: Tests with YOLOv3 with ICPR 2012

YOLOv3 Table 5 and Table 6 shows the tests with the scaled YOLOv3 model with a batch size of 16 and a range of numbers of epochs for each dateset. It takes around 60 epochs for the F-scores to reach their highest of around 0.91 and 0.92 for ICPR 2014 and 2012, respectively. Not only is the F-score much lower, but the runtime is much longer than the YOLOv5 and YOLOv4-scaled models for both datasets.

Epochs	Time (hrs)	Precision	Recall	F-score
30	0.84	0.94	0.91	0.93
60	1.34	0.91	0.89	0.90
120	2.64	0.92	1.0	0.92

Table 7: Tests with YOLOR on ICPR 2014

Epochs	Time (hrs)	Precision	Recall	F-score
30	0.16	0.92	0.92	0.92
60	0.32	0.97	0.93	0.95
120	0.65	0.93	0.94	0.94

Table 8: Tests with YOLOR on ICPR 2012

Time (hrs)	Precision	Recall	F-score
0.18	0.97	0.94	0.96

Table 9: Tests with YOLOv5-p5 combined with YOLOR for 30 Epochs with ICPR 2012

YOLOR Table 7 shows the tests with YOLOR when trained with increasing numbers of epochs, using a batch size of 8. This model has lower runtime and a higher F-score for any number of epochs. Only 30 epochs are required to reach the highest F-score for the model of 0.93.

Table 8 shows the tests with YOLOR when trained with increasing numbers of epochs, using a batch size of 8 for ICPR 2012. On this dataset this model provides a similar runtime and F-score to YOLOv5.

4.0.3 Combining YOLOv5m-p5 with YOLOR

The YOLOv5m-p5 model with YOLOR model run in parallel on separate GPUs at the same time, for only 30 epochs. Since the predictions made with YOLOv5m-p5 and with YOLOR were both very high yet both had different predictions, the combination of predictions was used to produce consistently the highest final scores and lowest runtimes (over multiple tests), as shown in Table 9. For example, YOLOv5m-p5 helped to eliminate false positives predicted by YOLOR, resulting in a higher precision than YOLOR alone and a lower runtime.

4.0.4 Combining Both Datasets

The Tables 10 and 11 below show the results of combining the ICPR 2012 and 2014 training datasets. For each test, YOLOR was trained for both 60 and 120 epochs with a batch size of 8.

By combining the training sets, some of the highest F-scores were obtained on both the ICPR 2012 test set and the 2014 test set. The runtime for training the ICPR 2014 dataset with YOLOR was also the lowest with the highest F-score. Although the training time was much higher, the highest F-score was obtained on the ICPR 2012 test dataset.

Epochs	Time (hrs)	Precision	Recall	F-score
60	0.44	0.93	0.96	0.94
120	0.87	0.90	0.98	0.94

Table 10: Tests with Combined Training Sets and YOLOR on ICPR 2014 Test Set

Epochs	Time (hrs)	Precision	Recall	F-score
60	1.5	0.93	0.96	0.92
120	3.2	0.89	1.0	0.96

Table 11: Tests with Combined Training Sets and YOLOR on ICPR 2012 Test Set

Train Dataset	Test Dataset	F-score
Aperio & Hamamatsu	Aperio	0.94
Aperio	Aperio	0.94
Aperio	Hamamatsu	0.81
Aperio & Hamamatsu	Hamamatsu	0.91
Hamamatsu	Hamamatsu	0.95
Hamamatsu	Aperio	0.95

Table 12: YOLOR with Different Combinations of Scanners for Train and Test Datasets ICPR 2014

4.0.5 Adding the Data from Another Scanner

As shown in the results in the Table 12 below, adding the Hamamatsu Nanozoomer 2.0-HT slide scanner data to the training set did not help in prediction in the tests on the Aperio Scanscope XT scanner data. However, when the training set consisted of the Hamamatsu combined with the Aperio or just consisted of the Hamamatsu the network predicted Hamamatsu scanner dataset alone, better. Interestingly, the network predicted the Hamamatsu slide scanner test set best when the Aperio data was removed from the training set. Therefore, when the network was trained on both scanner data it was not able to better predict the images from the test set consisting of one scanner alone. Adding the data from another scanner to the training set also significantly increases the runtime for training.

Interestingly, the Hamamatsu *only* training set helped to predict both the Aperio test set alone and the Hamamatsu alone test set the best, but only when it was trained with the Aperio images excluded. The Hamamatsu *only* training set actually produced a very slight increase in the F-score by 0.01.

5 Discussion

5.0.1 YOLOv3

It took the YOLOv3 model over an hour and a half to train to reach a high F-score of 0.91 on ICPR 2014, while other models reached up to 0.95. It also took longer and only reached 0.92 with ICPR 2012, which had a high of 0.96.

5.0.2 YOLOv3 vs. Newer Models

The neck of YOLOv4 and YOLOv5 is a PANet (Path Aggregation Network) which uses a more advanced technique called *path aggregation* to help preserve more of the spatial information in instance segmentation [13]. Since the complexity of the features in a CNN increases as the image passes through the network as spatial resolution of the image decreases, the pixel-level feature masks are extracted in layers far from the the deeper layers of the network.

On the other hand, an FPN is used in YOLOv3. This uses a top-down path through the CNN layers to extract and combine the semantically rich features with the precise localization information. This can be time-consuming for large objects or large networks because the information must be passed on through hundreds of layers. Wherefore, PANet takes a short-cut connection from both a bottom-up path as well as the top-down path originally taken by FPN. This makes for clean short cut paths from upper to lower layers, which are only around ten layers.

5.0.3 YOLOR

YOLOR [32] and YOLOv5 came extremely close in runtime and accuracy but YOLOv5 was not quite as fast. YOLOR improves upon the former models with an unified network architecture which combines the implicit and explicit knowledge in order to optimize the Kernel Space Alignment, multi-task learning, and prediction refinement for learning implicit features [32].

5.0.4 YOLOR & YOLOv5m-p5 in Parallel

Since both of these models achieved the top scores in the shortest runtimes, running them in parallel was superior. Both consistently produced the highest scores since each predicted slightly different. The YOLOv5m-p5 model helped to increase precision in YOLOR by eliminating false-positives.

5.0.5 Combining Datasets

The highest F-scores on the test sets were obtained by combining the two datasets, which proves the real-life potential for the use of deep learning frameworks for mitosis counting. If continuously adding data from other databases helps improve the prediction accuracy on any given test dataset, we would be able to keep updating the model weights by training or fine-tuning with new datasets for better results. The more variety in the training examples there are, the better the network learns the features of mitosis and is able to adapt to the slightly different features contained in the test set.

5.0.6 Combining Scanner Data

More tests need to be run with combinations of scanner data, since it was not necessarily the case that the combination produced better results than one alone. For example, better predictions were made with the Hamamatsu scanner data for each test.

5.0.7 Data Augmentation

For ICPR 2014 and 2012, the augmented data helped to increase the F-score by 0.17 and 0.28, respectively. Note, that when the original dataset (without augmentation) training time was increased to the same training time as adding the augmented data (eg. by doubling the number of epochs) the F-score only increased by about 0.1 (for each dataset). Therefore, the data augmentation up-front was critical to obtaining the very high F-score.

Differences in Augmentation Types The difference between the results of augmentation types is likely due to the fact that, YOLOv4 and on introduced new data augmentation techniques. Possibly, the combination of two mosaic augmentation applications compounded upon one another reduced the networks ability to learn from those examples because the region of interest became to small for the network parameters. For example, four images would have became eight different images, so the part of the bounding box would be cut off from most of the images.

The exposure (and brightness) likely had a slightly *better* result because HPF slides often have these types of variations in real-life. Parameters that can contribute to discrepancy in the representation of the HPF include scanner optics, camera sensors, and digital resolution, scan resolution, image viewer, monitor size, aspect ratio, and display resolution [8].

5.1 Comparison to Other Models

Accuracy The F-score is achieved is significantly higher than contest winners score of 0.356. Additionally, Table 13 shows a comparison to some of other top-performing groups. The YOLO-based model also achieved at least as high of an F-score as others who have trained and tested their models on test sets extracted from the ICPR public training dataset. [18]

Model	F-score	Inference Time (s)
CasNN [2]	0.482	4.62
Lightweight RCNN [12]	0.427	0.83
DeepMitosis (DeepDet+Seg+Ver) [11]	0.437	0.72
FRCNN [18]	0.503	0.58
MS-FRCNN [34]	0.507	0.55
Cascaded w/ U-net [14]	0.576	-
Faster R-CNN and deep CNNs [15]	0.691	-
Deep Cascaded + HC [24]	0.900	0.300
YOLOv5/R	0.950	0.110
MITOS-RCNN [18]	0.955	0.500

Table 13: Performance Comparison on ICPR 2014 Test Set

and [24] both trained their model on all 3 datasets, so the resulting F-scores are not necessarily comparable.

Further, most of the top performers in literature (such as those listed below) either used a version of a regional CNN (eg. RCNN) or a deep cascaded network (as the contest winners did).

Time Analysis The RCNN-based models above require multiple hours to train [34]. For example, the Lightweight RCNN approach still requires 11.4 hours to train. Without multiple GPUs some frameworks would even be infeasible, such as, [18] which requires 5 Tesla NVIDIA K80 GPUs and 3 parameter servers. This is better than the fully CNN-based approaches initially proposed in the ICPR contests which required days to weeks to train even with a GPU [4]. However, neither type of framework is appropriate for clinical use. On the other hand, the YOLO-based real-time model only takes 15-30 minutes to train on ICPR 2012 and around and hour on ICPR 2014. Not only is the training time far shorter than other models, but the inference time per full HPF is only about 0.11 which is significantly shorter than other models as shown in Table 13.

YOLO Models vs. RCNN Models YOLO [19], which stands for *You Only Look Once*), combines the CNN used to predict the bounding boxes and the class probabilities for each box, rather than separating the two (as in RCNN). Further, the later versions of YOLO used here improve further by including innovations such as Cross-Scaled-Partial (CSP) connections [30], a Path Aggregation Network (PANet) [13], and optimized Data Augmentation. Hence, this model can be many times faster than Faster RCNN, while still maintaining accuracy.

6 Conclusion

In this paper, YOLO-based [20] models were tested as a tool for mitosis counting. Multiple models and versions of YOLO were compared with different types of augmentation, and then top models were run in parallel for superior results. The model out-performed all earlier methods on the ICPR contest datasets when YOLOR [32] was run in parallel with YOLOv5m-p5 on two separate cloud GPUs with exposure augmentations added and the results were averaged. Additionally, it took a fraction of the time for both training and testing, making it clinically applicable.

REFERENCES

[1] Akinfenwa. Atanda, Mohammed. Imam, Ali. Umar, Ibrahim. Yusuf, and Shamsu. Bello. Audit of nottingham system grades assigned to breast cancer cases in a Teaching Hospital. *Annals of Tropical Pathology*, 8(2):104–107, 2017.

[2] Hao Chen, Qi Dou, Xi Wang, Jing Qin, and Pheng-Ann Heng. Mitosis detection in breast cancer histology images via deep cascaded networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 1160–1166. AAAI Press, 2016.

[3] Dan C. Ciresan, Alessandro Giusti, Luca Maria Gambardella, and Jürgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 16 Pt 2:411–8, 2013.

[4] Dan CireÅan, Alessandro Giusti, Luca Maria Gambardella, and JÃ¼rgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. volume 16, pages 411–8, 09 2013.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[6] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, Ayush Chaurasia, TaoXie, Liu Changyu, Abhiram V, Laughing, tkianai, yxNONG, Adam Hogan, lorenzomammana, AlexWang1900, Jan Hajek, Laurentiu Diaconu, Marc, Yonghye Kwon, oleg, wanghaoyang0106, Yann Defretin, Aditya Lohia, ml5ah, Ben Milanko, Benjamin Fineran, Daniel Khromov, Ding Yiwei, Doug, Durgesh, and Francisco Ingham. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations, April 2021.

[7] Glenn Jocher, Alex Stoken, Ayush Chaurasia, Jirka Borovec, NanoCode012, TaoXie, Yonghye Kwon, Kalen Michael, Liu Changyu, Jiacong Fang, Abhiram V, Laughing, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Jebastin Nadar, imyhxy, Lorenzo Mammana, AlexWang1900, Cristi Fati, Diego Montes, Jan Hajek, Laurentiu Diaconu, Mai Thanh Minh, Marc, albinxavi, fatih, oleg, and wanghaoyang0106. ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support, October 2021.

[8] David Kim, Liron Pantanowitz, Peter SchÃ¼ffler, DigVijay Yarlagadda, Orly Ardon, VictorE Reuter, Meera Hameed, DavidS Klimstra, and MatthewG Hanna. (re) defining the high-power field for digital pathology. *Journal of Pathology Informatics*, 11:33, 10 2020.

[9] Daisuke Komura and Shumpei Ishikawa. Machine learning methods for histopathological image analysis. *Computational and Structural Biotechnology Journal*, 16:34 – 42, 2018.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097â1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[11] Chao Li, Xinggang Wang, Wenyu Liu, and Longin Jan Latecki. Deepmitosis: Mitosis detection via deep detection, verification and segmentation networks. *Medical Image Analysis*, 45:121 – 133, 2018.

[12] Yuguang Li, Ezgi Mercan, Stevan Knezevitch, Joann G. Elmore, and Linda G. Shapiro. Efficient and accurate mitosis detection - a lightweight rcnn approach. In *ICPRAM*, 2018.

[13] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. *CoRR*, abs/1803.01534, 2018.

[14] Xi Lu, Zejun You, Miaomiao Sun, Jing Wu, and Zhihong Zhang. Breast cancer mitotic cell detection using cascade convolutional neural network with u-net. *Mathematical Biosciences and Engineering*, 18:673–695, 04 2021.

[15] Tahir Mahmood, Muhammad Arsalan, Muhammad Owais, Min Beom Lee, and Kang Ryoung Park. Artificial intelligence-based mitosis detection in breast cancer histopathology images using faster r-cnn and deep cnns. *Journal of Clinical Medicine*, 9(3), 2020.

[16] Xipeng Pan, Yinghua Lu, Rushi Lan, Zhenbing Liu, Zujun Qin, Huadeng Wang, and Zaiyi Liu. Mitosis detection techniques in h&e stained breast cancer pathological images: A comprehensive review. *Computers & Electrical Engineering*, 91:107038, 2021.

[17] Emad Rakha, Jorge Reis-Filho, Frederick Baehner, David Dabbs, Thomas Decker, Vincenzo Eusebi, Stephen Fox, Shu Ichihara, Jocelyne Jacquemier, Sr Lakhani, JosÃ© Palacios, Andrea Richardson, Stuart Schnitt, Fernando Schmitt, Puay-Hoon Tan, Gary Tse, Sunil Badve, and Ian Ellis. Breast cancer prognostic classification in the molecular era: the role of histological grade. *Breast cancer research : BCR*, 12:207, 07 2010.

[18] Siddhant Rao. MITOS-RCNN: A novel approach to mitotic figure detection in breast cancer histopathology images using region based convolutional neural networks. *CoRR*, abs/1807.01788, 2018.

[19] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[20] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[21] L. Roux. Mitos-atypia-14 - grand challenge, 2014.

[22] Ludovic. Roux, Daniel. Racoceanu, Nicolas. LomÃ©nie, Maria. Kulikova, Humayun. Irshad, Jacques. Klossa, FrÃ©dÃ©rique. Capron, Catherine. Genestie, Gilles. Naour, and Metin. Gurcan. Mitosis detection in breast cancer histological images An ICPR 2012 contest. *Journal of Pathology Informatics*, 4(1):8, 2013.

[23] Monjoy Saha, Chandan Chakraborty, and Daniel Racoceanu. Efficient deep learning model for mitosis detection using breast histopathology images. *Computerized Medical Imaging and Graphics*, 64, 12 2017.

[24] Monjoy Saha, Chandan Chakraborty, and Daniel Racoceanu. Efficient deep learning model for mitosis detection using breast histopathology images. *Computerized Medical Imaging and Graphics*, 64, 12 2017.

[25] Meriem Sebai, Xinggang Wang, and Tianjiang Wang. Maskmitosis: a deep learning framework for fully supervised, weakly supervised, and unsupervised mitosis detection in histopathology images. *Medical & Biological Engineering & Computing*, 58, 05 2020.

[26] Nida Shahid, Tim Rappon, and Whitney Berta. Applications of artificial neural networks in health care organizational decision-making: A scoping review. *PLOS ONE*, 14:1–22, 02 2019.

[27] Mitko Veta, Yujing J. Heng, Nikolas Stathonikos, Babak Ehteshami Bejnordi, Francisco Beca, Thomas Wollmann, Karl Rohr, Manan A. Shah, Dayong Wang, Mikaël Rousson, Martin Hedlund, David Tellez, Francesco Ciompi, Erwan Zerhouni, David Lanyi, Matheus Palhares Viana, Vassili A. Kovalev, Vitali Liauchuk, Hady Ahmady Phoulady, Talha Qaiser, Simon Graham, Nasir M. Rajpoot, Erik Sjöblom, Jesper Molin, Kyunghyun Paeng, Sangheum Hwang, Sunggyun Park, Zhipeng Jia, Eric I-Chao Chang, Yan Xu, Andrew H. Beck, Paul J. van Diest, and Josien P. W. Pluim. Predicting breast tumor proliferation from wholeâslide images: The tupac16 challenge. *Medical Image Analysis*, 54:111â121, 2019.

[28] Mitko Veta, Paul J. van Diest, Mehdi Jiwa, Shaimaa Al-Janabi, and Josien P. W. Pluim. Mitosis counting in breast cancer: Object-level interobserver agreement and comparison to an automatic method. *PLoS ONE*, 11(8), 8 2016.

[29] Mitko Veta, Paul J. van Diest, Stefan M. Willems, Haibo Wang, Anant Madabhushi, Angel Cruz-Roa, Fabio A. González, Anders Boesen Lindbo Larsen, Jacob S. Vestergaard, Anders B. Dahl, Dan C. Ciresan, Jürgen Schmidhuber, Alessandro Giusti, Luca Maria Gambardella, F. Boray Tek, Thomas Walter, Ching-Wei Wang, Satoshi Kondo, Bogdan J. Matuszewski, Frédéric Precioso, Violet Snell, Josef Kittler, Teófilo Emídio de Campos, Adnan Mujahid Khan, Nasir M. Rajpoot, Evdokia Arkoumani, Miangela M. Lacle, Max A. Viergever, and Josien P. W. Pluim. Assessment of algorithms for mitosis detection in breast cancer histopathology images. *CoRR*, abs/1411.5825, 2014.

[30] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-YOLOv4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13029–13038, June 2021.

[31] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks. *arXiv preprint arXiv:2105.04206*, 2021.

[32] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks. *CoRR*, abs/2105.04206, 2021.

[33] Haibo Wang, Angel Cruz-Roa, Ajay Basavanhally, Hannah Gilmore, Natalie Shih, Mike Feldman, John Tomaszewski, Fabio GonzÃ¡lez, and Anant Madabhushi. Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features. *Journal of Medical Imaging*, 1:1–8, 12 2014.

[34] Robin Elizabeth Yancey. Multi-stream faster rcnn for mitosis counting in breast cancer images, 2020.

A Physical Device to Help the Visually Impaired Read Money Using AI/Machine Learning in Third World Countries

Nidhi Mathihalli
Saratoga High School
20300 Herriman Ave
USA 95070, Saratoga,
California
nidhi@mathihalli.com

ABSTRACT

There are over 285 million blind people in the world, with approximately 87% of them living in developing countries. However, in the third world countries, there is currently very little technology to help the visually impaired, especially with financial independence. In this article we present the machine learning algorithms used to develop the device to help visually impaired distinguish between different forms of currency. Using the various currency images, we form a data set that is used to train the transfer learning model. Experimental results show over 94% accuracy with transfer learning model. The device is designed to be portable and hand-held. The device can distinguish between 1, 5, 10, and 20 dollar currency bills. Additionally, the model can work offline. Overall, the device is cost effective, portable, and can be used in the absence of internet connectivity.

Keywords
Affordable Bill Detector, Effective Money Reader, Computer Vision-based Accurate Algorithms

1 INTRODUCTION

Currently, there are at least 285 million people in the world who are legally blind. Of these people, 87% of them live in third world countries. Thus, around 247 million people live in these developing countries. However, there is still very little technology to help them navigate their world.

Age-related vision loss such as with conditions like Macular Degeneration, are on a rising trend in developing countries. The numbers seem to parallel the trend in developed countries such as America. Specifically, their financial independence is restricted, since they are often not able to distinguish between different dollar bill denominations anymore. There is also the possibility of them being a victim of fraud. Cashiers and vendors may take advantage of their loss of sight and overcharge them by not giving them the correct change back.

Additionally, sudden loss of eyesight is proven to make one less confident and can result in the development of mental health issues, such as depression and anxiety. With cases on the rise, it is even more important to help the visually impaired become more financially independent. Online shopping is a far reach in third world countries where internet is a luxury. Everyday routine is drastically different from the status quo because of visual impairment.

Visually impaired people have trouble completing everyday tasks: reading the news, going to school/work, cooking meals, or making purchases at various stores. This solution assists the visually impaired by reading dollar bills, improving the shopping experience, in turn boosting social confidence with peers/friends.

2 BACKGROUND AND RELATED WORKS

Our initial research was conducted at the Shree Ramana Maharishi Academy for the Blind, in Bangalore, India. This school helps visually impaired orphans and kids from very poor families by giving them a free education. Meeting groups of students on a regular cadence, helped decipher what would be a practical design for the product.

The survey results from the students, who were aware of the purposes of these questions and responses, revealed that 100% wanted to have more financial capabilities through the use of this proposed device, and that 75% would rather have a physical device than a smartphone app. Most students expressed that a device such as a snap on device on sunglasses or a device hung by a lanyard over the neck would be a preferred choice. A physical device was clearly the choice, and there are two reasons for this - smartphones are expensive and there are challenges on using a device with a small screen for the visually impaired.

Over the course of a month, we gathered information on what types of technology should be considered keeping in mind easy of use and cost effectiveness. Researching on available current solutions revealed a few mobile

https://www.doi.org/10.24132/CSRN.3201.33

apps, which as stated previously, are not effective since their small screens make them very inaccessible by the visually impaired. Other solutions include the iBill, and the OrCam, however, these devices cost over $115. The iBill is a handheld device that identifies the denomination of the bill by taking a picture of any corner of the bill [1]. The OrCam is another handheld identifier, used for reading text, identifying products, etc [2].

According to a 2016 report by the National Sample Survey Office by the Government of India [3], the rural farmer's monthly salary in the third world country India, is 6,426 rupees per month, or $86.67. With these salaries, affordability for such physical devices costing over $120 is out of question.

Keeping all the above in mind, it is evident that the visually impaired in third world countries prefer an easy-to-use physical money detection device that is accurate yet cheap. Research on cost effective hardware materials led to a conclusion that the raspberry pi, (a mini computer priced at $20), an ESP32 Cam Module (priced at $4), and a Battery Breakout board (priced at $0.5) would be an ideal combination. This brings the total price to $25, which is a 79.16% decrease in price compared to the current cheapest device.

In conclusion, the main goal was to create the money reader in an affordable, working, and effective way, with the key metrics being that the validity accuracy of the bill is over 90%, the time taken to predict the denomination is under 10 seconds, and the total cost of the device is under $30.

3 MATERIALS AND METHODS

Using the previously established goal for this project, we split the approach into multiple sections. Each section illustrates a different method we adopted to solve certain aspects dealt with creating the device. The sections either document the approaches we took to create/improve the machine learning model or build the hardware component. All code for this project can be found in this project's GitHub link [4].

3.1 Machine Learning

Please use a 10-point Times Roman font, or other Roman font with serifs, as close as possible in appearance to Times Roman in which these guidelines have been set. The goal is to have a 10-point text, as you see here. Please use sans-serif or non-proportional fonts only for special purposes, such as distinguishing source code text. If Times Roman is not available, try the font named Computer Modern Roman. On a Macintosh, use the font named Times. Right margins should be justified, not ragged.

Before creating each of our approaches, we decided on using machine learning as the identifier. Machine learning is an application of Artificial Intelligence that provides computers with the ability to automatically learn and improve from experience without having to program every possible case. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. In Machine Learning, the primary aim is to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly. Machine Learning programs complete their tasks using algorithms. Algorithms are sets of rules that the computer follows in calculating operations. In machine learning, there are four types of algorithms: supervised, semi-supervised, reinforced training, and unsupervised.[5]

Supervised machine learning algorithms use past images to predict what each new image is. Using a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly. Supervised machine learning uses two main processes: Classification and Regression. Classification is the process of predicting what a group of images is representing. Regression is the measure of the relation between the values of one variable group to corresponding values of another variable group. Using these two procedures, computers use supervised machine learning to classify images.[5]

In Unsupervised Machine Learning, one trains images without any labels. People use unsupervised machine learning techniques for clustering, detecting anomalies, association mining, and for creating latent variable models. In clustering, a machine splits the images into groups, although they might be incorrect. Anomaly detection is used to figure out otherwise unrecognizable patterns or details. Therefore this type of machine learning is used a lot in finding out if a fraudulent transaction has occurred, or if there is an outlier among some data points. Association mining is when machine groups together certain objects that are similar or of the same use. This type of machine learning is used for retail marketing or on online shopping Websites in order to show users what is of a similar type as to the item the customer is currently looking at. Finally, unsupervised machine learning is used in creating latent variable models. Latent variable models are machine learning models that relate observable details to latent, or hidden, details.[5]

Semi-supervised machine learning combines both unsupervised machine learning as well as supervised machine learning. These datasets train using both labeled and unlabeled images, usually more unlabeled images. Programmers use this kind of machine learning in order to save time on labeling images. Furthermore, labeling too many images can impose a human bias on the machine. Therefore, by using semi-supervised ma-

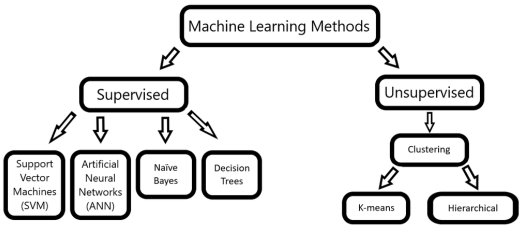


Figure 1: This image shows us the pipeline used for supervised versus unsupervised learning. Both of these algorithms were used throughout our report. [6]

chine learning, one saves time, doesnât create a human inclination, and makes sure that the images are being labeled properly.[5]

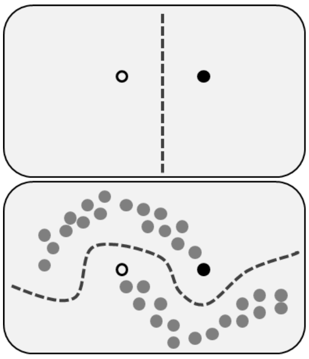


Figure 2: This figure shows us how semi-Supervised learning works, as there is both labeled and unlabeled data that helps the algorithm accurately sort the given data.[7]

Reinforcement learning is when a machine learns from trial and error. In this form of machine learning, the machine gets feedback from its actions and experiences. This sort of machine learning can be analogized to a game. In this game, the creator gives the model no hints as to how to solve the game. The model has to 1) figure out how to play the game, and 2) sort each image correctly, in order to maximize the score of the game. Since reinforcement learning is a new idea, it is currently not being used as an application today. However, its creators are planning to use it in the future for assisting humans, as it is an AGI, artificial general intelligence, or as a method in figuring out the consequences of different strategies.[5]

3.2 Google Vision API

For our first machine learning attempt, we used the Google Vision's Auto ML API. The initial dataset had 450 images of various denominations. After training the model, we uploaded a few images in order to test if those images would be evaluated correctly. When we used the model to predict the images on Auto ML, the model predicted the images accurately. The next step was to create a programmatic version for prediction using AutoML. We programmed the software, building off of the code outlined by AutoML API. To complete the code, we used the Auto ML modelâs id, the project id, and the images to create the final program. Additionally, we added the ability to programmatically capture the images of the bills.

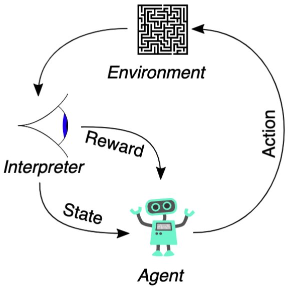


Figure 3: This figure shows us how reinforcement learning is able to analyze past results in order to learn from trial and error.[8]

3.3 Base Model

The base model is a native Tensorflow model. We used Tensorflow's “Convolutional Neural Network" Google Colab starter code [9]. This starter code looks at flower classification, and uses the CIFAR-10 data set. In order to use this starter code, we had to convert the images from the dataset into the format that the CIFAR-10 data set [10] uses. This was done by adding each image to one of two binary file, each respectively containing the training and testing data, and added information on the pixels and their colors. We then loaded the data into x_train, y_train, x_test, and y_test. We defined x_train and x_test as the image data, and y_train and y_test as the image labels.

3.3.1 *Data Augmentation*

Data augmentation is a method of adding more training data to the model by slightly altering the images. This can include rotating, reflecting, and cropping the image. Our goal of using data augmentation was for the addition of data for our model through the different image variations. By adding more data, the model was able to identify and classify the image under different image conditions.

3.4 Binary Classification

Binary classification is used to identify whether an image represents a certain set object or not.

Using this type of model, we were be able to predict whether a bill is a 1 dollar bill or not. Similarly, we can do the same for figuring out whether it is a 5 dollar bill or not, 10 dollar bill or not, and if it is a 20 dollar bill or not and so on. The result is binary - either a yes or a no. Binary classification took less necessary images and layers per model, but leads to better results, since the output is one of 2 values rather than 4 ($1, $5, $10, and $20).

3.5 Filtering Data

Filtering Data was another approach we experimented with. We applied another layer of abstraction by applying different filters on the training data images before having the model train on them. Since dollar bills have distinct edges, we tried two cases, using the Box Filter [11], Contrast Filter, Sobel Filter, and Canny Filter. Figure 1 shows the original picture:

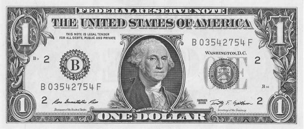


Figure 4: This figure is the original dollar bill upon which various filters were applied and contrasted in order to see if a specific filtered image gave a significant increase in accuracy

The Box Filter [11] is used to make images more blurry. It is a square array such that each element is $\frac{1}{numberoftotalelements}$. For a 5-by-5 Box Filter [11] array, each element will be $\frac{1}{25}$. Figure 2 shows the what Figure 1 looks like after the Box filter is applied to it:

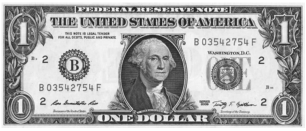


Figure 5: The Box Filter [11] made the image more blurry, making it easier to find large features

As shown in the Figure 2, this image is blurrier than the original one. Box Filters [11] are often used to point out big features. The other filter we experimented with was the Contrast Filter. Doing the opposite of the Box Filter, this filter is used to sharpen certain smaller details in the image. As the name suggests, Contrast Filter makes the image darker/lighter in certain parts to make smaller details contrast more. Figure 3 shows what the dollar bill looks like after the Contrast filter is applied to it:

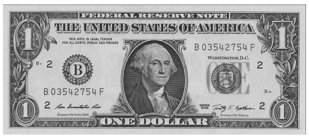


Figure 6: Contrast Filter slightly altered the image, but not very visibly

The third kind of filter is the Sobel Filter [12]. This filter is used to detect lines in an image. The Sobel Filter can be applied both vertically and horizontally. This brings out the lines in an image. Since lines are a big part of the features in dollar bills, we hoped that this would bring out the details that would help the model classify the denominations. Below are the pictures after applying the Sobel Filter. Figure 4 uses the Sobel Filter from the matrix we created while Figure 5 uses a Sobel Filter taken from the Skimage Library [12]:

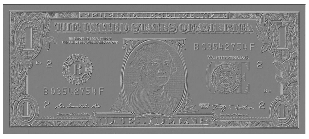


Figure 7: Sobel Filter used from matrix gave a textured feel to the image

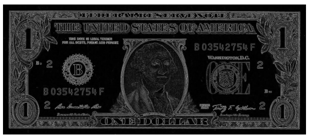


Figure 8: Sobel Filter used from Skimage Library gave similar results to that of the Canny Filter

The final filter we experimented with was the Canny Filter [13]. The Canny Filter [13] is another edge detection filter, which identifies a set of edges depending on a sigma value. The higher the sigma, the lower the resolution, make the features detected the larger ones, and vice versa. Figure 6 shows what the dollar bill looks like after the Canny Filter [13] is applied to it.

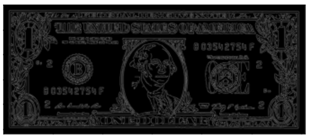


Figure 9: Canny Filter darkened the image, making the edges white thus greatly contrasting the two

3.6 KNN Feature Detection

The 5th algorithm we experimented with was the KNN algorithm [14]. The KNN algorithm [14] when applied to computer vision is was very useful when trying to find the most apparent features in the dollar bill.

The KNN algorithm [14] takes key points and values and finds where the key features are in each image and can easily translate one image to another. As shown in Figure 7, this is very useful in terms of dollar bill detection. This is because it can easily identify where these details are even if they're in different spots of the photo. For example if it sees an interesting curve at the top of a $1 bill, it will notice that same curve even if the dollar bill is flipped upside down and the curve is in a different location. Thus, KNN [14] enabled the classification of objects without having to feed the model augmented data. with few images, it is easy to identify if and how much an image is similar to another.

3.7 Transfer Learning

The final method of experimentation used transfer learning. Transfer learning is a broad field in machine

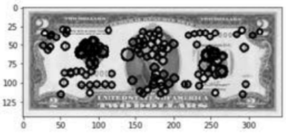


Figure 10: K-means feature detection using key points

learning and it is often used for all object classifications. This is because most transfer learning models work off of previously trained data heavy but very accurate models such as the Google Inception model [15], Microsoft ResNet model [16] and the MobileNet model [17]. Thus many applications will use these models to make an all object detection system. Since this was focused on dollar bills, we had to take off the last couple layers then feed in our own data and other specific information relevant to our application, but transfer learning helped create a really good model. Microsoft ResNet [16] is one of the most commonly used base models for transfer learning. Using ResNet [16] as the base, we removed the last couple of layers, added a Softmax layer with 4 labels, and trained the model. Since it was able to solely focus on dollar bills instead of other objects, this saved time, reducing the time it would normally take a ResNet model [16] to identify a dollar bill or any object.

3.8 Hardware Design

The initial design consisted of a Raspberry Pi and Pi Camera. The models were transferred to the Pi to see how much time it would take to recognize a picture that was taken with the Pi camera. The Raspberry Pi camera is very useful when working with the Raspberry Pi since a lot of the libraries that the Raspberry Pi camera uses are pre-installed with the Debian OS and hence this did not involve additional effort to install third party dependency libraries to process the picture or live stream.

However, we hit some roadblocks. We were originally using a Raspberry Pi 3. The downside of using transfer learning is that you have to import some libraries that are crucial to transfer learning. This included the Keras library and the Tensorflow library. These libraries take up a lot of memory. Additionally, these libraries cannot be installed in a 32-bit operating system.

Currently there are no known OS systems for the Raspberry Pi 3 that supports machine learning for the Raspberry Pi. However there is a beta version of an 64-bit operating system for the Raspberry Pi 4. This operating system did work, but because the OS was in its beta version, the libraries that the Raspberry Pi camera needed were not available. Thus we had to look for an alternate camera that was able to easily take pictures and send them to the Raspberry Pi.

3.9 WiFi Camera

The prognosis was that the Raspberry Pi unit will be a separate handheld device and will be a separate system from the camera itself. This was due to the fact that we didn't want something bulky on the camera, since it is meant to be wearable. Thus, we decided to make the camera and the minicomputer into separate systems.

The ESP32 camera is a Wi-Fi camera that is a part of the ESP32 modules. It is very useful for taking photos quickly, taking live streams and is often used with machine learning and AI projects. The ESP32 camera, however, works on Wi-Fi. Since the Raspberry Pi 4 can be made into an access point, we decided to make the minicomputer into an access point for the ESP32 camera. Thus, the final design comprised of the Raspberry Pi 4 and the ESP32. This allowed the ESP32 easily connect to the Wi-Fi on the Raspberry Pi and is able to take a picture then send it to the Raspberry Pi.

4 FIGURES/CAPTIONS

Place Tables/Figures/Images in text as close to the reference as possible (see Fig.**??**). It may extend across both columns to a maximum width of 16 cm (6.3"). Captions should be Times New Roman 10-points. They should be numbered (e.g., "Table 1" or "Figure 2"), please note that the word for Table and Figure are spelled out. Figure's and Table's captions should be centered beneath the image, picture or a table.

5 EXPERIMENTAL METHODOLOGY

While working with of our approaches, we needed to created a consistent training and testing plan in order to best compare the different approaches. To do so, we have noted below the experimental mythology that was used with each approach. We have defined below the 2 data sets that we created and defined our evaluation metrics.

5.1 Data set

In order to create accurate machine learning models, we created two data sets, each with varying levels of representative and accurate data.

5.1.1 Data set 1

In order to create the base model, named “Data set 1", we first had to collect the necessary data. Unlike many other classification problems, there were few data sets that had pictures of dollar bills and their labels. Thus, we decided to create our own data set. For the first data set created, we took pictures on our own. Asking others to take a couple pictures from their different points of view as well, we ended up with over 100 pictures for each label. Although this data set worked relatively well, we decided to add more data, since many of the

Approach Type	Accuracy Percent
Google Vision API	95%
Base Model	63.64%
Binary Classification	80-85%
Filtering Data	66-67%
KNN Feature Detection	95+%
Transfer Learning	94%

Table 1: All results from the various models in a tabulated format

pictures were taking at similar angles and with similar backgrounds, which could severely bias the model. A sample for this dataset is in this project's GitHub link [4].

5.1.2 Data set 2

For the second data set, named "Data set 2" (which was created after the "Binary Classification" [18] step), we decided to add more data by data scrapping images off of the Internet using PyPi's "google_images_download" API [19]. Through this, we were able to get a data set with over 200 images. Additionally, the pictures scrapped from online resources were able to provide a contrast to the hand-taken pictures in terms of quality, background light, etc. Thus, the accuracy measures greatly increased when using this data set versus the previous one.

5.2 Evaluation Metrics

To measure how accurate the data set is, we used the following metric to assess the performance of the model: Accuracy. The accuracy of a model measures the amount of true positives and true negatives over all data (consisting of all positives and negatives). For example, if a one dollar bill has been predicted to have 8 true positives, 2 false negatives, and 3 false positives, and 12 true negatives, the accuracy would be 80%. The accuracies for all approaches can be seen in Table 1, with further discussions on these results in the following explanations.

6 RESULTS AND DISCUSSION

Using the previously stated experimental methodology, we tested each stated approach and have listed the results derived. We used these results to discuss which approaches should be implemented in the final design.

6.1 Google Vision API

The results of the first model were not promising. We first tried using "Data set 1" as our training and testing data. Although the testing measures were not bad, at 81.818% recall and 88.235% precision, the validity measures were still not acceptable. When testing this model against 20 images, only 7 images were recognized correctly, while the remaining 13 are recognized incorrectly.

Next, we tried using "Data set 2". The results from this second model were better. Using the 20 test set pictures, this model had 19 images were recognized correctly, while only 1 was recognized incorrectly. Additionally, the recall and precision from this model were an improvement from the last model, as the recall was 97.895% and the precision was 96.875%, and both measurements of accuracy crossed 95%.

However, since working with AutoML requires Internet connectivity, if the user is not connected to the Internet, the money reader would stop working. Thus, we decided to abandon this approach since the device has to be connected to the internet in order to work or have its own public Wi-Fi.

6.2 Base Model

For this model, we trained the data on 2 Convolution 2D layers, and 3 Dense layers. We used data set "Data set 2" for our training and testing data. We also added a Softmax activation at the end. After training the model on 10 epochs, we got an accuracy of 63.64%. This was not acceptable for our purposes, so we had to find different ways to improve the model.

The data augmentation increased the accuracy, but not by enough. By doing this, we were able to increase the accuracy by 1-2%. This was mainly because there was not a lot of variety in the data, which led to the amount of data being increased, but the type of data staying relatively the same. Thus, the accuracy of the model barely increased.

However, the time that it took the Raspberry Pi to predict greatly increased to over a minute. Since the goal was to have a fast classification, specifically being that the Pi had predict the model within 10 seconds, this was not feasible.

Additionally, another source of error could be that the model was not specifically trained for the bill detection; rather, it was a generic model with a set type and number of layers.

6.3 Binary Classification

For the Binary Classification, we created four different models, each of which gave a binary response as to whether or not it was a certain type of denomination. "Data set 2" was used for the training and testing data, although it was formatted differently so to suit the different models. These 4 models had accuracy between 80-85%, which was an increase to the previous model. Additionally, by creating four threads and running them simultaneously, it would not take a lot of time to predict.

Although this worked initially, we were not able to transfer all 4 models to the Raspberry Pi. This was due to memory issues. Furthermore, when tested on a computer with enough computational power, the models would occasionally identify a certain dollar bill as both a 1 dollar bill and a 20 dollar bill. In these cases, it was very hard to figure out what the correct denomination was.

6.4 Filtering Data

We used “Data set 2" for our training and testing data. Although the filtered data was useful to the human eye since it was able to find many distinct features, it increased the accuracy by merely 2-3%. The main reason behind this was because many of the filters that we applied identified lines and features that were part of 2+ types of dollar bills rather than individual ones. Thus, although it was able to clearly identify where the dollar bill was in the photo, it had a hard time distinguishing between the types of dollar bills.

Additionally, if we were to feed this data into the model, we would have to apply these filters to the images while predicting. This could take a long time, especially with the Sobel and Canny filters, and since we want to reduce the amount of time taken to predict, this would not be feasible.

6.5 KNN Feature Detection

We used “Data set 2" for our training and testing data. Using KNN yielded the best results, with accuracies of over 95%. However, finding the features and comparing them takes a long time, often taking over 5 minutes per classification. Since the goal was to have the model predict the denomination under 10 seconds, this algorithm could not be used as well.

The source of error for this could be that a lot of the features that were recognized from the KNN feature detection were features that were common to all of the dollar bills. For example, it noticed the corners of each bill along with the words “Federal Reserve Bank." Although, this level of detail was useful, it also increased the time taken for determining the denomination.

6.6 Transfer Learning

We used “Data set 2" for our training and testing data. This model had an accuracy around 94% and this was only with three epochs. We decided against a higher epoch count because the more epochs we had would result in a model that was easily determine able to recognize its training and testing data, but would fail in the validation to data due to over fitting. Since ‘Data set 2" had only 100 images for each denomination, the amount of time for classification was greatly reduced. Currently, the model is able to perform in under 10 seconds to identify the dollar bill which is a vast contrast to the time the previous approaches offered. Due to the high accuracy and faster response time, this model was used for while testing the prototype.

6.7 Final Model and Results

We decided to use the transfer learning model. Through that, we were able to get a high accuracy prediction with little time taken. When testing 20 images taken, the model was able to get 100% accuracy. Additionally, the device did not need the use of Internet for it to work.

6.8 WiFi Camera

We decided to stick with the final product consisting of the Raspberry Pi and the ESP32 and a battery breakout board. First the ESP32 takes a picture and sends it to the Raspberry Pi. The Raspberry Pi then uses the transfer learning model, to predict what the denomination of the dollar bill is. After the prediction results are communicated through headphones. It is a very quick process, and due to the affordability of the materials, can be bought and used by anyone. Additionally, the accuracy of the model is over 94% and runs under the 10 second limitation , allowing users to quickly identify the denomination of the bill.

7 CONCLUSION

Overall, this application for determining the denomination of dollar bills is has high accuracy and is of need to the visually impaired, specifically in third world countries. However, there are some limitations. Due to the 94% accuracy, there is a chance that the wrong denomination could be identified. However, upon further analysis, we uncovered that this was only a problem in dark lighting conditions. However, this problem was mitigated when we programmed the transfer learning model to use colored images, the model was able to identify the portion of dollar bills we ran in the low light conditions correctly.

The next step is to to improve the accuracy of readings, especially at different angles. Additionally, we are working with the Shree Maharishi Academy for the Blind (Bangalore, India) in order to get 10 of these devices to them. Effort is also underway to make this device available on APH (American Print House) - a portal for products for the visually impaired.

We also plan to extend the use of this application to assist blind students in reading. Since the product can indeed read numbers on price tags and bills, we can also expand the use into reading words. This will help young, elementary school students to participate in class reading activities to build relationships with peers and teachers. By extending the use of reading, visually impaired adults will be able to read signs and other papers.

8 ACKNOWLEDGMENTS

Our thanks to Professor Anagha Kulkarni at UCSF and Makers Lab's Scott Drapeau for their mentorship on this project. We would also like to thank the Shree Ramana Maharishi Academy for letting us conduct our initial research there.

9 REFERENCES

[1] Research, O. iBill US Bank Note Reader. http://www.orbitresearch.com/product/ibill-talkingbanknote-identifier/.

[2] OrCam OrCam Read. https://www.orcam.com/en/

[3] Office, N. S. S. Income, expenditure, productive assets and indebtedness of agricultural households in india.

[4] Mathihalli, N. Github link for money reader project., https://github.com/nidhimath/moneyReader.

[5] Brownlee, J. 14 Different Types of Learning in Machine Learning. https://machinelearningmastery.com/typesof-learning-in-machine-learning/.

[6] CreightonMA, SarkarSaha Figure1A.png. https://upload.wikimedia.org/wikipedia/commons /5/52/Sarkar%26Saha_Figure1A.png

[7] Techerin, Example of unlabeled data in semisupervised learning.png

[8] Megajuice Reinforcement learning diagram.svg.

[9] authors, T. Convolutional neural network (cnn)., https://colab.research.google.com/github/ tensorflow/docs/blob/master/site/en/tutorials/ images/cnn.ipynb.

[10] Alex Krizhevsky, V. N. and Hinton, G. The cifar-10 dataset., https://www.cs.toronto.edu/ kriz/cifar.html.

[11] Bernardo Rodrigues Pires, J. M. F. M., Karanhaar Singh Approximating image filters with box filters.

[12] Image, S. Scikit Filters Library. https://scikitimage.org/docs/stable/api/skimage filtershtml.

[13] OpenCV Canny Edge Detection. https://docs.opencv.org/3.4/da/d22/tutorial py cannyhtml.

[14] Wikipedia k-nearest neighbors algorithm., https://en.wikipedia.org/wiki/Knearest neighbors algorithm.

[15] Google Advanced Guide to Inception v3 on Cloud TPU. https://cloud.google.com/tpu/docs/ inceptionv3advanced.

[16] Microsoft ResNet. https://docs.microsoft.com/enus/azure/machinelearning /componentreference/resnet.

[17] Keras MobileNet and MobileNetV2. https://keras.io/api/applications/mobilenet/.

[18] Wikipedia Binary classification. , https://en.wikipedia.org/wiki/Binary classification.

[19] PyPi google images download 2.8.0. https://pypi.org/project/google images download/.

Influence of the underwater environment in the procedural generation of marine alga *Asparagopsis Armata*

Nelson Rodrigues
FEUP
Porto, Portugal
up200705576@edu.fe.up.pt
https://orcid.org/0000-0002-0519-7151

António Augusto Sousa
FEUP / INESC TEC
Porto, Portugal
aas@fe.up.pt
https://orcid.org/0000-0002-9883-2686

Rui Rodrigues
FEUP / INESC TEC
Porto, Portugal
rui.rodrigues@fe.up.pt
https://orcid.org/0000-0003-4883-1375

António Coelho
FEUP / INESC TEC
Porto, Portugal
acoelho@fe.up.pt
https://orcid.org/0000-0001-7949-2877

ABSTRACT

Content generation is a heavy task in virtual worlds design. Procedural content generation techniques aim to agile this process by automating the 3D modelling with some degree of parametrisation. The novelty of this work is the procedural generation of the marine alga (*Asparagopsis armata*), taking into consideration the underwater environmental factors. The depth and the occlusion were the two parameters in this study to simulate how the alga growth is influenced by the environment where the alga grows. Starting by building a prototype to explore different L-systems categories to model the alga, the stochastic L-systems with parametric features were selected to generate different alga plasticities. Qualitative methods were used to evaluate the designed grammar and alga's animation results by comparing videos and images of the *Asparagopsis armata* with the computer-generated versions.

Keywords

Procedural Generation, Parametric L-systems, Underwater Environment, Asparagopsis Armata

1 INTRODUCTION

The recreation of virtual worlds is a process that involves multi-disciplinary skills from mathematics to design, and art. 3D asset generation is a heavy processing task, and it takes specialised knowledge from the designer. To agile this gap, procedural content generation techniques were developed.

The novelty of this work is the application of generative grammars to procedural generate a virtual marine alga (Asparagopsis armata) using stochastic and parametric L-systems, influenced by depth and occlusion as environment impact growth factors.

L-systems are a generative grammar used for procedural techniques to model different species of plants. The designed L-system grammar can generate parametric non-deterministic algae that look like *Asparagopsis armata* influenced by external factors of the underwater environment. Depth and occlusion are the two parameters associated with simulating the underwater environment's impact factors on the plant growth.

The grammar was tested by building a prototype based on WebGL to apply the proposed L-system grammar. The application permits to control the axiom and production rules of the grammar and to parametrize the underwater environment. For example, the user can define the number of algae to be spawn, the space between them and the spots where the algae are sown.

A "wave "animation based on shaders was added to the algae, as well as a terrain with rocks and an animated water surface were used to enhance the underwater environment sensation.

The rest of the paper is structured as follows. Section 2 presents the literature review related to procedural content generation, L-systems and its taxonomy. The review process explores how L-systems can generate plants and vegetation content look-alike and finishes by presenting the alga's growth dependency on the context of seeded and particularities of *Asparagopsis armata*. Section 3 describes how the experimental design was set up, how different types of L-system were applied. Section 4 details how depth and occlu-

https://www.doi.org/10.24132/CSRN.3201.34

sion penalty factors were applied to the plant growth. Section 5 presents the results, the visual comparison between the computer-generated and real-world algae, and section 6 discusses the results. The paper ends by presenting the conclusions and future work.

2 LITERATURE REVIEW

Content creation is one of the main costs of developing a video game, it is estimated to be around 30%-40% of the US 20$M-$150M average budget for AAA games [Bar19]. To agile the content creation process, procedural content generation is an alternative to manual design of the game assets, providing techniques to automate the content generation [Hen13]. Procedural Content Generation (PCG) is the algorithm creation of the game content with limited or indirect user input [Tog11a].
The term content is related to the assets in a game: levels, maps, game rules, textures, stories, buildings, music, etc [Tog11b]. The procedural content can be characterized by grouping in what kind of content is generated, how the content is represented, and how the quality/fitness of the content is evaluated [Noo16]:

- Online-Offline Generation: the content is generated in real-time or before the game start.
- Necessary-Optional Content: if it is necessary or optional for the particular game.
- Control Degrees: type of generation algorithm that is used and how it can be parameterised.
- Deterministic or Stochastic Generation: the degree of randomness in the build process.
- Constructive or Generative-with-test: output of the algorithm. Constructive algorithms generate the content and end execution, producing the output result.

The methods and techniques used to generate the content can categorise the procedural content generation (PCG) in the following groups [Hen13]:

- Pseudo-Random Generation (PRNG): e.g., algorithms that produce random numbers based on mathematical formulas can be used to generate textures.
- Generative grammars (GG), e.g., Lindermayer-system, split grammars, wall grammars, shape grammars.
- Image Filtering (IF), e.g., Binary Morphology, Convolution Filters.
- Spatial Algorithms (SA), e.g., Tiling and layering, Grid subdivision, Fractals, Voronoi Diagrams
- Modelling and Simulation of Complex Systems (CS), e.g., Cellular Automata, Tensor fields, Agent-based simulation
- Artificial Intelligence (AI), e.g., Genetic Algorithms, Artificial Neural Networks, Constraint Satisfaction and planning

Networks, Constraint Satisfaction and planning.

Aristid Lindenmayer [Lin68] created Lindenmayer systems (L-systems) to model multi-cellular organisms, but their versatility proved suitable for modelling plants [Fit18] with three-like structures [Cio09]. L-systems are a formal grammar that produces strings that get rewritten over time, in parallel. The plant structure is decomposed into modules or components corresponding to the plant's physical units, e.g. a leaf. Each component is represented by a character. The axiom, composed of a specific string of components, defines the initial state of the plant. The production rules are composed of strings with components and information about the rotations to be applied to x, y and z axis [Bou12]. The plant life cycle can be reproduced by applying the production rules to the initial axiom through successive accumulative repetitions. As an example, the following three components can formally define an L-system {G, R, a}, where: a) G : set of finite symbols that represent the plant components; b) R : production rules, specifying the transitions; c) a : axiom representing the initial state of the plant; Then apply an L-system grammar defined by:

$$\begin{array}{rl} \text{a:} & a_r \\ \text{R1:} & a_r \rightarrow a_l b_r \\ \text{R2:} & a_l \rightarrow b_l a_r \\ \text{R3:} & b_r \rightarrow a_r \\ \text{R4:} & b_l \rightarrow a_l \\ \text{G:} & a_r,\ a_l,\ b_r, b_l \end{array}$$

Will produce the following sequence:

a_r
$a_l b_r$
$b_l a_r a_r$
$a_l a_l b_r a_l b_r$
$b_l a_r b_l a_r a_r b_l a_r a_r$
(...)

The L-system taxonomy can be grouped into the following types [Pru96]:

- DOL-systems (Deterministic and context-free): These are the simplest L-system. They are deterministic: the rules applied to the axiom will always generate the same grammar. As a consequence, all produced content is the same.
- Bracketed L-systems (Improve branching by saving state): Group a set of components inside a closed pair of the brackets. Restore the initial state after applying the rules inside the brackets.
- Stochastic L-systems (Define rules to a symbol based on a probability): The non-deterministic

technique, named stochastic L-system, was used to generate different plants from the same axiom and rules. Each plant component has more than one rule associated. Then, for each iteration is randomly chose a different rule associated with a component.

- Context-sensitive L-systems (Depends on neighbour's symbols context): Captures the interaction between adjacent elements of the developing structure. The activation of the production rules is based not only on substituted symbols but also on their neighbours.
- Differential L-systems: Enables interactions between plant components during development. Combination of discrete-continuous models of development, modules are created discretely, obeying the production rules but developed continuously described by differential equations.
- Parametric L-systems (Add numerical parameters, e.g., line length): The previous techniques produce plants with the same angles and components length. The visual result of the plant is complete symmetrically. Parametric L-system overcomes this limitation by configuring the length of the different components and the values of the rotation angles.

Turtle geometry [Gol04] can be used to draw the L-systems. The plant components are drawn one by one, moving forward in a defined heading. As a metaphor, each component is drawn on paper without lifting the pen. At each iteration step, the turtle saves a state composed of a position and heading. For L-systems that support branching when the production rules parser find a '[', it will push the turtle state to a stack, and when it finds ']', the turtle state will be popped, and the turtle states switch to the state before the '['.

The virtual generation of virtual crops has into consideration the environment's influence on the growth and the plasticity of the plant [Mar17] essential to generate reliable simulations to study the evolution of the crops. Soil fertility and space distribution are examples of influencing factors [Tal20] that can be used to define a fit function for modelling plant growth. Small mutations were added to the L-system grammar generation process to augment the plant diversity and validate the more suitable plants to survive in the simulated environment [Bor09]. This work will focus on the procedural generation of underwater environments. Diverse impact factors can influence the sea-life distribution, such as space competition, wave energy and sunlight attenuation [Li14].

The algae intended to be procedurally generated in this work lives in a marine ecosystem. These ecosystems are composed of multiple actors that can be procedural generated: terrain, biofouling, vegetation, water (caustics), and sea life. The red alga *Asparagopsis armata* was chosen to perform the procedural generation of marine algae. The *Asparagopsis armata* was first described in 1885 by Harvey and is an invasive alga from Australia [And04] present in the North Atlantic ocean. The alga commonly grows at depths from zero to ten meters and has high potential growth in environments with more light intensity [Mon05].

3 METHODS AND MATERIALS

Identifying the main alga components and the influence growth factors were the initial steps to set up the experiment. Then, an L-system grammar was designed to model the alga and impact factors on alga growth were defined. The experimental setup is based on a web application where a user can write axioms and production rules to generate a 3D visualisation of the alga.

A WebGL based library was used to build the prototype. In addition, different categories of L-system grammars are supported by the application, and marine environment parameterisations are available to the users through a minimal user interface.

3.1 Algae Components

To correctly model the alga, an initial study about alga *Asparagopsis armata* morphology was performed. Three main components were identified to model the alga plasticity: branches, stolons, and filaments. The filaments are a thin branch that looks like cotton, and stolons look like a hook/harpoon, and the branches constitute alga structure support (Figure 1 [1]).

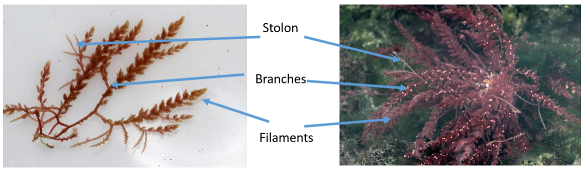


Figure 1: Algae plasticity.

Each component is computer-generated using basic geometric solids. A cylinder with a sphere on top is a branch. Filaments are represented by a cone. The stolon is composed of a cylinder as a base, a sphere and a cone (Figure 2).

3.2 Algae Grammar

A generative grammar was defined to model the alga, (Table 1). The branch is represented by the character 'B', the stolon by the character 'S', and filaments by the character 'F'. The '+' and '-' were used to rotate on the z-axis, the '\' and '/' were used to rotate on the x-axis, and the characters '<' and '>' to rotate on the y-axis. The characters '[' and ']' were used to create a new state and return to the previous form.

[1] http://www.omare.pt/pt/galeria-especies/687

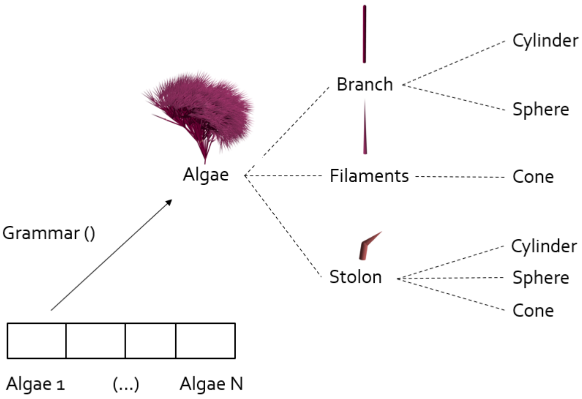


Figure 2: Alga's components.

Character	Action
Component	
B	Create a Branch component
S	Create a Stolon component
F	Create a Filament component
Axis rotation	
+	Rotate positive in the z-axis
-	Rotate negative in the z-axis
\	Rotate positive in the x-axis
/	Rotate negative in the x-axis
<	Rotate positive in the y-axis
>	Rotate negative in the y-axis
State	
[	Push (create a new state)
]	Pop (return to the previous state)

Table 1: L-system Grammar

The advance on the drawing process of the alga is performed by using the turtle graphics technique.

Different categories of L-systems were explored, starting by Deterministic Context-Free (DOL) with bracket support to branch preservation state, non-deterministic stochastic and parametric L-systems.

3.2.1 *Deterministic Context-Free (DOL) Systems*

With this category of L-system the rules applied to the axiom will always generate the same grammar. As a consequence, all algae look precisely the same. Starting by a simple axiom, a 'Branch 'will create a 'Stolon 'with a negative rotation of 30Â° degrees on the z-axis and a filament 'Stolon 'with a negative rotation of 30Â° degrees on the z-axis (Table 2).

Axiom	B
Rule for Branch	B-[S]-[F]
Rule for Stolons	S
Rule for Filaments	BF

Table 2: DOL-system grammar

The following character sequence is generated by iterating three times over the defined grammar:

Iteration 1 → B-[S]-[F]
Iteration 2 → B-[S]-[F]-[S]-[BF]
Iteration 3 → B-[S]-[F]-[S]-[BF]-[S]-[B-[S]-[F]BF]

The alga evolution dictated by the generated grammar is illustrated in Figure 3. Each iteration is added a new Branch, Stolon, and Filaments with a negative rotation in the z-axis.

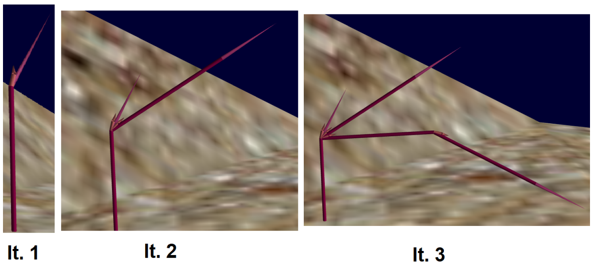


Figure 3: Iterating over the DOL-system grammar.

3.2.2 *Stochastic L-systems*

In nature, each alga is unique. To obtain similar results, non-deterministic grammar with stochastic behaviour was added to the prototype. Each component has more than one production rule, and each iteration is chosen randomly (Table 3). As a result, the same initial axiom and production rules will produce different characters chains and consequently different algae.

A weight factor can be attributed to each rule. This parametrization enables the creation of models of a stochastic alga in a specific direction or form. Figure 4, illustrates that, by randomly assigning different rules to the Filaments component made it possible to obtain three different algae at the end (Figure 5). The process is repeated for each component in iteration.

3.2.3 *Parametric L-System*

The previous techniques produce algae with the same angles and same components length. The final visual result of the algae is completely symmetrical. To configure the length of the different components and the values of the rotation angles two new parameters were added to the grammar, (Table 4).

A numeric value is assigned to the rotation angle before a character that corresponds to an axis rotation. If the character corresponds to a component, the numeric value will be assigned to the component's scale (Figure 6). If no value is present before the character, a default value is assigned.

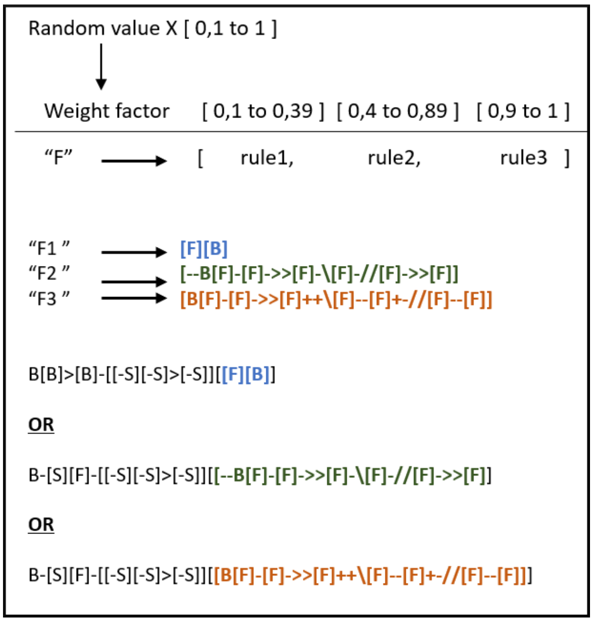
Thus, the value of the advance on the turtle geometry is the same as the component scale.

4 ALGAE GROWTH PENALTIES

According to [Mon05] the light intensity influences the alga (*Asparagopsis armata*) growth. In the underwater environment, the light tends to decrease with the increase of depth. Also, the occlusion by neighbour algae

Axiom	B
Rule for Branch 1	B[-S]<[[F]/[F]]+
Rule for Branch 2	B[B]>[B]
Rule for Stolons 1	[S]<[S]>[S]
Rule for Stolons 2	[-S][-S]>[-S]
Rule for Filaments 1	[F][B]
Rule for Filaments 2	[–B[F]-[F]-»[F]-\[F]-/ / [F]-»[F]]
Rule for Filaments 3	[B[F]-[F]-»[F]++\[F]–[F]+-//[F]–[F]]

Table 3: L-Stochastic grammar



Random value X [0,1 to 1]

Weight factor [0,1 to 0,39] [0,4 to 0,89] [0,9 to 1]

"F" → [rule1, rule2, rule3]

"F1 " → [F][B]

"F2 " → [--B[F]-[F]->>[F]-\[F]-//[F]->>[F]]

"F3 " → [B[F]-[F]->>[F]++\[F]--[F]+-//[F]--[F]]

B[B]>[B]-[[-S][-S]>[-S]][[F][B]]

OR

B-[S][F]-[[-S][-S]>[-S]][[--B[F]-[F]->>[F]-\[F]-//[F]->>[F]]

OR

B-[S][F]-[[-S][-S]>[-S]][[B[F]-[F]->>[F]++\[F]--[F]+-//[F]--[F]]]

Figure 4: Applying L-Stochastic production rules for the Filaments.

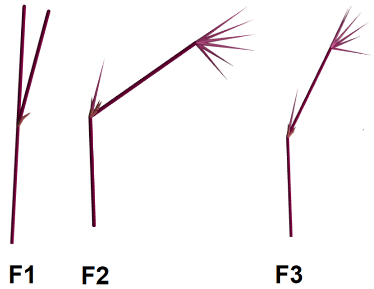


Figure 5: Possible outputs of applying L-stochastic Filaments production rules.

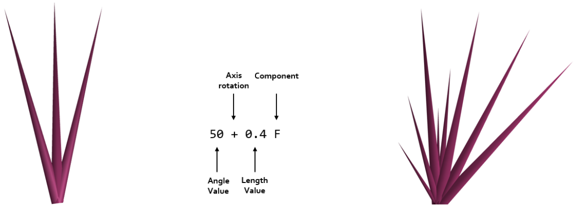


Figure 6: DOL-systems on the left with fixed angles and Filaments length, L-Parametric on the right enables to configure angles and components length.

Value	Action
0.1 ... 0.9	Value before a component character, corresponding to a scale value, e.g [0.9B] will scale the length of the branch to 90% of its unitary value.
1 ... 90	Value before a rotation component, corresponding to the value of the rotation angle, e.g [45+F] will rotate a Filaments component in positive on z-axis 45 degrees.

Table 4: Parametric L-system grammar

impacts the amount of light absorbed by s single alga. These environmental impact factors were chosen to influence the algae growth supported by the developed prototype, as illustrated on the fit function "(1)".

$$seed = \begin{cases} position = x, y, z \\ scale = 1 - (depth\,penalty + occlusion\,penalty) \end{cases} \tag{1}$$

4.1 Depth

In underwater environments, with the increase of depth, the light intensity decrease. The *Asparagopsis armata* produces algae with longer length on spots with higher light intensity. When the alga is seeded, a penalty factor is set to simulate the relation between depth and growth behaviour. With the decrease of y-value, a high level of penalty factor is set. This factor will correspond to a decrease in the scale of algae (Figure 7).

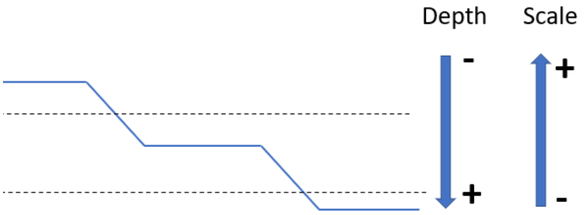


Figure 7: Seabed cross-cut relating depth with alga's scale.

4.2 Occlusion

The first approach to simulate the algae occlusion penalty was based on the z-value of the seed position related to the light spot. The algae that were further away from the light origin is assigned a higher penalty

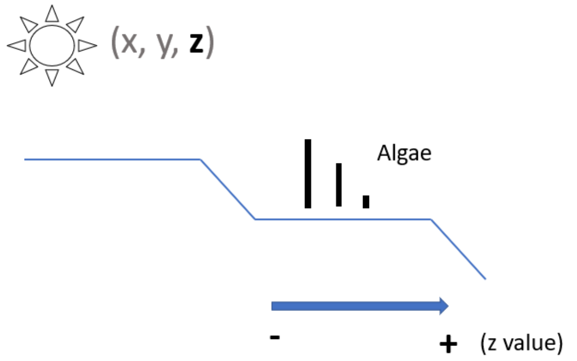


Figure 8: Seabed cross-cut representing algae further away from the fixed spotlight have a smaller scale.

factor. High values of this factor will correspond to a decrease in the alga's scale (Figure 8).

The second approach was based on the number of components produced from alga's grammar. Alga with more components will need more space to evolve and will occlude algae with fewer components. Algae with fewer components will have a high penalty factor that will decrease the scale of the alga (Figure 9).

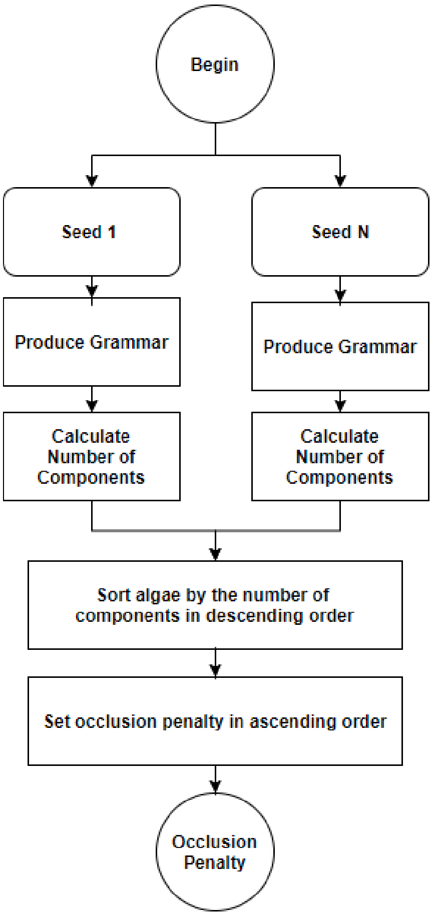

Begin
Seed 1
Seed N
Produce Grammar
Produce Grammar
Calculate Number of Components
Calculate Number of Components
Sort algae by the number of components in descending order
Set occlusion penalty in ascending order
Occlusion Penalty

Figure 9: Definition of the occlusion penalty based on the number of components.

5 ALGA ANIMATION

To animate the alga two different techniques were used. The first approach uses the CPU to animate the alga based on translations and rotations by changing only the x and y-coordinate values in the world coordinate system. When the alga is instantiated, a slightly random value is assigned to the z-axis (upwards direction) to give the sensation of individual wave movement to the alga within the group. The loop of the movement was based on a sinusoidal function (Figure 10).

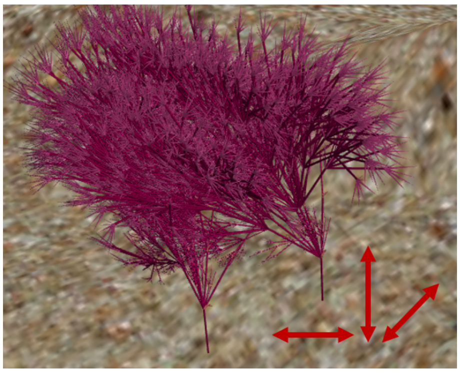


Figure 10: Alga animation based on translations.

The second approach of the alga animation was performed using shaders (GPU) that permit to obtain a dedicated animation for each alga. The vertice transformations are performed on the vertex shader. The x-coordinate is displaced, having as reference the y-coordinate value in the object coordinate system. The z-coordinate value is calculated based on a sinusoidal function to simulate the wave loop movement (Figure 11). The current 3D representation of the alga does not have a component hierarchy relation. The vertices with high values will decrease the offset value to avoid the algae components getting apart.

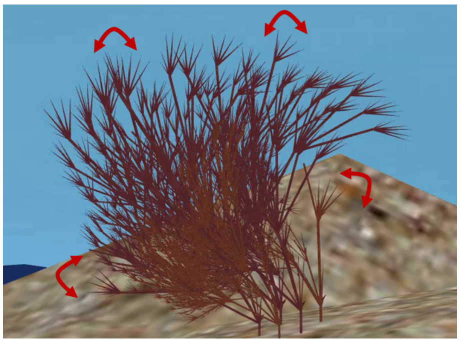


Figure 11: Individual alga animation using shaders.

6 RESULTS

6.1 L-system grammars

It was possible to explore different types of L-systems, starting from the most basic deterministic DOL-System to parametric L-systems, adding a non-deterministic behaviour (Figure 12).

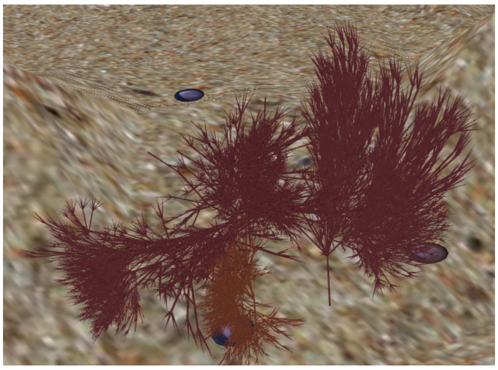


Figure 12: Three algae using parametric L-system with stochastic behaviour.

6.2 Depth penalty factor

The depth factor penalty produces the effect illustrated in Figure 13: algae instantiated in positions with lower y-values, are smaller than their neighbours with higher values on y-coordinates.

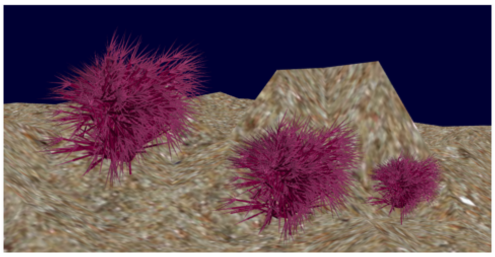


Figure 13: Visual representation of depth penalty.

6.3 Occlusion penalty factor

The first approach, using the z-coordinate as a reference relative to the light source, draws smaller algae comparing with algae spawned with high z-coordinate values (Figure 14).

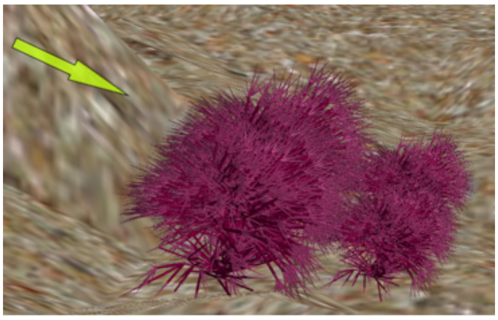


Figure 14: Occlusion using the alga distance to the spotlight. The green arrow illustrated the light direction from the light source

The second approach used to apply the occlusion penalty was based on the number of components generated by the initial grammar when sown. Algae with more number of components will have a minor penalty in their scale and, as a consequence, will produce larger algae (Figure 15).

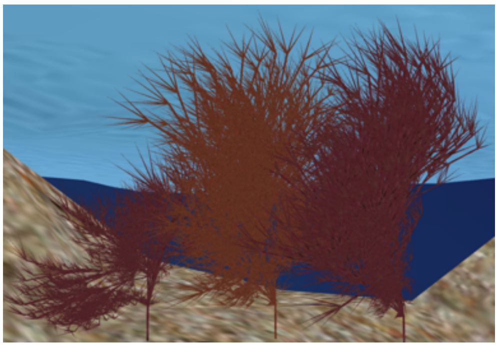


Figure 15: Occlusion is based on the number of components generated by the grammar.

6.4 Grammar render time

The grammar's performance was measured by logging the time it takes to display the L-system grammar during 1000 render loops. Table 5 presents the average time in milliseconds for the render loop session. The first trials were defined a DOL grammar and iterated 3 and 5 times through it. The same procedure was followed for the L-system parametric grammar.

Scene	Time (ms)
1 alga 3 iterations (DOL)	2.3
1 alga 3 iterations (Parametric)	100.7
1 alga 5 iterations (DOL)	39.8
1 alga 5 iterations (Parametric)	269.2
3 algae 3 iterations (DOL)	39.8
3 algae 3 iterations (Parametric)	179.5
3 algae 5 iterations (DOL)	54.1
3 algae 5 iterations (Parametric)	801.4

Table 5: Average time to render the L-system grammar.

The tests were performed in a machine with the following specifications: Processor: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz; RAM: 16 GB; Graphics card: NVIDIA GeForce RTX 2070 Max-Q; .OS: Windows 11.

7 DISCUSSION

The procedural content generation of the alga is validated using the qualitative methods by comparing the “look-and-feel ”of the alga generated by the grammar and penalty growth factors with images and videos of real *Asparagoris Armata*.

The procedural generation algorithm to generate the alga has evolved through the use of different categories of L-systems. The DOL-system grammar does not correctly model the alga. The Branches have all the same angle and length, which is not the default behaviour present in nature. The Stolons have the wrong proportions. With the introduction of parametric features to the L-system grammar, it was possible to model the components with different lengths and set different angles to the rotations to create a more natural look. The

Figure 16, illustrates the evolution of alga procedural generation starting on the left by presenting the output of the DOL-systems and on the right the output of parametric L-systems.

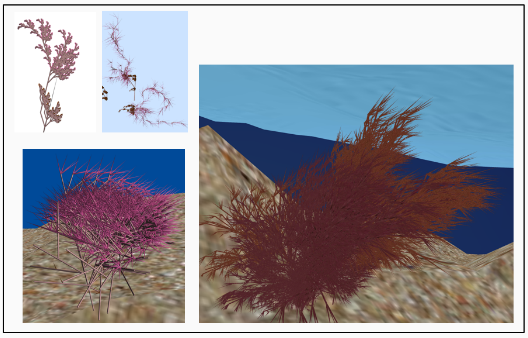


Figure 16: Alga procedural generation time-lapse.

In Figure 17, the alga generated with DOL-systems has the same angle for each rotation and length for all components of the same type. Consequently, this L-system generates algae that look symmetric and "quadratic". On the other side, using the parametric L-system with stochastic behaviour, it is possible to model algae with different rotation angles and component lengths. The final visual result is algae result with plasticities close to what is possible to encounter in nature.

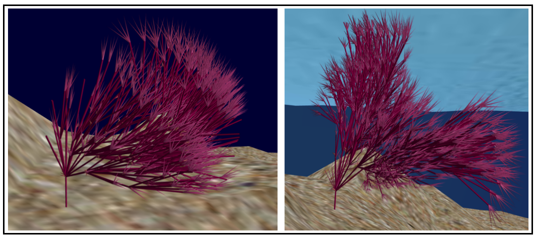


Figure 17: Comparison between the computer-generated DOL-system (left) and L-Parametric (right).

The occlusion penalty based only on the depth value was discarded (Figure 14) because the light is scattered in the underwater environment, and it is not directional from a single source. Having as a reference the distance to the spotlight will not get close to actual behaviour. The best occlusion technique is based on the number of alga components and the space it needs to evolve. The alga with more components will occlude the light to their neighbours that will grow less (Figure 15).

Concerning the alga animation, the algorithm that runs on the CPU is not visually correct because the algae move exclusively in a group and not individually, so the global algae visualisation looks too uniform. However, with the implementation of the animation on the shader (GPU), each alga has its own movement pattern and does not suffer from the uniform translation behaviour.

Using parametric L-systems with stochastic behaviour, plus the alga animation and the recreation of the underwater environment, it is possible to generate non-deterministic content similar to the alga *Asparagopsis armata*. Also, combining generative grammars based on the parametric L-systems, animation, and growth penalty factors makes it possible to model algae similarly to *Asparagopsis armata*. (Figure 18 [2]).

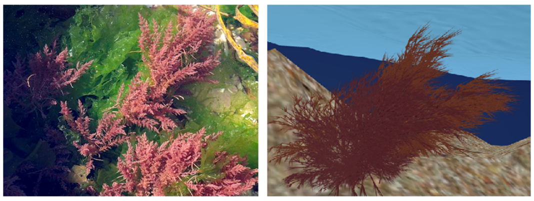


Figure 18: Comparison between alga in the underwater environment (left) and the computer-generated (right).

A comparison video between the algae in the underwater environment and the computer-generated version is available on supplemental material.

The DOL-Systems are faster regarding the grammar's render time to generate the alga(e). The parametric L-system uses a scale factor to resize individual components of the alga, and for each iteration, it is necessary to save the component's state, leading to high values related to the render time to display the alga(e).

8 CONCLUSIONS AND FUTURE WORK

With this work was possible to generate 3D models of the marina algae *Asparagopsis armata* look-alike. The developed prototype made it possible to explore different types of L-systems to model the marine alga *Asparagopsis armata*. Furthermore, the procedural generation process included external environmental factors (depth and occlusion) that influence algae growth. The generative grammar with the best results was the combination of parametric L-system with stochastic behaviour, enabling the generation of unique look for each alga and, at the same time, algae with similar morphologies to those found in nature. The procedural generation of the occlusion impact in the alga's growth, considering the sparsity of the light of the underwater environment, the number of the components of the alga's neighbours was the technique chosen.

As future work, it will be created a relationship or hierarchy between the different alga components. This hierarchy will enable the exploration of animation mechanisms, e,g. Kinematics, and add rules to the grammar to procedural generate them. Also the generation of multiple alga's component for a high number of iterations will be improved. When the complexity of the grammar grows, it is resource-heavy to draw all the primitives. Writing complex L-systems grammars is

[2] https://flic.kr/p/Mc8uLp

prone to errors, and it is challenging to design hierarchical relations between the plant and the components only by typing text into the system. In future iterations, incorporate strategies of inverse procedural modelling [Sta10], and control mechanism based on data-driven procedural techniques to infer the grammar from images or sketch-based inputs to provide intuitive interactions to the designers [Len21].

9 REFERENCES

[And04] Andreakis, N., Procaccini, G. & Wiebe HCF Kooistra Asparagopsis taxiformis and Asparagopsis armata (Bonnemaisoniales, Rhodophyta): genetic and morphological identification of Mediterranean populations. *European Journal Of Phycology*. 39, 273-283 (2004).

[Bar19] Barriga, N. A Short Introduction to Procedural Content Generation Algorithms for Videogames. *International Journal On Artificial Intelligence Tools*. 28, 1930001 (2019).

[Bor09] Bornhofen, S. & Lattaud, C. Competition and evolution in virtual plant communities: a new modeling approach. *Natural Computing*. 8, 349-385 (2009), https://doi.org/10.1007/s11047-008-9089-5

[Bou12] Boudon, F., Pradal, C., Cokelaer, T., Prusinkiewicz, P. & Godin, C. L-Py: An L-System Simulation Framework for Modeling Plant Architecture Development Based on a Dynamic Language. *Frontiers In Plant Science*. 3 pp. 76 (2012).

[Cio09] Ciosek., K. & Kotowski., P. GENERATING 3D PLANTS USING LINDENMAYER SYSTEM. *Proceedings Of The Fourth International Conference On Computer Graphics Theory And Applications - GRAPP, (VISIGRAPP 2009)*. pp. 76-81 (2009).

[Fit18] Fitch, B., Parslow, P. & Lundqvist, K. Evolving Complete L-Systems: Using Genetic Algorithms for the Generation of Realistic Plants. *Artificial Life And Intelligent Agents*. pp. 16-23 (2018).

[Gol04] Goldman, R., Schaefer, S. & Ju, T. Turtle geometry in computer graphics and computer-aided design. *Computer-Aided Design*. 36, 1471-1482 (2004).

[Hen13] Hendrikx, M., Meijer, S., Van Der Velden, J. & Iosup, A. Procedural content generation for games: A survey. *ACM Transactions On Multimedia Computing, Communications And Applications*. 9, 1-22 (2013).

[Len21] Lena Gieseke, Paul Asente, Radomír Měch, Bedrich Benes, and Martin Fuchs. A survey of control mechanisms for creative pattern generation. *Computer Graphics Forum*, 40(2), pp.585-609, 2021

[Li14] Li, R., Ding, X., Yu, J., Gao, T., Zheng, W., Wang, R. & Bao, H. Procedural generation and real-time rendering of a marine ecosystem. *Journal Of Zhejiang University SCIENCE C*. 15, 514-524 (2014).

[Lin68] Lindenmayer, A. Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs. *Journal Of Theoretical Biology*. 18, 300-315 (1968).

[Mar17] Marshall-Colon, A., Long, S. P., Allen, D. K., Allen, G., Beard, D. A., Benes, B., von Caemmerer, S., Christensen, A. J., Cox, D. J., Hart, J. C., Hirst, P. M., Kannan, K., Katz, D. S., Lynch, J. P., Millar, A. J., Panneerselvam, B., Price, N. D., Prusinkiewicz, P., Raila, D., Shekar, R. G., Shrivastava, S., Shukla, D., Srinivasan, V., Stitt, M., Turk, M. J., Voit, E. O., Wang, Y., Yin, X., and Zhu, X.-G. Crops In Silico: Generating Virtual Crops Using an Integrative and Multi-scale Modeling Platform *Frontiers in Plant Science*, vol.8, (2017).

[Mon05] Monro, K. & Poore, A. Light quantity and quality induce shade-avoiding plasticity in a marine macroalga. *Journal Of Evolutionary Biology*. 18, 426-435 (2005).

[Noo16] Noor Shaker, Julian Togelius & Mark J. Nelson Procedural Content Generation in Games. (Springer, Cham,2016).

[Pru96] Prusinkiewicz, P. & Lindenmayer, A. The Algorithmic Beauty of Plants. (Springer-Verlag,1996).

[Sta10] Št'ava, Ondrej and Beneš, Bedrich and Měch, Radomir and Aliaga, Daniel G and Krištof, Peter. Inverse procedural modeling by automatic generation of L-systems. *Computer Graphics Forum*, vol. 29, pp.665-674, (2010)

[Tal20] Talle, J. & Kosinka, J. Evolving L-Systems in a Competitive Environment. *Advances In Computer Graphics*. pp. 326-350 (2020).

[Tog11a] Togelius, J., Kastbjerg, E., Schedl, D. & Yannakakis, G. What is Procedural Content Generation? Mario on the Borderline. *Proceedings Of The 2nd International Workshop On Procedural Content Generation In Games*. (2011).

[Tog11b] Togelius, J., Yannakakis, G., Stanley, K. & Browne, C. Search-Based Procedural Content Generation: A Taxonomy and Survey. *IEEE Transactions On Computational Intelligence And AI In Games*. 3, 172-186 (2011).

Fitting Parameters for Procedural Plant Generation

Albert Garifullin
Lomonosov Moscow State University
GSP-1, Leninskie Gory
119991, Moscow, Russia
albgar-14@yandex.ru

Alexandr Shcherbakov
Lomonosov Moscow State University
GSP-1, Leninskie Gory
119991, Moscow, Russia
alex.shcherbakov@graphics.cs.msu.ru

Frolov Vladimir
Lomonosov Moscow State University
Keldysh Institute of Applied Mathematics
GSP-1, Leninskie Gory
119991, Moscow, Russia
vfrolov@graphics.cs.msu.ru

ABSTRACT

We propose a novel method to obtain a 3D model of a tree based on a single input image by fitting parameters for some procedural plant generator. Unlike other methods, our approach can work with any plant generator, treating it as a black-box function. It is also possible to specify the desired characteristics of the plant, such as the geometric complexity of the model or its size. We propose a similarity function between the given image and generated model, that better catches the significant differences between tree shapes. To find the appropriate parameter set, we use a specific variant of a genetic algorithm designed for this purpose to maximize similarity function. This approach can greatly simplify the artist's work. We demonstrate the results of our algorithm with several procedural generators, from a very simple to a fairly advanced one.

Keywords

3D modeling, plants modeling, tree reconstruction, genetic algorithms

1. INTRODUCTION

Trees and other plants play a key role in shaping the landscapes around us, and therefore a realistic representation of vegetation is one of the important tasks of computer graphics. The use of tree models is necessary for many industries - from computer games and virtual reality systems to architecture and urban planning. Nowadays diversity and realism in the visualization of vegetation are needed.

It is possible to solve this problem with the help of 3D reconstruction, creating a model of a tree that exists in the real world. However, this requires a detailed representation of the tree structure like laser scanning data with a high level of detail, but such data is difficult and expensive to obtain. The reconstruction of the tree from the images probably won't be accurate, as many branches are hidden by the crown or strongly intertwined with each other. An alternative to reconstructing trees can be their creation using procedural generation. The main problem with this approach is that the tree is created from a set of some input parameters and their correct selection requires a lot of time from the artist. Various parameters significantly affect each other and it is often not obvious to the user how changing some value of the input parameter affects the final model.

We propose an algorithm that combines both of these approaches and performs image-based modeling of plants. It takes a single image and a procedural generator and finds such a set of parameters of this generator that results in a tree most similar to the image. The appropriate set of parameters is found as the maximum point of the “similarity” function of the model and the original image. To optimize this function, a special version of the genetic algorithm was implemented. The main advantage of our solution is its independence from the generator, as the algorithm treats it like a black box, while existing solutions rely heavily on their own generators, inevitably limited in their capabilities. Also, our approach allows users to specify some other desired properties of the model, such as its geometric complexity, which is useful for practical application.

2. RELATED WORK

Trees Modeling

The problem of vegetation modeling has been an area of scientific interest for many years. At the moment, all approaches to plant modeling can be divided into 3 groups - interactive, procedural, and reconstructive [Sme14]. Interactive methods are the most widely used in the industry. Projects like SpeedTree [St17] or Xfrog [Xf17] provide an artist with a tool that allows him to interactively create highly detailed tree models, but their use requires a certain level of skill, and creating a model takes a lot of time. Procedural

https://www.doi.org/10.24132/CSRN.3201.35

generation methods can create a model of a plant without human involvement, relying only on a certain set of input parameters and, possibly, a description of the environment. There are many different methods of procedural modeling of plants. Early works used sets of rules to describe the structure of a tree [Hon71][Web95], L-systems [Pru86][Pru12], cellular automata [Gre89], and particle systems [Ree85]. All of them somehow come down to the recursive construction of a plant model, without taking into account the environment. Newer works concentrate on simulating the growth process and the influence of the environment on it [Lon12][Had17][Yi15][Yi18]. There are, although less commonly used in practice, methods for reconstructing tree models from multiple images [Neu07][Ch08], video [Li11], and laser-scanned 3D point clouds [Liv10][Du19], including using neural networks [Liu21].

Image-Based Modeling

The closest to our work are combined methods involving the use of a procedural generator for model creation. So in [St14], a 3D model of the tree is used, according to which the generator parameters are selected. The work [Tan08] focuses on modeling a tree based on a single image, in which the user needs to manually select the main branches and crown. According to this image, a 3D model of the tree is assembled from a given set of branches. The work [Li21] also uses only one image for creating a model, but fully automates this process by using three neural networks: the first is used to segment the image, the second to form an approximate representation of the tree in the form of radial bounding volumes, and the third to determine the type of plant. A plant species is defined as a set of parameters for a specific procedural generator, also described in the paper.

All these works are based on the use of their own procedural methods of plant modeling. This imposes serious restrictions on the application of their results since only those trees that can be described by the procedural model used in the work can be created. In addition, existing works do not allow the user to control how complex and detailed a 3D model will turn out, although this is necessary in many cases. Our method implements the same image-based approach but does not have these restrictions because it can work with different procedural generators.

Differentiable Rendering

In addition to specific reconstruction methods for trees, there are more general approaches. Currently, approaches based on differentiable rendering are popular for solving problems of accurate reconstruction [Has21][Mun21][Tak22][Hu22]. In these approaches, differentiable rendering can be used to propagate the error backward from the triangular mesh to the original model. So in [Mun21] the error spread from the mesh to the tetrahedron grid to optimize the topology, and in [Hu22] to the parameters of the procedural texture generator. Although this approach looks promising, there are several difficulties in applying it to the procedural generation of vegetation. Firstly, the procedural model must be differentiable, which is generally not satisfied. Secondly, small changes in the parameters of procedural generators do not always lead to small changes in the resulting 3D model. This means that gradient optimization methods will not have the expected efficiency that can be observed in well-known applications of differentiable rendering.

3. OVERVIEW

Our goal is to obtain a 3D model of a tree from its 2D image or sketch using some given procedural generator. In this case, the generator is considered as a black box, at the input of which is a certain set of numerical parameters, and at the output is a plant model represented by a graph of branches. In addition to the image, the user can specify their requirements for some properties of the resulting model that cannot be obtained directly from the image. For example, limit the number of nodes in the branch graph, i.e. the geometric complexity of the resulting model.

To solve this problem, a function is proposed that evaluates the degree of similarity of the model made by the generator with the original image, taking into account restrictions. Then the task of creating a tree similar to the image is reduced to finding the global maximum of this function on the entire set of acceptable parameters. It is important to note that, although the function has values from 0 to 1, its maximum value is unknown, because it is not guaranteed that the procedural generator can create a tree exactly of the required type. Moreover, it is obvious that this problem has not the only solution, since there may be different 3D models that are equally similar to the input image. To simplify working with the function, we transform each parameter's value into [0, 1] interval so the function is defined on the set $[0, 1]^n$, where n - is a number of generator's parameters.

To solve this optimization problem, a special version of the genetic algorithm was developed. Like any genetic algorithm, it does not guarantee the achievement of a global maximum, but on average it can find points good enough in terms of the function value. In its work, the genetic algorithm uses statistical information about the entire family of functions corresponding to different input images, but the same procedural generator.

The results of the algorithm with three different procedural generators and several different images are also demonstrated.

4. SIMILARITY FUNCTION

The value of similarity function is a multiplication of image similarity value and characteristic multipliers.
Image similarity value is obtained by comparings the source image with the impostors of the generated tree model. Semantic masks are used for comparison, where each pixel belongs to one of three categories: branch, foliage, background. To obtain such a mask from the source image, a neural network can be used similarly to [Li21], and in simple cases, we can get it based only on pixel color (green corresponds to leaves, brown or gray - to branches and trunk).

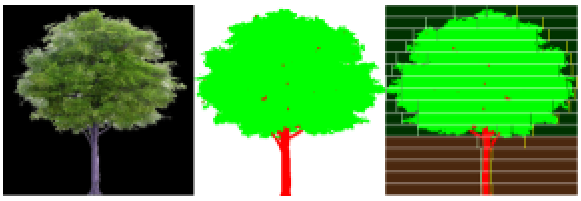


Figure 1. The original image, semantic mask, visualization of the division into stripes. Stripes of brown color correspond to the trunk, green color to the crown. Vertical lines inside each stripe - values ai, bi, ci, di respectively.

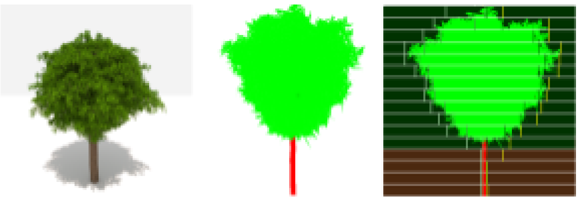


Figure 2. Generated tree, semantic mask (imposter), visualization of the division into stripes.

The image is divided into 20-30 narrow horizontal stripes, for each of which are determined:

- $[a_i, d_i]$ - crown borders
- $[b_i, c_i]$ - dense crown borders (>75% leaves pixels)
- B_i - branches pixels percentage
- L_i - leaves pixels percentage

According to the ratio B_i/L_i, each stripe refers either to the crown or to the trunk, see Figs. 1 and 2.
Comparing the parameters a_i, b_i, c_i, d_i and B_i/L_i ratio for every stripe of the original image and the image of the generated model, we calculate the value of image similarity $ImSim$.
Characteristic multiplier shows the difference in the characteristics of the model and given one. For characteristic C, the multiplier has the following formula:

$$C_{mul} = min(C_{model}, C_{reference}) / max(C_{model}, C_{reference})$$

The final function value:

$$Sim = ImSim * \prod_{c \in \mathbb{C}} C_{mul}$$

$\mathbb{C}$ - the set of all given characteristics. Among them may be:

- Number of vertices in the branch graph
- Height and width in some scale
- Average branches and leaves density
- Average leaf size

None of these characteristics is mandatory, but it is recommended to specify the number of vertices in the branch graph, otherwise the search for a solution will slow down due to the need to search for it among models with very high geometric complexity.

5. GENETIC ALGORITHM IMPLEMENTATION

The previously mentioned similarity function is used as an objective function for the genetic algorithm.
A proposed genetic algorithm consists of several elementary genetic algorithms, with a selection of the best results of each of them. Each elementary GA includes the initialization of a population and its evolution over a fixed number of generations. In the figure, each vertex of the tree is such an elementary GA. Algorithms on the leaves of the tree start with a randomly initialized population, and all the others form a population of the fittest “individuals” obtained in the child vertices.
All elementary GA work according to the same strategy ($f(x)$ - objective function)

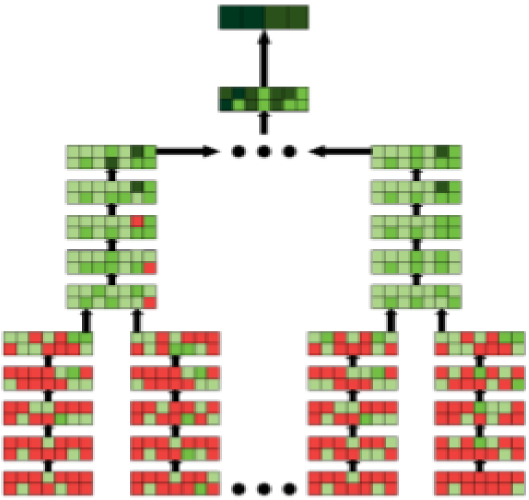


Figure 3. Tree-like population structure. Zero-level elementary GA are started with random genes, while GA from level i takes best species from two final populations of the previous level. The final result of a whole algorithm is a small set of best species on the very top level.

Selection

At the beginning of each iteration, half of the population with the worst fitness value is removed. The remaining individuals take part in the creation of a new generation. At each of the vacant places in the population, a new individual is created with one-dot crossover, its parents are selected from the remaining species of the previous generation. The fitness proportionate selection is used, which means that the probability of choosing an individual as a parent is proportional to the value of its fitness. For representatives of the new generation, the values of the objective function and the fitness function are calculated.

Mutation

Mutation chance M_{chance} and percentage of genes to change M_{genes} are constant.

Here is a proposed method for genome G mutation:

1) values of $n * M_{genes}$ randomly chosen genes are changed. The probability of mutation in the k gene is proportional to the *average rate of change* of this gene $V(k)$
2) For the result genome G' we estimate its *quality* $Q(G')$
3) Steps 1-2 are repeated several times (500 in the experiment) and the mutation results in a gene with a better quality score

Functions $V(k)$ and $Q(G')$ based on pre-collected information about the entire family of objective functions F = $\{f(x)\}$ (each function corresponds to its own input image and set of properties)

Gene value

$$V(k) \ = \ Average(\frac{f(\bar{x} + h^{*}x_k) - f(\bar{x})}{h})$$

Genome quality

$$P(i,j) \ = P(f(\bar{x}) \ > \ eps \,|(j \ - \ 1)/k \ < \ x_i \ < j/k),$$
$$i \in \{1, ..., n\}, \, j \ \in \{1, ..., k\}$$
$$P_0 \ = \ P(f(\bar{x}) \ > \ eps)$$
$$Q(G') \ = \ \sum_{i=1}^{n} Q(i, \lceil G_i^{'} * \, k \rceil)$$
$$Q(i, \, j) \ = \ sgn(P(i,j) \ - P_0) \ * \ max(P(i,j), P_0)/min(P(i,j), P_0)$$

Several hundred thousand calculations of functions were carried out for a fairly accurate assessment of $f(\bar{x})$ values for several dozen different images with each of the used procedural generators.

6. RESULTS

We implemented three different procedural generators to demonstrate the results of our method:

1) WeberPennGen - implementation of the algorithm described in [Web95]
2) GEGen - a generator simulating the process of tree growth, taking into account the environment, based on [Yi15][Yi18]
3) SimpleGen - a generator with a simple set of rules for the recursive description of the tree structure, used for testing purposes during development

In the experiments, images of trees were used as input data, the only additional requirement was the geometric complexity of the model. The result of the algorithm was a group of several best candidates, the selection of which was performed manually. To get the result, 40-60 thousand calls of the procedural generator were required, 10-50 minutes for calculation on a PC with AMD Ryzen 7 3700X, 16 GB RAM, Nvidia GeForce RTX 3070. Figures 4-8 show the ability of our algorithm to deal with different tree species. You could notice that the model takes from the image not only the approximate shape of the crown but also the structure of its edge. Images in Fig. 6 and 9 both have cone-shaped crowns, but the thuja in Fig. 6 has smooth edges, close to a real cone and the spruce tree in Fig. 9 has distinct branches with visible gaps between them. The density of the crown is also preserved, as you can see in Fig. 5 (both reference image and model have dense crows) and Fig. 8, where there are trees with a lot of gaps between branches. Fig. 7 shows the ability of our algorithm to use a very simple sketch as an input.

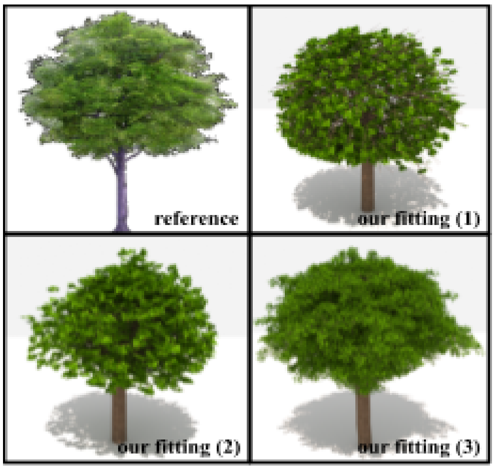


Figure 4. Input image and 3 best candidates created with WeberPennGen

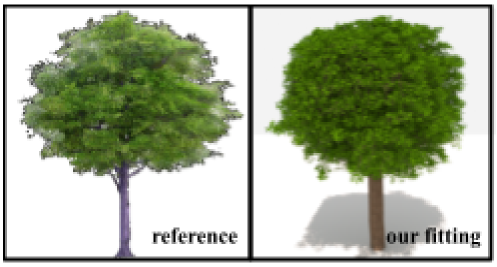


Figure 5. The same reference image as in Fig. 4, but GEGen generator is used

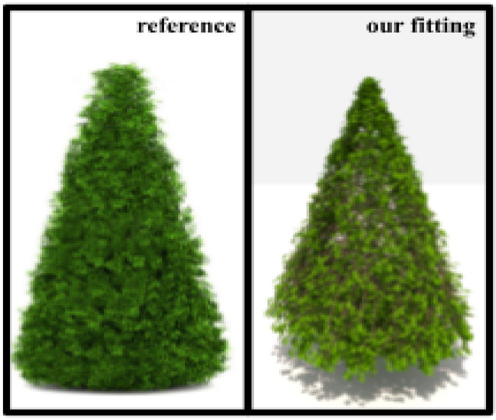


Figure 6. Decorative thuja tree

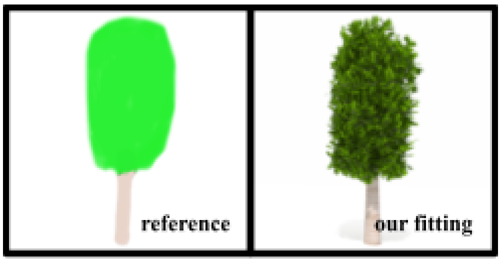


Figure 7. Tree based on simple sketch

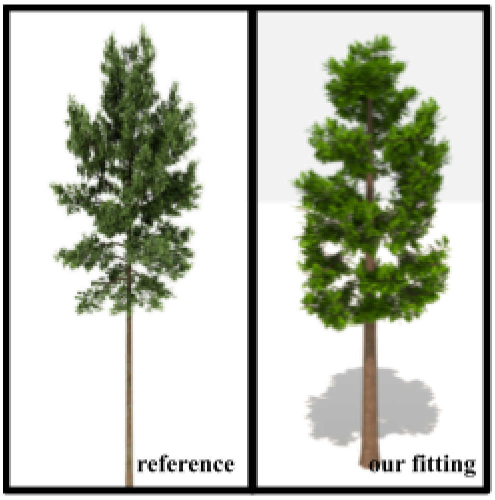


Figure 8. Small pine tree

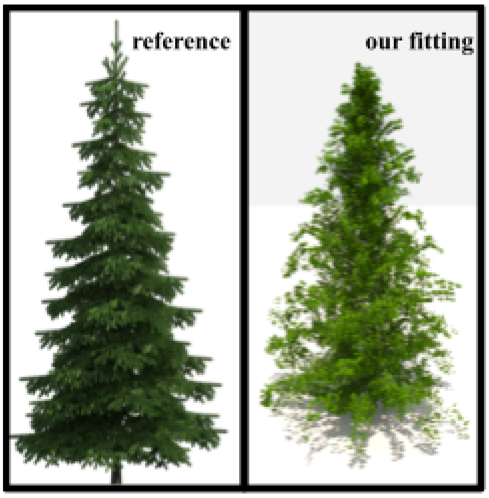


Figure 9. Spruce tree

7. CONCLUSION

A novel method of obtaining a 3D model of a tree from a single image was proposed. It is mostly autonomous: a user is needed at the very end to choose one of the created models. Unlike previous works, our method can work with an arbitrary procedural plant generator, which makes it easy to use modern solutions in this area without changing the algorithm itself. This, as well as the ability to specify the required complexity of the model, makes it more applicable for computer graphics applications.

The framework implemented in the process of working on the paper demonstrates the abilities of the method, but it can be improved in many different aspects. We consider the most promising refinement of the similarity function, as well as an additional assessment of the realism of the model in addition to its comparison with the sample. It is also promising to create a specialized tool for 3D modeling of plants using our algorithm or integrate it into existing ones.

8. ACKNOWLEDGMENTS

The work was sponsored by the non-profit Foundation for the Development of Science and Education "Intellect".

9. REFERENCES

[Sme14] Smelik, Ruben M., Tim Tutenel, Rafael Bidarra, and Bedrich Benes. "A survey on procedural modelling for virtual worlds." In Computer Graphics Forum, vol. 33, no. 6, pp. 31-50. 2014.

[St17] SpeedTree IDV Inc, 2017. URL: http://www.speedtree.com

[Xf17] Greenworks Organic Software. Xfrog procedural organic 3D modeler, 2017. URL: http://xfrog.com

[Pru86] Prusinkiewicz, Przemyslaw. "Graphical applications of L-systems." In Proceedings of graphics interface, vol. 86, no. 86, pp. 247-253. 1986.

[Yi15] Yi, Lei, Hongjun Li, Jianwei Guo, Oliver Deussen, and Xiaopeng Zhang. "Light-Guided Tree Modeling of Diverse Biomorphs." In PG (Short Papers), pp. 53-57. 2015.

[Yi18] Yi, Lei, Hongjun Li, Jianwei Guo, Oliver Deussen, and Xiaopeng Zhang. "Tree growth modelling constrained by growth equations." In Computer Graphics Forum, vol. 37, no. 1, pp. 239-253. 2018.

[St14] Stava, Ondrej, Sören Pirk, Julian Kratt, Baoquan Chen, Radomír Měch, Oliver Deussen, and Bedrich Benes. "Inverse procedural modelling of trees." In Computer Graphics Forum, vol. 33, no. 6, pp. 118-131. 2014.

[Li21] Li, Bosheng, Jacek Kałużny, Jonathan Klein, Dominik L. Michels, Wojtek Pałubicki, Bedrich Benes, and Sören Pirk. "Learning to reconstruct botanical trees from single images." ACM Transactions on Graphics (TOG) 40, no. 6 (2021): 1-15.

[Pru12] Prusinkiewicz, Przemyslaw, and Aristid Lindenmayer. The algorithmic beauty of plants. Springer Science & Business Media, 2012.

[Hon71] Honda, Hisao. "Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body." Journal of theoretical biology 31, no. 2 (1971): 331-338.

[Web95] Weber, Jason, and Joseph Penn. "Creation and rendering of realistic trees." In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pp. 119-128. 1995.

[Gre89] Greene, Ned. "Voxel space automata: Modeling with stochastic growth processes in voxel space." In Proceedings of the 16th annual conference on Computer graphics and interactive techniques, pp. 175-184. 1989.

[Ree85] Reeves, William T., and Ricki Blau. "Approximate and probabilistic algorithms for shading and rendering structured particle systems." ACM siggraph computer graphics 19, no. 3 (1985): 313-322.

[Lon12] Longay, Steven, Adam Runions, Frédéric Boudon, and Przemyslaw Prusinkiewicz. "TreeSketch: Interactive Procedural Modeling of Trees on a Tablet." In SBIM@ Expressive, pp. 107-120. 2012.

[Had17] Hädrich, Torsten, Bedrich Benes, Oliver Deussen, and Sören Pirk. "Interactive modeling and authoring of climbing plants." In Computer Graphics Forum, vol. 36, no. 2, pp. 49-61. 2017.

[Neu07] Neubert, Boris, Thomas Franken, and Oliver Deussen. "Approximate image-based tree-modeling using particle flows." In ACM SIGGRAPH 2007 papers, pp. 88-es. 2007.

[Li11] Li, Chuan, Oliver Deussen, Yi-Zhe Song, Phil Willis, and Peter Hall. "Modeling and generating moving trees from video." ACM Transactions on Graphics (TOG) 30, no. 6 (2011): 1-12.

[Liv10] Livny, Yotam, Feilong Yan, Matt Olson, Baoquan Chen, Hao Zhang, and Jihad El-Sana. "Automatic reconstruction of tree skeletal structures from point clouds." In ACM SIGGRAPH Asia 2010 papers, pp. 1-8. 2010.

[Du19] Du, Shenglan, Roderik Lindenbergh, Hugo Ledoux, Jantien Stoter, and Liangliang Nan. "AdTree: accurate, detailed, and automatic modelling of laser-scanned trees." Remote Sensing 11, no. 18 (2019): 2074.

[Liu21] Liu, Yanchao, Jianwei Guo, Bedrich Benes, Oliver Deussen, Xiaopeng Zhang, and Hui Huang. "TreePartNet: neural decomposition of point clouds for 3D tree reconstruction." ACM Transactions on Graphics 40, no. 6 (2021).

[Tan08] Tan, Ping, Tian Fang, Jianxiong Xiao, Peng Zhao, and Long Quan. "Single image tree modeling." ACM Transactions on Graphics (TOG) 27, no. 5 (2008): 1-7.

[Ch08] Chen, Xuejin, Boris Neubert, Ying-Qing Xu, Oliver Deussen, and Sing Bing Kang. "Sketch-based tree modeling using markov random field." In ACM SIGGRAPH Asia 2008 papers, pp. 1-9. 2008.

[Has21] Hasselgren, Jon, Jacob Munkberg, Jaakko Lehtinen, Miika Aittala, and Samuli Laine. "Appearance-Driven Automatic 3D Model Simplification." arXiv preprint arXiv:2104.03989 (2021).

[Mun21] Munkberg, Jacob, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. "Extracting Triangular 3D Models, Materials, and Lighting From Images." arXiv preprint arXiv:2111.12503 (2021).

[Tak22] Takimoto, Yusuke, Hiroyuki Sato, Hikari Takehara, Keishiro Uragaki, Takehiro Tawara, Xiao Liang, Kentaro Oku, Wataru Kishimoto, and Bo Zheng. "Dressi: A Hardware-Agnostic Differentiable Renderer with Reactive Shader Packing and Soft Rasterization." arXiv preprint arXiv:2204.01386 (2022).

[Hu22] Hu, Yiwei, Chengan He, Valentin Deschaintre, Julie Dorsey, and Holly Rushmeier. "An Inverse Procedural Modeling Pipeline for SVBRDF Maps." ACM Transactions on Graphics (TOG) 41, no. 2 (2022): 1-17.

A browser based system for remote evaluation of subjective quality of videos

Robert Furman Rafał Hadzicki Damian Karwowski Krzysztof Klimaszewski

Poznan University of Technology
Pl. M. Skłodowskiej-Curie 5
Poland, 60-965, Poznań
krzysztof.klimaszewski@put.poznan.pl

ABSTRACT

The evaluation of the quality of videos is posing a significant problem in research on video processing, including compression. Different algorithms and their modifications aim to improve the quality of the video, while keeping the necessary bitrate as low as possible, and every time, the performance of the method must be evaluated. The bitrate can be measured readily, while for quality there are different metrics and, therefore – different results. The ultimate verdict always has to be the subjective opinion about the quality that is expressed by the viewers. For now, the only reliable way to measure the quality of the video is to perform a subjective test with a group of viewers. Subjective tests are difficult to perform – not only one has to gather all viewers in one place, but also the evaluation schemes may be difficult to follow. In the paper we present an implementation of a system that greatly simplifies the process of performing the subjective quality assessment of videos. We discuss strengths and weaknesses of the system when compared to the commonly used procedures. Finally, we also provide the results of a quality assessment for the HEVC video encoder. Presented data proves that the idea of remote quality evaluation together with the usage of the two-level grade is a valid one and provides reliable results.

Keywords
subjective video quality

1. INTRODUCTION

In the process of developing the new video compression algorithms or modifying some processing steps in the video compression pipeline, several factors must be accounted for. Surely, one of the factors is the computational complexity, that translates to the time required to perform the processing step, the ability to perform the calculations in parallel, then there is the bitrate gain or loss that influences the size of encoded data stream, and finally, there is the quality of the reconstructed video. The last factor is difficult to evaluate because it has to take into account the way that humans perceive video, since most of the time the goal is to have as high a quality perceived by the viewers as possible. It is difficult to obtain the results similar to the perceived quality in software. Many commonly used metrics, like PSNR or SSIM are aimed at static image quality evaluation, and even then, for some specific cases, they provide results dramatically different than subjective tests performed by viewers. Therefore, new metrics are being developed, like VMAF that aim to model the perception mechanisms of video and provide results that closely correlate with the subjective tests.

The importance of being able to use a piece of software to evaluate the quality of the video and get results that agree well with the subjective tests is obvious, when we consider the applications in rate-distortion evaluation in video compression algorithms. For such applications one needs to perform many evaluations during a compression run.

In the absence of such algorithms, the only reliable way to evaluate the quality of the video is to perform the subjective tests, which is a very difficult process.

The methods for performing the subjective quality assessment has been formalized a long time ago and the details can be found in official recommendations, like ITU-R BT.500 [BT500] as well as the ITU-T P.910 [P910], P.911 [P911] and P.913 [P913]. Most of the recommendations are regularly updated. The update of recommendations is necessitated mostly by the developments in the video delivery methods.

In the recent years, the viewers are no longer limited to cinemas and stationary TV-sets in order to view videos. Nowadays the viewers often wish to watch videos on their smartphones or laptops in places, that usually provide less than perfect viewing environment (e.g. lighting, external distractions).

https://www.doi.org/10.24132/CSRN.3201.36

2. SUBJECTIVE QUALITY EVALUATION

The evaluation process

The quality tests that follow the abovementioned recommendations require to gather volunteers to act as viewers and present the videos to be evaluated in a certain way, usually in a controlled environment, that is free from external distractions. There are several main types of quality evaluation procedures, and the most important ones are presented below.

Single stimulus (absolute category rating) in which the viewers are shown a single sequence and are asked to evaluate its quality in a 1 to 5 scale, 1 being "bad" and 5 being "Excellent". Each viewer is asked to evaluate many videos in such a way. Since the results of such a simple evaluation depend on the sequence content in a broad sense, the test is usually augmented by adding a hidden reference to the test to normalize the results. Such a reference would, for example, be the original, uncompressed video.

A more reliable method is a double stimulus one, where the viewer is shown two sequences and asked to evaluate the quality. One of the sequences that are shown is a reference (i.e. original, uncompressed) and the other one is the evaluated sequence. Depending on the variation of the method, the viewer does know or does not know which sequence is the reference one. In the first case, the viewer evaluates the impairment of the evaluated sequence in the scale of 1 (annoying artifacts) to 5 (imperceptible differences) (DSIS: double stimulus impairment scale), in the second case, the viewer evaluates the comparative quality of the second video in the scale of -3 (much worse quality) to +3 (much better quality) (DSCS: double stimulus comparison scale).

The result of the evaluation can be presented in a form of an average quality (MOS – mean opinion score) and with the confidence interval calculated with the use of the standard deviation of the results.

For DSIS, the results can be directly used to arrange a set of many different sequences with respect to their quality, while for DSCS there is no such direct possibility. This should make the DSIS to be the superior evaluation model, however, there are some inherent problems with this model, that will be discussed below.

Challenges in the process

The methods described above pose some challenges. First of all, a significant number of viewers has to be involved in the evaluation process, usually well above 10 participants are required. The participants should not be experts in video processing, since this could bias their evaluation. The participants need to be trained so that they know what kind of distortions to expect and where to put their attention during the viewing. Also, the whole process of evaluation needs to be explained. The evaluation needs to be performed in a somewhat controlled environment, and this limits the number of people that can view and evaluate the sequences at once. The viewing is usually time consuming, as the viewers are usually expected to evaluate significant number of sequences. Especially in double stimulus scenarios this consumes a lot of time and is very arduous for the viewers.

In some circumstances it is extremely difficult to gather that many people willing to spare a significant amount of their time to perform the evaluation. In the recent years it has become even more challenging, due to the sustained pandemic situation. Therefore, the idea to perform the evaluation on-line emerged, that will be presented in the paper.

Another challenge is interpretation of the results. The wide scale of the evaluation poses a significant risk for the viewers, since it is really difficult to decide whether to give the just viewed video the mark of 4 or 5. Also, there is a question whether the video is slightly worse, worse or much worse than the previously seen one. Those are difficult questions that the viewers have to consider, and the result is not easily predicted. Any attempts to show the viewers what artifacts to look for and how to judge them is only biasing the viewers and should be avoided.

Therefore, the interpretation of the results is difficult. The results can be influenced by the order of the videos during the evaluation, since the viewers may start to modify the marks given to consecutive videos. As long as the number of viewers is not high enough, those factors may significantly bias the results, even when random order of sequences is used.

3. THE PROPOSED METHOD

Motivation

During the pandemic, it is difficult to gather viewers to evaluate sequences. Even before the pandemic it was difficult, since the viewers can rarely expect to be paid for their time spent on the evaluation. Therefore, an online tool is required to perform the evaluation.

Also, the time required for the evaluation by a single user needs to be decreased. In the ITU recommendations it is suggested that a single session should be kept as short as possible, the suggested limit being 20 minutes for short sequences. In the contemporary research the test sequences are usually short, in the order of 10-20 seconds each, so the advised limit of 20 minutes does apply.

This limit means that about 20 pairs of sequences can be shown to a single viewer. This may cause fatigue for the viewer and also means that significant amounts of data need to be transferred to the viewer. Therefore, we suggest to limit the number of sequences shown to a single viewer. This way we are able to recruit more volunteers. The most of the time is spent on training and explaining the judging procedure. This can easily be done online for all the participants at once. The viewing can be done later by each viewer individually, therefore not requiring that much time from the viewers and no waiting for their viewing turn.

We also observe the problem with the marks that was signaled earlier – many users hesitate about the rating they should give and their judgement can easily saturate during the test or be adjusted halfway through the test. To alleviate those problems, we suggest to use the DSCS scheme but to make the judging procedure much easier and significantly limit the number of possible marks.

To summarize this section, the motivation for developing the described system was threefold: to increase the number of recruited viewers, to make the judging simpler and to make the whole process less time consuming, both for the viewers and for the researchers.

Existing solutions

The problem of the subjective quality evaluation is not new, therefore there exist several solutions for the remote evaluation of subjective quality of video. A review of such solutions is presented in [Uhr20a]. The author present their own solution of such a system as well. Some other systems for remote evaluation of videos are described in [Jai13], where a system is developed that enables an online gathering of voting results and is available for download and use. Another system for evaluation of video quality is presented in [Rai13]. This system is very close to the system described in our paper, however it seems that it does not standardize the coding format of the videos and therefore can only be used to evaluate the quality for video coders for which a plug-in or codec pack already exists. It is therefore not suited for research on new codecs or new, non-standard modifications of existing codecs. No comments on possible use for any kind of video processing (like post processing of videos) are given. Another advanced system is described in [Che10]. This system is fully based on an Adobe Flash technology, that is outdated and not supported any more.

The comparison of the remote and local video quality assessment results are presented in the [Uhr20b]. The comparison of the results of the same evaluations performed remotely and locally show very high correlation of the MOS values and authors claim that the remote quality evaluation can replace the laboratory tests.

Many of the systems mentioned above are, unfortunately, either not accessible any more or use outdated technology (like Adobe Flash scripts).

Our system is developed to be able to evaluate any kind of video processing technique, including postprocessing or any kind of modification. It is meant to work based entirely on web browser, not requiring any extensions nor codecs. It is also designed so that no access to commandline on the server is required and a simple web server services with a database access are sufficient.

The implementation

The schematic of our proposed system is shown in Figure 1.

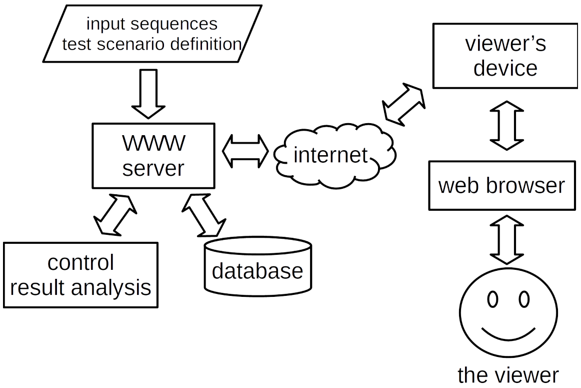


Figure 1. The schematic of the proposed system.

From the point of view of the viewer, the whole system is based on the web browser. The videos to be viewed are sent from the web server to the viewer's device and presented to the viewer in the correct order. The viewer is expected to evaluate the sequences by selecting the proper mark in the browser. The rating, along with the information about the user, such as the reported screen resolution of the user's device, is sent back to the server and is stored in the database for further analysis. Also the basic information about the viewer are stored: the age and sex of the viewer.

We decided that in order to make the whole process for the viewers as nonintrusive as possible, we should aim to prepare the whole system to use a web browser as an interface. This means that the user does not have to install any software in order to take part in the evaluation. Although this is a significantly limiting factor (a custom-made application would provide far more possibilities), we believe that the tradeoff is in favor of the browser based solution. Our choice also means that no non-standard libraries nor codecs should be required and the videos have to be encoded in a way that enables most of the web browsers to decode them natively.

In fact, video sequences that are on the web server are in compressed form. An important factor for this compression is not to introduce any artifacts to the videos. Therefore the coder should provide a lossless or nearly-lossless compression. The lossless video compression, even for very short clips (10 seconds long) produces very large amounts of data, therefore the nearly-lossless compression is preferred from the point of view of the feasibility of the system. The results of the tests show that the use of nearly-lossless compression should significantly reduce the amount of data to be sent to the user while the difference in perception of the data before and after such compression is expected to be insignificant.

The review of possible codecs was performed and the final choice was to use the VP9 codec using a WebM container. Currently, the lossless coding is used, although the system can accept nearly-lossless coded videos readily. VP9 supports both coding methods.

The system is prepared in the even more popular and capable JavaScript language and the React library. The database system used is MySQL.

The fact that the system is fully dependent on internet browser results in some difficulties caused by incompatibility of some platforms that require special approach, but, on the other way, allows many different platforms that are compatible to be used during evaluation. Still, the browser approach seems to be superior to preparing applications for wide variety of platforms, especially for smartphones. The evaluation can, in principle, be done on a smartphone and personal computer. This provides a wide variety of screen resolutions, viewing distances and screen sizes, together with different surroundings. One can even expect to get results from people that were performing the session in means of public transport on a loud street. Such a possibility means that we get results for the actual surroundings in which the user is usually watching video content. This seems to be a significant advantage of the proposed method.

To differentiate between the most important types of devices, the system stores the screen resolution reported by the browser.

The evaluation process

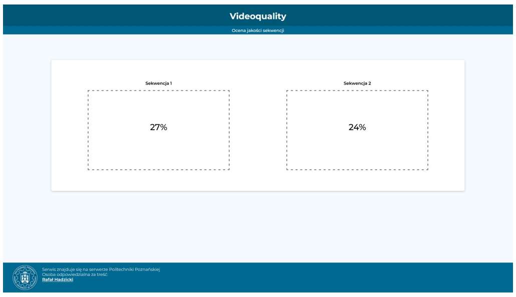
When user enters the website dedicated to the system, the basic information about the age and sex of the viewer is gathered. Then, a specific set of sequences to be shown to the user are randomly chosen.

Next, the pairs of videos are shown and the user is asked to evaluate the comparative quality of the videos. The pairs of videos are shown on full screen. This is a problematic feature, since there are compatibility issues between different systems and web browsers that still need to be addressed separately for some combinations. Before the playback of the pair of sequences, they are buffered in the viewer's device. This is done in order to decrease the probability of pauses during the playback. This is another challenging issue, since even the compressed streams are large (hundreds of megabytes each) and the need to download and buffer them poses a set of challenges for the network connection as well as the buffer memory for the web browser on the viewer's device. The challenge is much smaller if almost-lossless compression for the evaluated videos are used.

In the current form, the system is used to perform a direct comparison between two sequences and only two choices are given – the viewer is asked to select the video with better quality from the two shown. This makes the process much simpler for the viewer, since a simple question needs to be answered. It needs to be stressed that videos of the same quality are never shown, therefore there always are differences between the two sequences. The same approach is suggested in [Che10] and agrees well with our observations described above.

Each user can perform multiple sessions and since the videos for each session are randomly selected, such an approach is welcome and provides additional data.

The screenshots of the consecutive pages of the developed webpage as seen on a personal computer using a 1920x1080 screen are presented in Figures from 2 to 5. First, the data is buffered to avoid stalls during the playback. At this stage, the page displays the progress, as shown in Figure 2.



Videoquality

Ocena jakości sekwencji

Sekwencja 1

27%

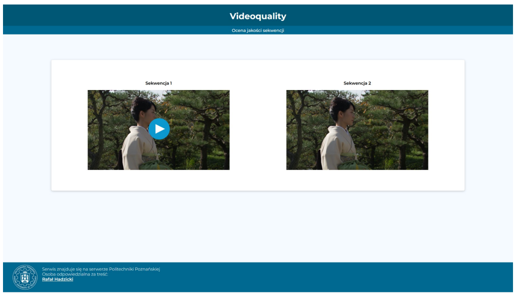
Sekwencja 2

24%

Serwis znajduje się na serwerze Politechniki Poznańskiej
Osoba odpowiedzialna za treść:
Rafał Hadzicki

Figure 2. Loading the video data.

When the data is buffered, the first video can be played. Playing of the second video is not possible now. When the play button is clicked (see Figure 3), the video is shown on the full screen.



Videoquality

Ocena jakości sekwencji

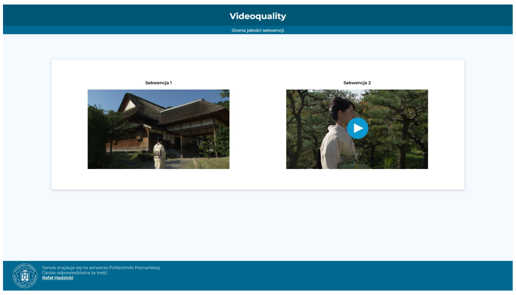
Sekwencja 1

Sekwencja 2

Serwis znajduje się na serwerze Politechniki Poznańskiej
Osoba odpowiedzialna za treść:
Rafał Hadzicki

Figure 3. Data loaded - ready to show first video.

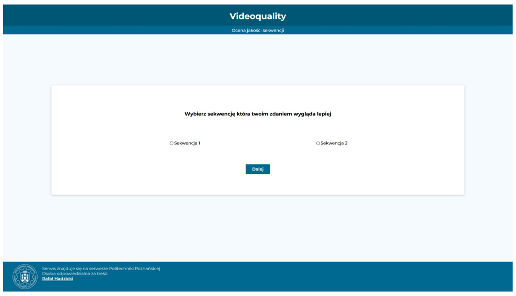
After viewing the first video, the second video can be played. The play button is shown on the second video and replaying the first video is not possible (see Figure 4).



Serwis znajduje się na serwerze Politechniki Poznańskiej
Osoba odpowiedzialna za treść:
Rafał Hadzicki

Figure 4. Ready to show the second video.

Directly after viewing the second video, the voting starts. The voting page is shown on Figure 5. After voting, the new pair of videos is loaded, or, when the last pair was graded, the session ends and a "thank you" page is displayed.



Wybierz sekwencję która twoim zdaniem wygląda lepiej

○ Sekwencja 1 ○ Sekwencja 2

Serwis znajduje się na serwerze Politechniki Poznańskiej
Osoba odpowiedzialna za treść:
Rafał Hadzicki

Figure 5. Voting.

Result analysis

After a significant number of viewers finish their voting sessions, the results can be analyzed.

For cases when viewers select a sequence with a better quality, the direct comparison of the results is not straightforward and cannot be performed by a simple comparison of values, as it is the case for DSIS method. A more sophisticated method is required. Here, we adopt the method for result analysis using the appropriate preference matrix, similar to the one used in [Che10].

The chosen way of grading the videos does not, at this time, allow to order a number of sequences with respect to their perceived quality. Only comparative results are possible. This, however does not necessarily limit the usability significantly. Many times one is interested to compare two methods directly and either demonstrate the superiority or inferiority of one method, or prove the equal performance of two different methods. For such cases the developed system seems to be perfectly suited.

4. TEST OF THE SYSTEM

Test scenario

For a test scenario used to verify the developed system we use a set of HD sequences compressed with HEVC encoder using different QP indices. The actual encoder used was the reference implementation HM version 16.6. The QP values used were 22, 27, 32 and 37. The original (i.e. uncompressed) sequences are not used in the survey. Two test sequences were used: Kimono and ParkScene. Those sequences are a part of the set of sequences recommended by ITU-T and ISO/IEC for research on video coding [Bos12]. Every user is asked to select the better quality video for two pairs of videos. One pair of videos is two randomly selected encoded videos of Kimono sequence, the second pair is two randomly selected videos of ParkScene sequence. The sequences are transmitted to the viewers' device in a form of losslessly compressed streams. The sequences are displayed at full screen, therefore are resized to the display resolution by the web browser.

Results

The test involved gathering 70 sets of votes. This means that 140 pairs of videos were rated and the better one for each pair was selected.

The results for the Park Scene sequence are shown in Table 1.

		Chosen as better (QP)			
		22	27	32	37
Not chosen as better (QP)	22		8	4	0
	27	8		7	2
	32	6	7		0
	37	9	11	8	

Table 1. Results for the ParkScene sequence.

The results from the Table 1 are interpreted in the following way. When the pair of sequences compressed with QPs of 32 and 22 were shown to the viewers, 6 times the sequence with QP=22 was selected as better (blue frame in Table 1) and 4 times the sequence with QP=32 was selected as better (green frame in Table 1). The pair 22-32 was shown 10 times, and 6 times out of 10 the sequence with QP=22 was selected as the better one. The numbers below the main diagonal correspond to the cases when the sequence with the lower QP was selected as the better one (and, in general, that is the expected result). For ParkScene this happened 49 out of 70 times, while 21 times the video with higher QP was selected. The results for the Kimono sequence are shown in Table 2.

		Chosen as better (QP)			
		22	27	32	37
Not chosen as better (QP)	22		1	2	5
	27	8		3	0
	32	12	10		0
	37	11	10	8	

Table 2. Results for the Kimono sequence.

It can be seen that for the Kimono sequence the users most of the times selected the sequence with lower QP (59 out of 70 times). It may seem that for the Kimono sequence the differences between different QP values are more visible.

The interesting comparison can be done when analyzing the results for cases when a regular computer and a smartphone was used. The results for smartphones are shown in Table 3, and the results for regular PC (usually FullHD or bigger resolution screen) are shown in Table 4.

		Chosen as better (QP)			
		22	27	32	37
Not chosen as better (QP)	22		3	3	3
	27	3		4	1
	32	2	1		0
	37	2	3	1	

Table 3. Results for smartphones (both sequences).

		Chosen as better (QP)			
		22	27	32	37
Not chosen as better (QP)	22		6	3	2
	27	13		6	1
	32	16	16		0
	37	18	18	15	

Table 4. Results for computers (both sequences).

The ratio between the cases when the lower QP is chosen over higher QP to the total cases is 12 over 26 (only 46% of cases), while for computers this ratio is 96 over 114 (84% of cases).

Such results are not surprising, since it can be expected that for smartphones the differences in quality of the sequences compressed with different QP may not be visible at all. The screen can simply be to small to notice the differences easily. We can, for example, notice, that when Kimono sequence encoded with QP of 37 was compared to the sequence with QP of 22 (a really surprising choice!), only in 5 cases the QP37 sequence won. Out of those 5 cases, 3 cases were the smartphone users. The two remaining cases may be regarded as mistakes, since it is really difficult to believe that when viewing on a big screen the viewers would not notice the significant artifacts for the QP 37 case.

Unfortunately, such mistakes or deliberate actions cannot be avoided entirely and especially in short viewing sessions it is not possible to filter them out.

Confidence interval calculation

The results presented above are given only for a certain sample of the population. It is expected, therefore, that the ratios calculated for the results from a sample of population may differ from the ratio calculated hypothetically for the entire population. In order to measure the confidence of the calculated results one needs to perform statistical evaluation of the results. For the example described above, the only possible choices for the user are better/worse quality within a pair of sequences, therefore the viewing results may be regarded as the results of a binomial trial (Bernoulli trial). For such cases, there are established methods for calculating the confidence interval, as explained in [Ros03]. The method of choice of estimating the confidence intervals for the conducted experiments is the “exact” Clopper-Pearson method, due to its popularity and robustness. An example below is given for the case when the quality of the Kimono sequence compressed with QP 22 is compared to the quality of the sequence encoded with QP 32. From Table 2 we can see that the QP22 sequence is chosen as the better one in 12 cases and the QP32 is chosen in 2 cases. Therefore the proportion of cases when the lower QP produces a sequence that is regarded to have a higher quality is 12/14 = 85,7%. The estimate of the confidence interval at the 95% confidence level would be from 57.19% to 98.22%. Since the lower bound is higher than 50%, we can, at this level of confidence, say that majority of population would perceive the QP22 case as that of a higher quality than QP32.

The confidence interval in this case is quite wide, and this alone supports the idea about performing tests at as high a number of viewers, as possible. For example if the results were 120 cases in 140 cases total, the confidence interval would shrink to from 78,8% to 91,05%. Our system makes it much easier to gather such number of marks.

5. SUMMARY

In the paper we presented an idea and an implementation of the system for remote evaluation of subjective quality of video. The system enables the grading to be performed in the real life situations, for example on smartphones in places where people actually watch videos, and also on personal computers. This, however, may be perceived as a drawback – inability to fully control the environment and the viewing method, but this, we believe, is offset by the real life experience during the tests.

The system makes it much easier to gather significant number of grades for the videos, when compared to traditional (“face to face” or “local“) viewing sessions. The system is configurable, any desired method of double stimulus judging can be implemented, although the main idea behind the system was to implement a “binary” method of selecting a better sequence among the two shown.

The results are available in real time, even during the ongoing tests. The results can be gathered any time and the basic statistics can be calculated.

The important feature of the system is that it is coder agnostic. The raw videos are encoded to a common format (lossless VP9 in WebM container) and therefore does not require any external codecs nor applications. Most of the contemporary web browsers supporting JavaScript are compatible with the system.

The main drawbacks of the system are concerned with the huge amounts of data, in the form of the test sequences, that not only need to be stored on the server, but also transmitted to the viewers. This limits the possible number of the sequences used during the test and limits the overall test length for a single viewer (some people are not prepared to download hundreds of megabytes of test video streams at once).

6. FUTURE WORK

The most important changes of the system, required for any further development and widespread use, would be to limit the amount of data that needs to be stored and transmitted. The only viable option here is to use a nearly-lossless compression. This, however, requires further study to choose the proper settings.

The compatibility of the system needs to be improved, especially in relation to the iOS system.

If the ordering of the quality of several sequences is required, the method similar to the Transitivity Satisfaction Rate principle described in [Che10] can be tried in the processing of the results.

Further developments, regarding the test scenario configuration flexibility, test sequences storage and the overall security and reliability of the system, are envisaged in the near future. The ability to perform different test scenarios in parallel is one of the other possible modifications.

7. ACKNOWLEDGMENTS

The work was supported by the Ministry of Education and Science, Poland.

8. REFERENCES

[Bos12] F. Bossen, Common test conditions and software reference configurations, Joint Collaborative Team on Video Coding (JCT-VC) of ITUT SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. JCTVC-J1100, Stockholm, Sweden, July 2012.

[Bt500] ITU-R Rec. BT.500-14, Methodologies for the subjective assessment of the quality of television images, ITU: Geneva, Switzerland, 2019.

[Che10] K. Chen, C. Chang, C. Wu, Y. Chang and C. Lei, Quadrant of euphoria: a crowdsourcing platform for QoE assessment, IEEE Network, vol. 24, no. 2, pp. 28-35, March-April 2010, doi: 10.1109/MNET.2010.5430141.

[Jai13] A. K. Jain, C. Bal and T. Q. Nguyen, Tally: A web-based subjective testing tool, 2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX), 2013, pp. 128-129, doi: 10.1109/QoMEX.2013.6603224.

[P910] ITU-T Rec. P910, Subjective video quality assessment methods for multimedia applications, ITU: Geneva, Switzerland, 2021.

[P911] ITU-T Rec. P.911, Subjective audiovisual quality assessment methods for multimedia applications, ITU: Geneva, Switzerland, 1998.

[P913] ITU-T Rec. P.913, Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment, ITU: Geneva, Switzerland, 2021.

[Rai13] B. Rainer, M. Waltl and C. Timmerer, A web based subjective evaluation platform, 2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX), 2013, pp. 24-25, doi: 10.1109/QoMEX.2013.6603196.

[Ros03] T.D. Ross, Accurate confidence intervals for binomial proportion and Poisson rate estimation, Computers in Biology and Medicine, vol. 33, Issue 6, 2003, pp. 509-531, ISSN 0010-4825, doi: 10.1016/S0010-4825(03)00019-2.

[Uhr20a] M. Uhrina, A. Holesova, Development of web-based crowdsourcing framework used for video quality assessment, 2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA), 2020, pp. 718-723, doi: 10.1109/ICETA51985.2020.9379172.

[Uhr20b] M. Uhrina, J. Bienik, T. Mizdos, QoE on H.264 and H.265: Crowdsourcing versus Laboratory Testing, 2020 30th International Conference Radioelektronika (RADIOELEKTRONIKA), 2020, doi: 10.1109/RADIOELEKTRONIKA49387.2020.9092424.

ISSN 2464-4617 (print)
ISSN 2464-4625 (DVD)
Computer Science Research Notes
CSRN 3201
WSCG 2022 Proceedings

Towards an Automated System for Reverse Geocoding of Aerial Photographs

Christoph Praschl[1], Michael Stradner[1],
Yuta Ono[2], and Gerald Zwettler[1,3]

[1]{christoph.praschl, michael.stradner, gerald.zwettler}@fh-hagenberg.at

[2]g236s001@s.iwate-pu.ac.jp

[1]Research Group Advanced Information Systems and Technology, Research and Development Department, University of Applied Sciences Upper Austria

[2]Graduate School of Software and Information Science, Iwate Prefectural University Japan

[3]Department of Software Engineering, School of Informatics, Communications and Media, University of Applied Sciences Upper Austria

ABSTRACT

Aerial photographs of buildings are often used as memorabilia sold by trading companies. Such photographs come with an issue regarding the address of the shown buildings, since the recording location of the camera may be known, but shows a spatial distance to the actual subject of the image. In addition to that, also this recording location is often not known in detail but only roughly in the form of the flight route/area. To address this problem, a methodology for reverse geocoding is proposed, allowing to identify the position of buildings that are photographed from aerial vehicles. This is done using a process for extending recording locations and a second process based on the registration of invariant features within aerial shots compared to maps.

Keywords
Aerial Photography, Reverse Geocoding, Building, Segmentation

1 INTRODUCTION

Aerial shots of private buildings are a kind of memorabilia, that is typical for european countries, especially Austria. For this, trading companies are flying across the country with helicopters, airplanes or even drones making those unique products. Especially, in the case of the first two vehicles, a pilot is accompanied by an additional photographer. This photographer is typically using some system camera in combination with a telephoto lens and is responsible for creating the photos. To do so, the photographer sits in the respective aerial vehicle and tries to capture a good clipping of the target buildings from the vehicle's side window by adapting the recording angle (as far as possible) and especially by changing the zoom level. Depending on the area, respectively the number of buildings at the specific location, multiple clippings may be overlapping and show related information like neighborhoods. After the flight, salespersons are trying to reference the photographs with maps to identify the exact addresses and like this be able to contact the owners of the buildings as basis for a sales pitch. Especially, this manual reference process still requires a lot of time and work due to the situation that although the position of the aerial vehicle is known, the actual viewing direction of the photographer as well as the position and with this the addresses of the focused buildings are unknown. Next to already globally referenced photographs, such trading companies often also have huge archives of never sold images, for which not even the exact recording position but only the rough flying route/area is known. To tackle these problems, the present work introduces a methodology for reverse geocoding aerial shots.

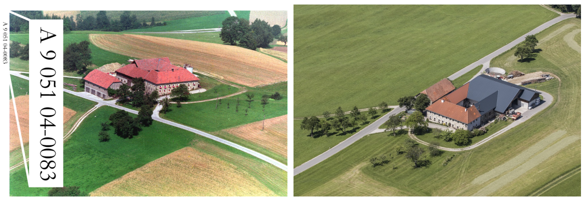
2 MATERIAL

The proposed methodology is developed in reference to the image archive of our project partner. This archive consists of hundreds of images, which partially contain coordinates of the Global Positioning System (GPS) [HWLC12] related to the recording positions. Next to images with GPS information, there are also many images with an artificial annotation, referencing a physical map or a textual protocol, where the flight area/route is recorded. One sample for each case is shown in Figure 1.

3 METHODOLOGY

In this work a methodology is proposed based on two independent sub processes, with the first process used for images, where the recording location is known, and the second process, where only the rough flight

https://www.doi.org/10.24132/CSRN.3201.37



(a) (b)

Figure 1: Samples images of the used aerial photography archive showing (a) a not geo-referenced image from 2004 with a textual note at the left corner (zoomed in for better visibility) and (b) an image from 2018 for which the recording location is known in the form of EXIF data.

route/area is known, which leads to time-intensive calculations. This methodology is shown in Figure 2.

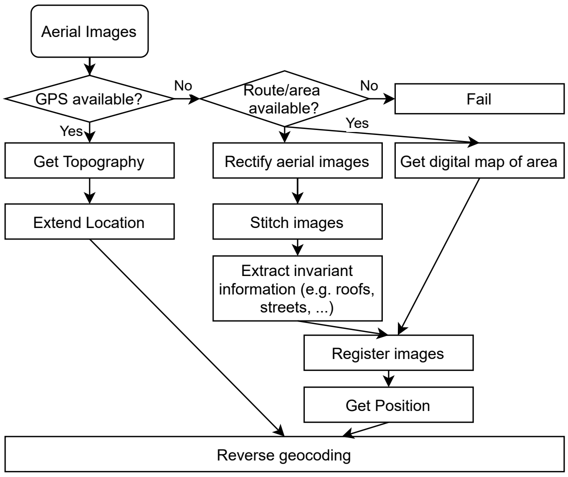


Figure 2: The overall process used for geocoding aerial images. Depending on the situation, if the images contain GPS information, the process either tries to reference the photographed buildings based on a topography model of the area or based on a registration process in reference to digital map services.

3.1 Extending Recording Location

For images, where the recording location is known in the form of GPS information, the reverse geocoding is done based on a ray tracing approach that tries to find the position and with this the address of the building based on the position of the helicopter, its flight height and a topography model of the surrounding area. The topography model is required because of the deviating ground level of the photographed building due to e.g. hills compared to the sea level, which is used as reference of the flight height. This setup is shown in Figure 3.

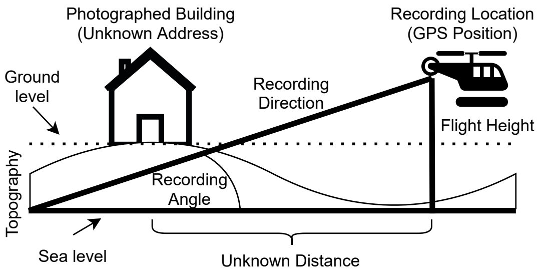


Figure 3: Record situation showing the known record position, the flight height and the unknown distance as well recording angle.

The GPS positions of the recording locations during a flight allow estimating the route of the helicopter/airplane using splines. In combination with the knowledge, that the photographer either takes the images from the left or the right side window of the aerial vehicle, the possible recording direction can be estimated orthogonal to the flight route as shown in Figure 4. This recording direction can be represented as global bearing angle β, with 0° representing the magnetic north and 180° pointing to the south. Finally, the recording angle can also be estimated to around $45^\circ \pm 10^\circ$, due to the situation that the photographer tries to create a clipping, where not only the roof but also the building itself is visible, so a direct overflight or a too flat angle are unsuitable. Knowing both, the flight height fh and the recording angle α, allows calculating the distance d between the aerial vehicle and the targeted position on sea level, like:

$$d = fh/tan(\alpha) \tag{1}$$

Utilizing the earth radius e, the bearing angle β and the recording location defined by its latitude lat and longitude lng the targeted position can be calculated, which is defined by lat_2 and lng_2, as:

$$\begin{aligned} lat_2 = asin(sin(lat) * cos(d/e) + \\ cos(lat) * sin(d) * cos(\beta)) \end{aligned} \tag{2}$$

$$a = sin(\beta) * sin(d/e) * cos(lat) \tag{3}$$

$$b = cos(d/e) - sin(lat) * sin(lat_2) \tag{4}$$

$$lng_2 = lng + atan2(a,b) \tag{5}$$

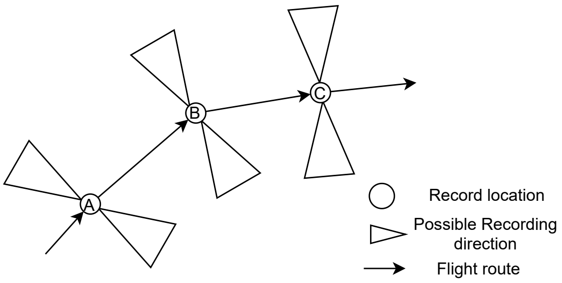


Figure 4: Based on the known recording locations of aerial shots, the flight route of the aerial vehicle can be reconstructed, allowing to estimate the recording direction.

Knowing both the recording position and the targeted position, allows to sample-wise determine the position of the photographed building. This is done by creating n sample points with a distance d/n starting from the recording position towards to the target position. Based on the topography model, the geo-referenced position of the photographed building can be reconstructed by finding the best matching intersection of the sample positions and the trace between recording and target position. Knowing the GPS position of the building, a reverse geocoding service like Google Maps [Sve10] can be used to provide suitable addresses to the salesperson.

3.2 Comparison with digital map services

Next to the tracing based approach, the second process is intended for images where only the rough flight route/area is known. This process is based on the idea that multiple aerial shots show a partially overlapping area with more or less invariant information such as streets, roofs or even natural features such as rivers. Within a registration process these invariant information is compared to a digital map to identify the location of the photographed area.

3.2.1 Rectification

In a first step, the aerial images are rectified by projecting the image plane onto the ground plane, as shown in Figure 5. This step is required to approach a bird's eye view, comparable to the digital map used in the registration step. This is done based on the image meta information in the form of its width w, height h, the estimated recording angle α and the field of view angle δ of the used camera lens. Based on this information, the focal length fl in pixel and the vertical field of view f_v as well as the horizontal one f_h can be calculated with the subsequent formulas:

$$fl = sqrt(w^2 * h^2)/2 \tag{6}$$

$$f_h = atan(w/2/fl) * 2 \tag{7}$$

$$f_v = atan(h/2/fl) * 2 \tag{8}$$

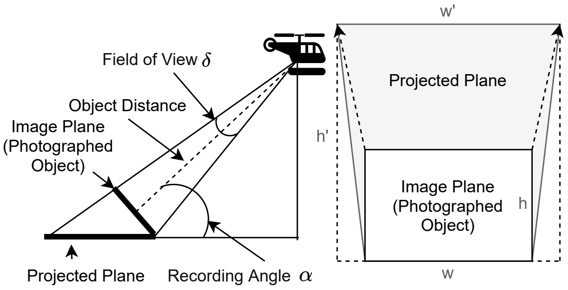


Figure 5: Projection of the image plane showing the photographed building to the ground plane within a rectification process.

Using these values in turn allows to calculate the general projection information like the height h' of the projected image using the following trigonomic equations:

$$\varepsilon = 90 - \alpha \tag{9}$$

$$\lambda = (180 - \delta)/2 \tag{10}$$

$$\mu = 180 - \lambda \tag{11}$$

$$\omega = 180 - \mu - \varepsilon \tag{12}$$

$$p = h * sin(\varepsilon)/sin(\omega) \tag{13}$$

$$h' = sqrt(h^2 + p^2 - 2 * h * p * cos(\mu)) \tag{14}$$

Using this information, also the width w' of the projected image can be calculated like:

$$e = a/sin(\delta/2) \tag{15}$$

$$d = p + e \tag{16}$$

$$w' = sin(f_h/2) * d * 2 \tag{17}$$

Knowing both the target height and width allows calculating the respective projection angle Ω for every row defined by y within the range [0, h] like:

$$g = abs(y - h/2) \tag{18}$$

$$\phi = atan(g/fl) \tag{19}$$

$$\Omega = \left\{ \begin{array}{ll} \phi + (f_v/2), & \text{for } y > h/2 \\ f_v/2, & \text{for } y == h/2 \\ (f_v/2) - \phi, & \text{for } y < h/2 \end{array} \right\} \tag{20}$$

Based on Ω the actual pixel mapping $I(x,y) \rightarrow T(x',y')$ for the source image I to its projection T can be applied using multiple sub steps as shown below:

$$\zeta = 180 - \Omega - \lambda \tag{21}$$

$$\gamma = 180 - \zeta \tag{22}$$

$$\rho = 180 - \gamma - \varepsilon \tag{23}$$

$$p = y * sin(\varepsilon)/sin(\rho) \tag{24}$$

$$y' = sqrt(y^2 + p^2 - 2 * y * p * cos(\mu)) \tag{25}$$

$$f = sqrt(e^2 + y^2 - 2 * e * y * cos(\lambda)) \tag{26}$$

$$w'' = sin(f_h/2) * (p + f) * 2 \tag{27}$$

$$u = sqrt((p + f)^2 - (w''/2)^2) \tag{28}$$

$$o = abs(x - w/2) \tag{29}$$

$$\Phi = atan(o/fl) \tag{30}$$

$$i = u * tan(\Phi) \tag{31}$$

$$x' = \left\{ \begin{array}{ll} (w''/2) - i, & \text{for } x < w/2 \\ (w''/2) + i, & \text{for } x >= 2/2 \end{array} \right\} \tag{32}$$

3.2.2 *Image stitching and extraction of invariant visual features*

After the rectification of the spatially connected aerial shots, the images are stitched together to create a virtual map. To do so, a SIFT [Low99] operator is used to extract scale invariant features in the images that are compared using a brute force feature matcher [Nob16]. Based on the retrieved information of this feature matching process, a homography matrix is determined, that is used to warp the individual images, so they can be stitched together [BL07].

Using the stitched image, invariant visual features such as streets, roofs or rivers can be extracted. To do so, one or multiple segmentation models such as U-Nets [RFB15] can be used to separate the background pixels from the desired foreground pixels, representing the mentioned invariant information.

3.2.3 *Registration*

Using both the segmentation mask for the stitched aerial shot as well as a digital map of the flight area, a registration process can be used to identify the geo-referenced position of the photographed buildings. This registration is done using an euclidean distance map $\mathscr{D}_{euclid}$ with P and P'' representing the set of pixels of invariant features. The rigid registration problem is defined using the mean-squared error (MSE) metric as

$$\begin{aligned} P' = Trans(&P, \theta_{best}, T_{x_{best}}, T_{y_{best}}, \\ &Sc_{x_{best}}, Sc_{y_{best}}, Sk_{x_{best}}, Sk_{y_{best}}) \end{aligned} \tag{33}$$

where the best transformation parameters lead to minimal squared distances between P' and P''. These best parameters are determined in a discrete search space with rotation $\theta \in [-\theta_{min}; \theta_{max}]$, translation along x-axis $T_x \in [-T_x; T_x]$, translation along y-axis $T_y \in [-T_y; T_y]$, as well as x-scaling $Sc_x \in [-Sc_x; Sc_x]$, y-scaling $Sc_y \in [-Sc_y; Sc_y]$, x-skweness $Sk_x \in [-Sk_x; Sk_x]$ and y-skweness $Sk_y \in [-Sk_y; Sk_y]$.

$$\begin{aligned} &\theta_{best}, T_{x_{best}}, T_{y_{best}}, Sc_{x_{best}}, Sc_{y_{best}}, Sk_{x_{best}}, Sk_{y_{best}} = \\ &\underset{\theta,T_x,T_y,Sc_x,Sc_y,Sk_x,Sk_y}{\operatorname{argmin}} \sum_{i=1}^{|P|} (\mathscr{D}_{euclid}(P'')[Trans(P, \theta, T_x, T_y, \\ &\qquad Sc_x, Sc_y, Sk_x, Sk_y)[i]])^2 \end{aligned} \tag{34}$$

The discrete search space thereby comprises $k = 11$ steps, i.e. radius $r = 5$, for each of the seven variables $(\theta, T_x, T_y, Sc_x, Sc_y, Sk_x, Sk_y)$ and for each of the $m = 10$ optimization runs according to the scale factor s_i with $[-r*s_i, -(r-1)*s_i, ..., 0, ..., (r-1)*s_i, r*s_i]$ as search offset for globally optimal parameters from the last entire run. To move from global to local search with increasing number of optimization runs performed, the search space scale factor is reduced with $s_i = s_{i-1} * 0.9$ and initial s_1 defined from image resolution.

4 IMPLEMENTATION

The presented methodology is implemented using Python [vR95] and the OpenCV library [Bra00]. For the extraction of invariant landscape features in the form of streets, a U-Net model has been trained using Tensorflow [AAB$^+$16]. Due to the lack of a suitable training dataset for segmented aerial photographs, this convolutional neural network is trained using 7500 RGB satellite images from a digital map service. Despite the different perspectives of satellite images compared to aerial photographs, both capture methods represent bird's eye views of the recorded area and are intended as interchangeable in the sense of a transfer learning methodology [WKW16] for the proposed area of application. As ground truth for the streets, the corresponding map views of these satellite images are used. The used training dataset contains map sections with a size of 1920 × 1080 px from different zoom levels. Based on this base dataset, multiple augmentation strategies are applied to further increase the amount of images and with this the variety of image properties regarding e.g. the brightness. For this task, a random selection of up to five augmentation methods per image is applied, allowing to vertically and/or horizontally flip, scale, rotate, translate, blur the image or changing its brightness, contrast, saturation or hue.

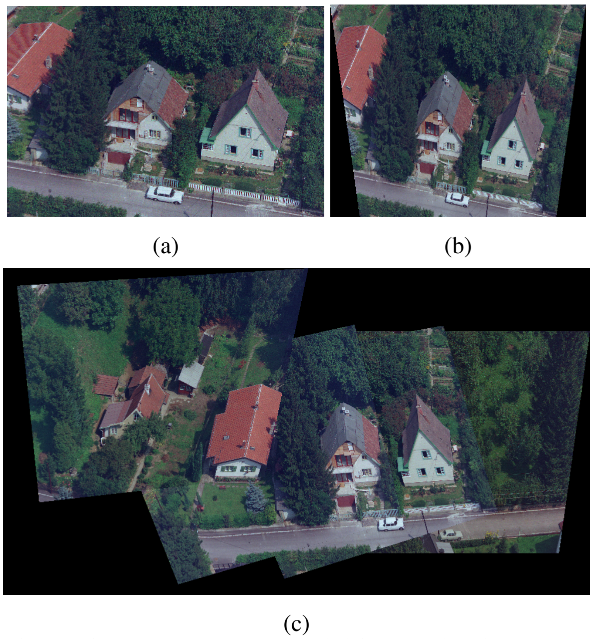
5 RESULTS

For the evaluation of the first sub process based on the extension of the recording location, an image subset of 36 subsequent aerial shots is used from our project partner's archive. Using the GPS coordinates of these images allows reconstructing the flight route and to retrieve the address of the photographed buildings. This process is tested using Google Maps as digital map service for the final reverse geocoding step and the Open Elevation API[1] for the topography model. This sub process is manually evaluated by comparing the proposed address with its real world counterpart. The addresses are identified correctly for 14 of the 36 shots. For the remaining images, the reversed geocoded address deviates from the real one with in a range of some streets to completely wrong positions.

Next to that, the individual steps of the second approach are also evaluated. First, the rectification and the image stitching steps are evaluated using a sequence of four partially overlapping aerial shots, as shown exemplary in Figure 6. Additionally, also the extraction of invariant features is tested using a U-Net model, that allows to segment aerial photographs as proposed. Applying this segmentation model allows to extract invariant information in the form of streets, as shown in Figure 7. Based on such invariant features, the registration approach can

[1] https://open-elevation.com/

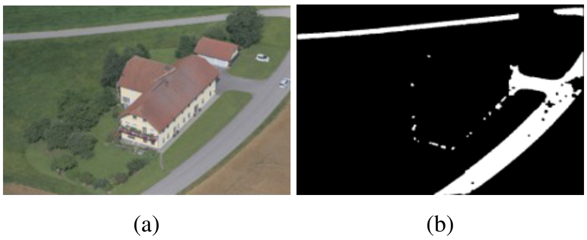
be used to automatically find the best matching position within multiple map sections in questions. Such reference images can for example be retrieved from Google Maps for the known flight area, as shown in Figure 8.



(a) (b)

(c)

Figure 6: (a) One sample aerial image that is (b) rectified using the proposed approach and (c) stitched together with three additional rectified images.



(a) (b)

Figure 7: (a) An aerial shot for which the U-Net segmentation model is applied to create (b) a street binary mask

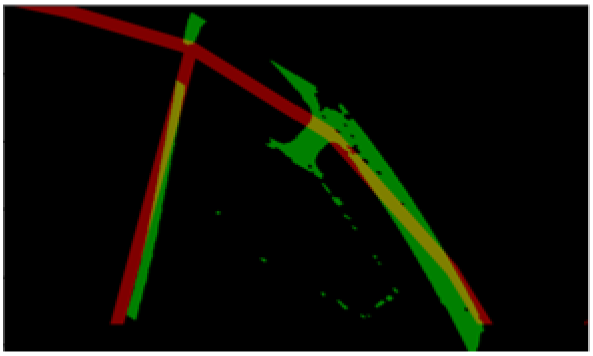


Figure 8: Overlayed binary masks showing streets for a requested area from a digital map service (red) and the registered segmentation result (green) for the aerial shot shown in Figure 7b.

6 RELATED WORK

Jaimes and Castro [JC18] describe a comparable method for the rectification of aerial images. In comparison to our approach, the authors also take the exact orientation (pitch, roll and yaw) of the aerial vehicle into account, which is unknown in our use case.

Nowak [Nov92] compares methods such as rectification for the creation of digital orthophotos. As in our approach, the rectification is done based on the projection of the image plane onto the photographed ground plane and its correction based on the landscape of the photographed area. In contrast to our work, the author also includes exterior orientation parameters (c.f. Jaimes and Castro [JC18]).

Additionally, also Cheng et al. [CYT00] compare approaches in the context of rectifying remote sensing images utilizing polynomial trend mappings, multiquadric interpolation functions, and their own proposed ordinary kriging technique. By taking the spatial structure variation of the terrain into account, the authors are able to outperform the two other approaches. In contrast to our rectification method, the accuracy and with this also the calculation complexity of this anisotropic spatial modeling approach is not required and would require exact knowledge of the photographed area, which is not known.

Allison and Muller [AM93] describe a method for geocoding aerial images. In comparison to our work, the authors have registered multi-spectral images and focus on a pixel-wise registration process. This accuracy is not required for the reverse geocoding of aerial images of buildings used by salesperson.

Karel et al. [KDV$^+$13] present a method for georeferencing aerial photographs in the context of archaeological applications. To do so, the authors propose a rectification approach. Like in the image stitching step of our approach, the authors try to automatically find the relative orientation of the aerial photographs using scale invariant keypoints to combine the available visual information. The so oriented images are then geo-referenced for reconstructing a sparse point cloud. In contrast to our work, the authors are using their approach for creating a 3D reconstruction in the form of an orthophoto map of the photographed scenery and are not doing a reverse geocoding, since the position of the region of interest is known for such archaeological applications.

Long et al. [LJH$^+$15] present a generic framework for the rectification in the context of remote sensing. Like in our registration approach, the idea of this framework is the utilization of invariant, landscape features. The authors present a feature extraction method allowing to find visual features in landscapes including points, straight lines, free-form curves and areal regions, that are used for the rectification process. In contrast to our work, the authors neither use invariant landscape features for the registration of images, nor for a reverse geocoding process, but for the automatic rectification.

7 CONCLUSION AND OUTLOOK

The proposed methodology shows promise for a semi-automated reverse geocoding process for aerial shots allowing to decrease the required amount of time for salespersons to identify the address of buildings shown in such images. In the future, we plan to extend our tests and improve the current results, especially for the recording location extension approach. Additionally, a retraining of the created segmentation model and the training of additional models for other invariant information such as roofs or rivers is planned.

ACKNOWLEDGMENT

Our thanks to our research partners of the Amido HandelsgesmbH, especially the owner Andreas Ratzenberger, for the close cooperation and the efforts. Next to that, we also thank the province of Upper Austria for facilitating the research project with the easy2innovate funding program as well as the Austrian Research Promotion Agency with the Innovationscheck+ program.

REFERENCES

[AAB+16] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467, 2016.

[AM93] David Allison and Jan Muller. An automated system for sub-pixel correction and geocoding of multi-spectral and multi-look aerial imagery. Int. Archives of Photogrammetry and Remote Sensing, 1993.

[BL07] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. International journal of computer vision, 74(1):59–73, 2007.

[Bra00] G. Bradski. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.

[CYT00] Ke-Sheng Cheng, Hui-Chung Yeh, and Chang-Hsuan Tsai. An anisotropic spatial modeling approach for remote sensing image rectification. Remote Sensing of Environment, 73(1):46–54, 2000.

[HWLC12] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, and James Collins. Global positioning system: theory and practice. Springer Science & Business Media, 2012.

[JC18] B Jaimes and C Castro. Perspective correction in aerial images. 10.13140/RG.2.2.34885.29926, 2018.

[KDV+13] Wilfried Karel, Michael Doneus, Geert Verhoeven, Christian Briese, Camillo Ressl, and Norbert Pfeifer. Oriental: Automatic geo-referencing and ortho-rectification of archaeological aerial photographs. In XXIV International CIPA Symposium, volume 2, pages 175–180, 2013.

[LJH+15] Tengfei Long, Weili Jiao, Guojin He, Zhaoming Zhang, Bo Cheng, and Wei Wang. A generic framework for image rectification using multiple types of feature. ISPRS Journal of Photogrammetry and Remote Sensing, 102:161–171, 2015.

[Low99] David G Lowe. Object recognition from local scale-invariant features. In Proc. of 7th IEEE int. conf. on comp. vision, 1999.

[Nob16] Frazer K Noble. Comparison of opencv's feature detectors and feature matchers. In 2016 23rd Int. Conf. on Mechatronics and Machine Vision in Practice (M2VIP), 2016.

[Nov92] Kurt Novak. Rectification of digital imagery. Photogrammetric engineering and remote sensing, 58:339–339, 1992.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Int. Conf. on Medical image computing and computer-assisted intervention, 2015.

[Sve10] Gabriel Svennerberg. Beginning google maps API 3. Apress, 2010.

[vR95] Guido van Rossum. Python reference manual. Department of Computer Science [CS], (R 9525), 1995.

[WKW16] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. Journal of Big data, 3(1):1–40, 2016.

POSTER: Fast and Precise Binary Instance Segmentation of 2D Objects for Automotive Applications

Ganganna Ravindra, Darshan
CMORE Automotive GmbH
Germany, D-88131 Lindau
darshangr4293@gmail.com

Dinges, Laslo
Otto-von-Guericke-University
Germany, D-39016 Magdeburg
Laslo.Dinges@ovgu.de

Ayoub, Al-Hamadi
Otto-von-Guericke-University
Germany, D-39016 Magdeburg
Ayoub.Al-Hamadi@ovgu.de

Baranau, Vasili
CMORE Automotive GmbH
Germany, D-88131 Lindau
vasili.baranov@gmail.com

ABSTRACT

In this paper, we focus on improving binary 2D instance segmentation to assist humans in labeling ground truth datasets with polygons. Humans labeler just have to draw boxes around objects, and polygons are generated automatically. To be useful, our system has to run on CPUs in real-time. The most usual approach for binary instance segmentation involves encoder-decoder networks. This report evaluates state-of-the-art encoder-decoder networks and proposes a method for improving instance segmentation quality using these networks. Alongside network architecture improvements, our proposed method relies upon providing extra information to the network input, so-called "extreme points", i.e. the outermost points on the object silhouette. The user can label them instead of a bounding box almost as quickly. The bounding box can be deduced from the extreme points as well. This method produces better IoU compared to other state-of-the-art encoder-decoder networks and also runs fast enough when it is deployed on a CPU.

Keywords

Extreme points, IoU, Encoder-Decoder, Instance binary segmentation.

1 INTRODUCTION

Visual recognition tasks are currently active research topics in the areas of autonomous driving, biomedical image processing, and scene understanding. To accomplish automatic visual recognition based on deep learning, training a deep neural network to learn to extract features from images is essential. To do this there is a need for a lot of annotated training data. For this, manual labeling is used, which is very time-consuming. It includes locating manually the precise positions, drawing bounding boxes, assigning labels, and drawing polygons around the objects. Hence, this process is tardy, requires a lot of manpower for labeling enormous data and human labelers are prone to errors as well. The objective of this paper was to develop a fast and precise system that can perform binary instance segmentation so that human labelers are significantly assisted in the task of manual 2D semantic segmentation of road-scene objects.

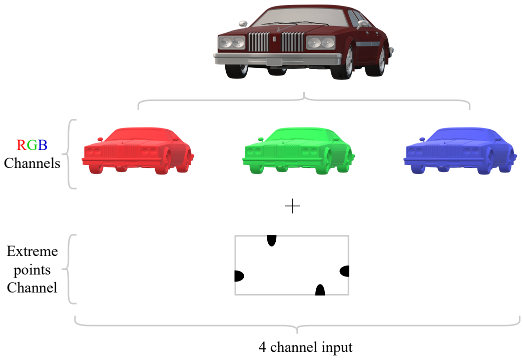


Figure 1: Extreme points channel along with RGB channels.

Instance segmentation task is simultaneously solving both tasks of object detection and semantic segmentation [Khe17, Hua19], it is needed to locate in an image every object from a predefined set of classes as well as to give a binary mask for each of these objects. Other flavor of instance segmentation is binary segmentation [Ron15], here it is only a single object that has to be segmented, i.e. pixels are classified as foreground and background because the segmentation happens only in the object bounding box, which has to be obtained from a detection algorithm or manual annotation.

Initial manual annotation of data is crucial for training deep learning models. A typical manual annotation is drawing polygons around each object of interest. It will be possible to significantly assist human labelers in the task of manual 2D semantic segmentation (i.e. specifying polygons around each object of interest in a scene), if we develop a fast and accurate system-a

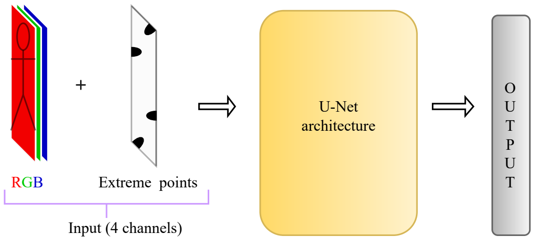


Figure 2: Extreme points input in the U-Net architecture.

neural network-for binary instance segmentation (foreground/background). The network shall produce a silhouette of an object (e.g., a car or a person) when a user specifies a corresponding 2D bounding box.

In order to speedup manual drawing of polygons around objects, this paper investigates state-of-the-art encoder-decoder architectures for binary instance segmentation and proposes a method with extreme points as shown in Figure 1. By providing only the bounding box or extreme points (see below), one can generate polygons through our binary instance segmentation algorithm inside the C.LABEL labeling tool developed by CMORE Automotive GmbH (where the present research was performed). The proposed method significantly assists human labelers in the task of manual 2D semantic segmentation of road-scene objects by automatically generating polygons around each object of interest in an image. This paper also focuses on deploying the proposed method on a normal CPU rather than using it in a GPU (human annotators often do not have access to GPUs) such that it shall run fast enough and also occupy little hard drive and RAM space. In this way, it can be easily deployed with the labeling tool. Our target was inference time $\leq$ 200 ms on a CPU.

2 NETWORK ARCHITECTURE

Our approach is built upon the encoder-decoder architecture with skip connections, U-Net [Ron15]. The U-Net architecture consists of a encoder path to capture the context, a symmetrical decoder path that enables accurate object detection, and skip connections [Ron15]. The main idea of encoder-decoder networks is to supplement a usual encoder network by successive layers, where pooling operators are replaced by upsampling operators [Ron15]. This results in increase in the resolution than the encoded representation. The high-resolution features from the encoder path are additionally combined with upsampled decoder features [Ron15]. The decoder network can learn to produce more accurate output based on the information from skip connections [Ron15]. The output of the model is the binary mask of the object in an image.

We modified the U-Net architecture in several ways: (i) by using depth-wise separable convolutions instead of convolutions [Cho17]; (ii) by using residual blocks [Hez16] instead of normal convolutions; (iii) by using dense blocks [Hua17] instead of normal convolutions; (iv) by using contextual convolutions [Ono18]; and (v) by using semantically similar skip connections [Zho18] in order to improve the performance with U-Net. These modifications didn't lead to significant segmentation quality improvements under our inference time restriction ($\leq$ 200 ms on a CPU), that's why we explored providing more input data to the network.

To provide more input data to the network, we pass extreme points information [Man18] to the network along with the normal RGB input (RGB is a standard choice) (cf. Figure 1 and Figure 2). The main intention is to increase the precision of the segmented masks by letting the network use extra information. The extreme points are left, right, top, bottom –most pixels of the object, i.e. the points where the object touches its bounding box. The technique of using extreme points for binary segmentation was introduced in [Man18]. That paper demonstrates the networks accepting RGB channels + extreme points perform better than equivalent networks that accept RGB channels only. Extreme points are represented with a binary mask in the additional channel. In the original paper, a 2D Gaussian is applied to each of these points in the binary mask. In this work we used drawing circle around every point. Marking extreme points is as fast or faster as marking bounding boxes. That is why we can rely on this information in our work.

The use of an additional channel along with the input 3-channel image in the U-Net architecture [Ron15] is shown in Figure 2. The convolution architecture is the same as U-Net but the basic filter depth used is $f = 16$ in the first level. The filter size is progressively increased in the encoder as: f, $2f$, $4f$, $8f$, and $16f$. Similarly, the filter size in decoder is decreased from $16f$ as $8f$, $4f$, $2f$, and f. The input size is proportionally decreased in every level in the encoder using max pooling operation as 128×128, 64×64, 32×32, 16×16, and 8×8. In the same way in decoder the input size is increased using upsampling operation from 8×8 as 16×16, 32×32, 64×64, and 128×128. The final output size is 128×128.

3 TRAINING

The network is trained from scratch with a total 83,403 instances for training and 15,926 instances for validation from the Cityscapes dataset [Cor16], simultaneously for the 9 most prevalent classes from the dataset (“car”, “traffic sign”, “bicycle”, “person”, “rider”, “motorcycle”, “traffic light”, “truck”, and “bus”). The instances are re-scaled to 128×128 pixels before feeding into the network.

We choose Intersection Over Union (IoU) and "border error" as the evaluation metric. The "border error" is the per-pixel distance from the predicted object boundary to the ground truth boundary. For comparison, we calculate different flavors of IoU, such as average IoU (aIoU- the IoU that is calculated for each validation batch and then averaged for all the batches to produce aIoU of that epoch), mean IoU (mIoU- the IoU that is calculated by averaging class-based IoUs), and instance IoU (iIoU- the IoU that is calculated by averaging IoUs calculated per single instance) as introduced in [Kir19]. The extreme points channel is created during training as an additional channel with a binary mask and passed to the network along with a RGB channels. We apply the following data augmentations to instances during training: flip and rotation. We use several loss functions: differentiable IoU loss as introduces in [Kir19], combination of the IoU loss and binary cross entropy [Igl18], and our custom loss function that approximates the average distance error (see below). Our custom loss function showed better results than other loss functions. The average distance error is the line integral $\int_{border} EDT(s)ds/L$, where EDT is the Euclidean Distance Transform and L is the ground truth boundary length. We approximate L as $\sqrt{A_g}$, and the approximate average distance error $\overline{\Delta d}$ is given by

$$\overline{\Delta d} = \frac{A_u - A_i}{\sqrt{A_g}}, \qquad (1)$$

$$\overline{\Delta d} = \frac{|y \cup y'| - |y \cap y'|}{\sqrt{A_g}}, \qquad (2)$$

where A_u and A_i is the area of union and area of intersection of the groundtruth and predicted masks respectively, y is the predicted pixel probabilities, y' is the groundtruth binary labels, A_g is the area of the groundtruth boundary, $\cap$ is the intersection operation and $\cup$ is the union operation.

During validation and during training, the average IoU (aIoU) is calculated as the metric. After a network is trained and the best epoch is selected, we calculated the mean IoU (mIoU), instance IoU (iIoU), and per-pixel distance. The aIoU, mIoU, iIou, and per-pixel distance (border error) are calculated in images scaled to the network input size (128 × 128 pixels). Data augmentation techniques are used only during training (metrics are calculated over the validation dataset after each epoch). During evaluation (testing the model after training) over the validation dataset, data augmentation techniques are switched off to calculate per-pixel distance, mIoU, and iIoU. The discussed approaches are trained on one Nvidia GeForce GTX 1080 Ti GPU and are deployed during inference on an Intel(R) Core(TM) i5-6300U CPU for measuring the time taken for segmenting objects.

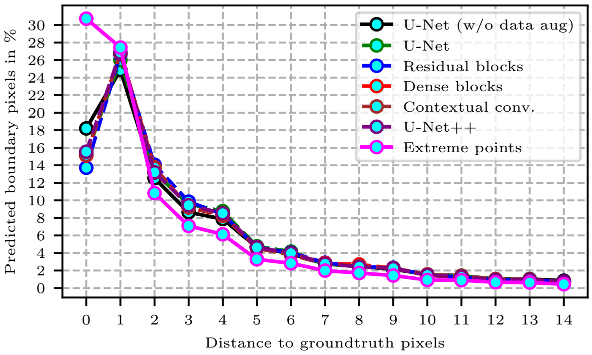


Figure 3: Comparison of per-pixel distance distributions for the discussed approaches.

4 RESULTS

The network is trained from scratch with the "Adam" optimizer and the learning rate of 1×10^{-3} using the average distance loss function. In this method, the network is trained for 100 epochs with a batch size of 32, while we observed in experiments that training starts to converge after 18–20 epochs and the best results are obtained within this range. We have used early stop criteria and selected the best model. The per-pixel distance is calculated over the validation dataset after training, using the best model from the network.

The comparison of the border error in different U-Net variants studied is shown in Figure 3. The approach with extreme points has a higher probability of error at 0 px, which implies a higher accuracy of the model. In Table 1, the comparison of aIoU for the best models over the validation dataset shows that the approach with extreme points performs better than other approaches used in this work with, **89.16%** IoU. The mIoU and iIoU, which are calculated after training (data augmentation is switched off) are **90.65%** and **87.25%** for the network with extreme points, respectively. These results evidently show that adding extreme points leads to higher quality of segmentation. The comparison of per-class mIoUs of the most interesting classes for the discussed U-Net variants trained with data augmentations and the average distance loss is shown in Table 2. The table also demonstrates that the approach with extreme points produces the best results for all the displayed classes. In Figure 4 one can see a qualitative evaluation with examples of the important car and person classes.

We found out that all the architecture modifications except the extreme points one perform almost identically to the basic U-Net architecture. If inference time is kept fixed, changes in the architecture do not seem to be able to significantly improve network accuracy. But, providing additional user input (extreme points) allowed the network to produce significantly better results, on the other hand.

Approach	*aIoU*(%)	*iIoU*(%)	*mIou*(%)	Inference Time (ms)
U-Net	84.45	86.38	83.36	~140
Residual blocks	85.11	86.47	83.88	~180
Dense blocks	85.88	86.99	84.92	~320
Contextual conv.	85.12	86.75	83.70	~240
U-Net++	85.47	86.10	82.73	~380
Extreme points	**89.16**	**90.65**	**87.25**	~140

Table 1: Comparison of performance metrics of U-Net variants trained with the average distance loss

Approach	*car*	*bus*	*bicycle*	*person*	*rider*	*motor-cycle*
U-Net	90.39	90.22	77.09	80.43	75.30	76.98
Residual blocks	90.42	89.91	76.68	80.97	75.58	76.63
Dense blocks	90.52	91.01	77.20	81.13	75.77	77.04
Contextual conv.	90.57	91.07	76.72	80.41	75.95	75.50
U-Net++	90.85	91.25	77.27	81.29	75.66	76.69
Extreme points	**92.49**	**93.26**	**81.16**	**84.92**	**80.08**	**81.54**

Table 2: Comparison of per-class mIoUs for U-Net variants trained with the average distance loss

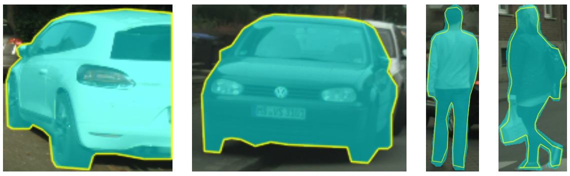


Figure 4: Results of our method for the car and person class (turquoise surface is ground truth, yellow contour the prediction).

5 CONCLUSION

Our method with extreme points in the U-Net [Ron15] architecture achieves good performance and clearly outperforms the other approaches discussed in this paper. Our approach achieves 3.7 points gain of aIoU, 4.5 points gain of iIoU and 4.5 points gain of mIoU over the basic U-Net architecture. The inference time for binary instance segmentation is the same for the basic U-Net and U-Net with extreme points approaches, which is about 140 ms. In this work, we also introduced a new custom loss function that matches the per-pixel error slightly better than the differentiable IoU loss. This improved architecture is already being integrated into the manual annotation tool of CMORE Automotive GmbH with deep learning assistance capabilities, C.LABEL. For commercial use, the network was retrained on an internal dataset.

One possible approach to further improve the accuracy of segmentation results is to use Generative Adversarial Networks (GANs) [Goo14], where the generator shall produce a binary segmentation and the discriminator shall distinguish true segmentation masks from generated ones. The discriminator shall essentially learn the evaluation metric on its own.

6 REFERENCES

[Cho17] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” 2017 IEEE Conference on CVPR.

[Cor16] M. Cordts, M. Omran, and et al., “The cityscapes dataset for semantic urban scene understanding,” in Proc. of the IEEE Conference on CVPR, 2016.

[Goo14] I. Goodfellow, J. Pouget-Abadie, and et al., “Generative adversarial nets,” in Advances in Neural Information Processing Systems 27. Curran Associates, Inc., 2014, pp. 2672-2680.

[Hez16] K. He, X. Zhang, and et al., “Deep residual learning for image recognition,” 2016 CVPR. IEEE Conference on CVPR.

[Hua19] Z. Huang, L. Huang, and et al., “Mask scoring r-cnn,” 2019 IEEE Conference on CVPR, 2019.

[Hua17] G. Huang, Z. Liu, and et al., “Densely connected convolutional networks,” 2017 IEEE Conference on CVPR.

[Igl18] V. Iglovikov, S. Seferbekov, and et al., “Ternausnetv2: Fully convolutional network for instance segmentation,” 2018 IEEE/CVF Conference on CVPRW, 2018.

[Khe17] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask r-cnn,” 2017 IEEE International Conference on Computer Vision (ICCV), 2017.

[Kir19] A. Kirillov, R. Girshick, and et al., “Panoptic feature pyramid networks,” 2019.

[Man18] K.-K. Maninis, S. Caelles, and et al., “Deep extreme cut: From extreme points to object segmentation,” 2018 IEEE/CVF Conference on CVPR.

[Ono18] D. Onoro-Rubio and M. Niepert, “Contextual hourglass networks for segmentation and density estimation,” 2018.

[Ron15] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” MICCAI 2015, pp.234-241, 2015.

[Zho18] Z. Zhou, M. M. Rahman Siddiquee, and et al., “Unet++: A nested u-net architecture for medical image segmentation,” Lecture Notes in Computer Science, pp. 3-11, 2018.

POSTER: Image Resizing Level Impact on Eye Fundus Optic Disc and Optic Cup Segmentation

Sandra Virbukaite
Institute of Data Science and Digital Technologies
Vilnius University
Akademijos str. 4
08663, Vilnius, Lithuania
sandra.virbukaite@mif.vu.lt

Jolita Bernataviciene
Institute of Data Science and Digital Technologies
Vilnius University
Akademijos str. 4
08663, Vilnius, Lithuania
jolita.bernataviciene@mif.vu.lt

ABSTRACT

Optic disc (OD) and Optic Cup (OC) segmentation play an important role in the automatic assessment of eye health where the Convolutional Neural Networks (CNNs) have been extensively employed. The application of CNNs requires identical image size to work properly but the eye fundus images vary due to different datasets. In this paper we evaluate eye fundus image resizing level impact on OD and OC segmentation. For this evaluation we apply the most popular medical images segmentation autoencoder named U-Net. The experiments demonstrate that OD and OC segmentation results are improved averagely by 5.5 percent resizing images to size of 512x512 than 128x128.

Keywords

OD and OC segmentation, Convolutional Neural Networks.

1. INTRODUCTION

Diseases such as glaucoma, diabetic retinopathy, hypertension can be diagnosed from biomedical images, more specifically from eye fundus images. Therefore, biomedical image analysis is required where an image segmentation is one of the initial steps. Manual image segmentation is a time-consuming task that is closely related to the expertise of the medical profession. This encourages researchers to develop fast and accurate solutions for automated image segmentation [Vir20]. Segmentation distinguishes and defines different objects in the image, thus classifying them into different object classes.

Solving image segmentation tasks by applying CNNs, the size of images plays an important role [Zhu21] as CNNs require identical image size but the size varies due to different datasets. Most of the lately proposed networks are working on the images of the same dataset but it has a poor segmentation accuracy for different fundus images datasets (Table 2. and Table 3.). To make the network to be working for different datasets, the mixed images strategy can be applied but then the alignment of image size is needed. In this paper we focus on evaluation of image resizing level and its impact on OD and OC segmentation as these are one of the key parameters in glaucoma identification.

2. RELATED WORK

Lately many CNN based methods have been proposed for OD and OC segmentation in fundus images (some achieved results are shown in Table 4.). Liu W et. al. [Liu20] presented an autoencoder for OD segmentation called Multi-level Light U-Net and Atrous Spatial Pyramid Pooling that aimed to implicate the significant spatial information in high-level semantic feature maps. The proposed Light U-Net (LU-Net) incorporates the encoder module of two max-pooling operations for down-sampling and the decoder consisted of two up-sampling operations. The reduction of convolutional layers and pooling operations helped to avoid the loss of the spatial information. Liu B. et. al. [Liu21] proposed a U-Net based autoencoder named Densely Connected Depth-wise Separable Convolution Network (DDSC-Net) for OD and OC semantic segmentation. The usage of depth-wise separable convolution layers reduced the amount of computation that made the proposed network to differ from the original U-Net. Sevastopolsky A. [Sev17] presented a universal approach based on U-Net for automatic OD and OC

https://www.doi.org/10.24132/CSRN.3201.39

segmentation. The modification presented in the paper contained less filters in all convolutional layers compared to the original U-Net and did not contain an increasing number of filters to reduce resolution. These changes made the architecture much lighter in terms of number of parameters and training time. Veena H.N. et. al. [Vee21] designed two separate CNN models composed of 39 layers for OD and OC segmentation to calculate the Cup-to-Disc-Ratio (CDR). The increased number of layers helped to extract more features and minimize the errors. According to the authors, one of the main problems of the existing CNN model is the change of image resolution during the model training process. That causes the loss of essential information. The authors proposed to solve this problem by adding an up-sampling layer for each down-sampling layer. By this approach the lost image resolution was recovered and the output image resolution was the same as the resolution of the input image. Gao J. et. al. [Gao20] proposed a Recurrent Fully Convolution Network (RFC-Net) for automatic joint segmentation of OD and OC. This network was able to capture high-level information, subtle edge information and minimize the loss of spatial information. The authors achieved an improvement of OD and OC segmentation performance by applying the polar transformation, multi-scale input and multiple output, four recurrent units and adding skip connections. Zhu Q., et. al., [Zhu21] proposed an encoder-decoder named GDCSeg-Net for OD and OC segmentation and applied mixed training strategy based on different datasets to solve image segmentation for different fundus image datasets problem that existing deep learning networks are facing. The network was conducted by a novel multi-scale weight-shared attention (MSA) module and densely connected depth-wise separable convolution (DSC) module. With these modules, OD and OC feature information was obtained more effectively and helped to improve the segmentation performance.

3. RESEARCH METHODOLOGY AND METHODS

To evaluate the impact of image resizing to the OD and OC segmentation results, we use the original U-Net network [Ron15] that is realized on several different scenarios:

1. The U-Net is trained on Drishti-GS (101 images of size 2047x1759) [Siv15] training dataset and tested on testing dataset of Drishti-GS, RIM-ONE v.3 (159 images of size 2144x1424) [Rim21] and Kaunas Clinics (39 images of size 1920x1440) [Kau21].

2. The U-Net is trained on the RIM-ONE v.3 training dataset and tested on testing dataset of RIM-ONE v.3, Drishti-GS and Kaunas Clinics.

3. The U-Net is trained on Kaunas Clinics training dataset and tested on testing dataset of Kaunas Clinics, RIM-ONE v.3 and Drishti-GS.

4. The U-Net is trained on dataset compiled from all these datasets and tested on testing datasets of RIM-ONE v.3, Drishti-GS and Kaunas Clinics separately.

The images are divided into training and test datasets. As the number of images in training datasets is too small, a data augmentation on each dataset is performed. By applying a random horizontal, vertical, and diagonal flip on each image by 20%, the number of images is increased to 3000 for each dataset that is used as training datasets. The test datasets consist of 39 original eye fundus images of each dataset. The images for training and test datasets are individually cropped by area of their OD (Fig 1.). The sizes of images vary from 340x340 to 1308x1308.

4. EXPERIMENT AND RESULTS

The experiments of OD and OC segmentation are performed on cropped images resized into pixel values of 512x512, 256x256 and 128x128 by bilinear interpolation. The learning rate of 0.1 and batch size of 3 are used. The training runs for 250 epochs. Table 4. shows the computation time on GPU 5 cores with 70 GB memory [ITR22] needed to train the network according to image size. The segmentation performance is evaluated by Dice score, which is defined as follow:

$$Dice = \frac{2TP}{2TP+FP+FN} \qquad (1)$$

where *TP* denotes true positive, *FP* – false positive and *FN* – false negative.

The results of Dice are presented in Table 1. for OD segmentation and in Table 2. for OC segmentation. The results of Dice of other methods are presented in Table 3. The visual comparison of OD and OC segmentation results of one scenario is shown in Figure. 2.

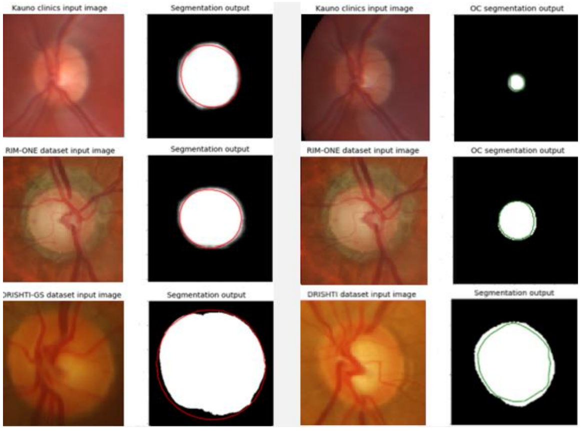

Kauno clinics input image
Segmentation output
Kauno clinics input image
OC segmentation output
RIM-ONE dataset input image
Segmentation output
RIM-ONE dataset input image
OC segmentation output
DRISHTI-GS dataset input image
Segmentation output
DRISHTI dataset input image
Segmentation output

Figure 2. Visual comparison of OD and OC segmentation, where red circle indicates the ground truth of OD, green - the ground truth OC.

Train dataset	Test dataset								
	Kaunas Clinics			Drishti			RIM-ONE		
	128x128	256x256	512x512	128x128	256x256	512x512	128x128	256x256	512x512
Kaunas Clinics	0.9213	0.9441	0.9530	0.7914	0.8249	0.8516	0.7514	0.7989	0.8285
Drishti	0.7028	0.7346	0.8075	0.9348	0.9659	0.9760	0.7125	0.7737	0.8248
RIM-ONE	0.7495	0.7710	0.8118	0.7112	0.7721	0.8242	0.9210	0.9558	0.9657
Mixed	0.8702	0.8913	0.9161	0.9116	0.9386	0.9528	0.8875	0.9093	0.9341

Table 1. OD segmentation results of the experiment.

Train dataset	Test dataset								
	Kaunas Clinics			Drishti			RIM-ONE		
	128x128	256x256	512x512	128x128	256x256	512x512	128x128	256x256	512x512
Kaunas Clinics	0.8698	0.8770	0.8861	0.4008	0.4297	0.4990	0.3918	0.4037	0.4448
Drishti	0.4015	0.4220	0.4905	0.8593	0.8765	0.9053	0.5368	0.5706	0.6066
RIM-ONE	0.4007	0.4114	0.4514	0.5223	0.5657	0.5930	0.8261	0.8769	0.9068
Mixed	0.7873	0.8231	0.8699	0.8006	0.8449	0.8990	0.7938	0.8304	0.8767

Table 2. OC segmentation results of the experiment.

Other methods	Input Image Resolution	Image resolution after resizing	REFUGE		Drishti		RIM-ONE	
			OD	OC	OD	OC	OD	OC
Liu W et. al., LU-Net [Liu20]	448x448	-	0.982	-	0.997	-	-	-
Liu B. et. al., DDSC-Net [Liu21]	480x480	240×240	0.960	0.890	-	-	-	-
Liu B. et. al., DDSC-Net [Liu21]	560x560	240×240	-	-	0.978	0.912	-	-
Sevastopolsky A., U-Net [Sev17]	256x256	128x128	-	-	-		0.95	
Sevastopolsky A., U-Net [Sev17]	512x512	128x128	-	-	-	0.85	-	0.82
Zhu Q., et. al., GDCSeg-Net [Zhu21]	512x512	-	0.964	0.894	0.974	0.900	0.956	0.824
Veena H.N. et. al., CNN [Vee21]	512x512	-	-	-	-	0.971	-	-
Veena H.N. et. al., CNN [Vee21]	128x128	-	-	-	0.988	-	-	-
Gao J. et. al., RFC-Net [Gao20]	400x400 - 900x900	512x512	-	-	0.9788	0.906	-	-

Table 3. OD and OC segmentation of other methods

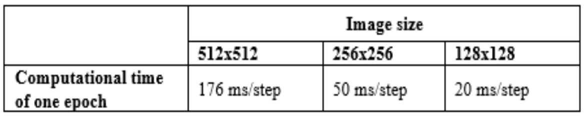


	Image size		
	512x512	256x256	128x128
Computational time of one epoch	176 ms/step	50 ms/step	20 ms/step

Table 4. Computation time to train the network.

CONCLUSION

The experiments indicate that the higher image resolution, the better OD and OC segmentation results are achieved but the cost of computation time to train the network on images of size 512x512 in comparison of images with size of 128x128 increased by 8 times. The highest Dice score is achieved by training the network on mixed images dataset with the images resized to 512x512. The Dice score of 0.9161 for OD and 0.8699 for OC is achieved on Kaunas clinics dataset, 0.9528 for OD and 0.8990 for OC on Drishti-GS, 0.9341 for OD and 0.8767 for OC on RIM-ONE. The bilinear interpolation is applied to resize the images but it causes the loss of OC boundaries. Due to this, the other interpolation methods such as nearest-neighbor or bi-cubic will be investigated in our future work to evaluate the impact of interpolation on emphasis of segment.

5. ACKNOWLEDGMENTS

We are thankful for the HPC resources provided by the IT Research Center of Vilnius University.

6. REFERENCES

[Ron15] Ronneberger, O., Fischer, P., Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation (2015).

[Liu20] Liu, W., Lei H., Xie, H., Zhao, B., Yue, G., Lei, B. Multi-level Light U-Net and Atrous Spatial Pyramid Pooling for Optic Disc Segmentation on Fundus Image, Springer, 2020.

[Liu21] Liu, B., Pan, D., Song, H. Joint optic disc and cup segmentation based on densely connected depthwise separable convolution deep network. BMC Med Imaging, 2021.

[Sev17] Sevastopolsky, A. Optic disc and cup segmentation methods for glaucoma detection with modification of U-Net convolutional neural network. Pattern Recognit. Image Anal. 27, 618-624, 2017.

[Zhu21] Zhu, Q., Xhen, X., Meng, Q., Song, J., Luo, G., Wang, M., Shi, F., Chen, Z., Xiang, D., Pan, L., Li, Z., Zhu, W. GDCSeg-Net: general optic disc and cup segmentation network for multi-device fundus images. Biomedical Optics Express, 2021.

[Vee21] Veena, H.N., Muruganandham, A., Senthil Kumaran, T. A novel optic disc and optic cup segmentation technique to diagnose glaucoma using deep learning convolutional neural network over retinal fundus images. Journal of King Saud University-Computer and Information Sciences, 2021.

[Gao20] Gao, J., Jiang, Y., Zhang, H., Wang, F. Joint disc and cup segmentation based on recurrent fully convolutional network. PLoS ONE 15(9): e0238983, 2020.

[Vir20] Virbukaite, S., Bernataviciene, J. Deep Learning Methods for Glaucoma Identification Using Digital Fundus Images. Baltic J. Modern Computing, Vol. 8, 2020.

[Siv15] Sivaswamy J, Krishnadas K. R, Joshi G. D, Jain Madhulika, Ujjwal and Syed Abbas T. Drishti-GS: Retinal Image Dataset for Optic Nerve Head (ONH) Segmentation. IEEE ISBI, Beijing, 2015.

[Rim21] RIME-ONE v.3 Dataset Homepage, http://medimrg.webs.ull.es/research/retinal-imaging/rim-one/, last accessed 2021.

[Kau21] Kaunas Clinics Dataset, a private dataset collected during the project "Development of a depersonalized eye fundus images database". Vilnius Regional Biomedical Research Ethics Committee permit No:158200-18/11-1057-572, last accessed 2021.

[ITR22] IT Research Center of Vilnius University, https://mif.vu.lt/lt3/en/about/structure/it-research-center#about-center, last accessed 2022.

A Deep CNN Model For Age Estimation

Beichen Zhang
Tokyo City University
1-28-1 Tamazutsumi
Setagaya-ku
158-8557, Tokyo, Japan
bzhang@tcu.ac.jp

Yue Bao
Tokyo City University
1-28-1 Tamazutsumi
Setagaya-ku
158-8557, Tokyo, Japan
bao@g.tcu.ac.jp

ABSTRACT

Age estimation from human faces is an important yet challenging task in computer vision because of the large differences between physical age and apparent age. Although many inspiring works have focused on the age estimation of a single human face through deep learning, the existing methods still have lower performance when dealing with faces in videos because of the differences in head pose between frames. In this paper, a combined system of age estimation and head pose estimation is proposed to improve the performance of age estimation from faces in videos. We use deep regression forests (DRFs) to estimate the age of facial images, while a multi-loss convolutional neural network is also utilized to estimate the head pose. Accordingly, we estimate the age of faces only for head poses within a set degree threshold to enable value refinement. First, we divided the images in the Cross-Age Celebrity Dataset (CACD) and the Asian Face Age Dataset (AFAD) according to the estimated head pose degrees and generated separate age estimates for images with different poses. The experimental results showed that the accuracy of age estimation from frontal facial images was better than that for faces at different angles. Further experiments were conducted on several videos to estimate the age of the same person with his or her face at different angles, and the results show that our proposed combined system can provide more precise and reliable age estimates than a system without head pose estimation.

Keywords
age estimation; deep learning; CNN; head pose estimation

1 INTRODUCTION

Age estimation from a facial image has become an important yet challenging problem in many applications, such as human-computer interaction[1], identification[2], security[5], and precision advertising[3].

In recent years, deep learning has made impressive works on various computer vision tasks[4], including age estimation[8, 7]. However, all these works have used datasets including only frontal facial images, which cannot adequately reflect the conditions of real-life applications. Different from most facial images in datasets, the head pose may vary greatly in videos or webcam streams, leading to intolerable errors in the estimated age.

In this work, a combined system of age estimation and head pose estimation is proposed to solve the problem of age estimation from faces in videos or webcam streams. First, we use deep regression forests (DRFs) [7] to estimate the age of facial images, which can achieve high precision for frontal facial images. Meanwhile, a multiloss convolutional neural network (CNN) is also utilized to estimate the head pose [9]. Then, we can use the trained system to estimate age and head pose from several videos frame by frame. When using the trained mapping between age and head pose, we set a degree threshold for the head pose and perform age estimation only for frames where the head pose is within this threshold to enable value refinement of the age estimated from the video.

2 RELATED WORK

For age estimation from faces in videos, the most closely related work is the Deep Age Estimation Model [11], in which Ji et al. use a CNN with an attention mechanism; Facial features are extracted by CNN then aggregated from features vectors to a single feature by an attention block. They trained the model using a new loss function leading to better precision and stability across every frames for age estimation. However, to guarantee stability, this model used continuous frames as input data rather than using a single image. In

https://www.doi.org/10.24132/CSRN.3201.40

addition, to train this model, a new dataset must be collected with labels annotation.

Another work for age estimaion that static and dynamic features can be learned from expressions of face simultaneously in videos called the Spatially-Indexed Attention Model (SIAM) [10]. In this model, Ji Pei et al. employ CNNs to extract the latent appearance features from each frame and then uses recurrent networks to process all the features to simulate time dynamics. However, this method has limitations in terms of which types of facial expression images it can consider; specifically, only smile and disgust databases were used in experiments.

3 PROPOSED METHOD

In this section, each step of the system flow will be explained in detail.

3.1 Age estimation

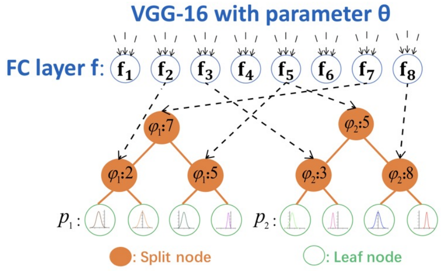


Figure 1: Illustration of a DRF.

Fig. 1 shows a diagram of a DRF [7].

A CNN combined with deep regression forests is introduced in this work and estimate the real age from facial image. The model is trained on facial image datasets with known ages and face landmarks as labels. The training process in this paper begins with the pre-trained weights from the ImageNet dataset, as with the same model used in [12]. Then, the CNN is fine-tuned on the two target datasets using for age estimation. The fine-tuning process make the CNN to obtain the features, distribution, and bias of each dataset and optimizes the performance.

The upper blue circles represent the output neurons from the CNN defined by the function **f** with parameter Θ. All these neurons come from the last fully-connected (FC) layer of VGG-16. The middle orange circles represent the split nodes and the bottom green circles represent the leaf nodes of deep regression forests. φ_1 and φ_2 represent the index functions of each tree. The black dashed arrows point out the correspondence from the split nodes of each tree to the neurons of VGG-16 FC layer. Each neuron may correspond to the split nodes of different trees. Each tree has its own distribution π for its leaf node(represented by the distribution curves on the leaf nodes). The final output for the whole forest can be calculated as the mix of the predictions of the individual trees. The parameter $f(\cdot;\Theta)$ and π will be trained simultaneously end- to-end.

3.2 Head pose estimation

We adopted Ruiz's method [9], in which deep multiloss CNNs are trained for head pose estimation with satisfactory accuracy. The ResNet50 networks [13] was introduced for headpose estimation and three losses are used for three angles separately. There are two parts of each loss: the mean squared error regressed directly and the cross-entropy loss from classification of pose. There are three FC layers being used for three angles and shared the previous parts of the network. By adopting additional cross-entropy losses from classification, we constructed three signals to be backpropagated to improve the learning process. The predictions of three output angles was computed as the final head pose results. The details of the architecture are shown in Fig. 2.

4 EXPERIMENTS

4.1 Testing on AFAD and CACD

In this section, the performance of DRFs for age estimation based on frontal and nonfrontal facial images is presented. The frequently used AFAD and CACD datasets, representing Asians and Europeans, respectively, were used in this experiment. We used the trained multiloss CNN to estimate the head poses in both datasets. For each facial image, three rotational angles were estimated, one on each axis. We set 30 degrees as the threshold for the sum of the three angles, and images with head pose angle estimates summing to more than 30 degrees were defined as nonfrontal images. Fig. 3 depicts exemplar images of nonfrontal facial images from the datasets.

On AFAD

Based on the estimated angles, AFAD was divided into frontal and nonfrontal subsets consisting of 53,983 and 5,361 images, respectively. Both subsets were randomly split into training/test (85%/15%) sets, and the training process was repeated 5 times with different random separation and the final outcome is the average of 5 times outputs. The quantitative results are summarized in Table 1. The results show that the accuracy of age estimation from frontal facial images is significantly better than that for nonfrontal images.

Subset	MAE
Frontal	3.73
Nonfrontal	4.97

Table 1: Performance (MAE) comparison on the frontal and nonfrontal subsets of AFAD [14]

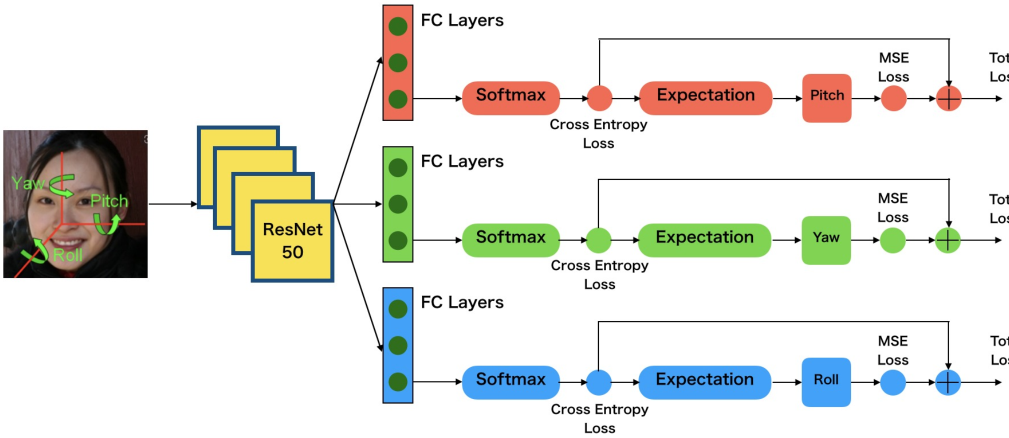


Figure 2: CNN with combined mean squared error and cross-entropy losses.

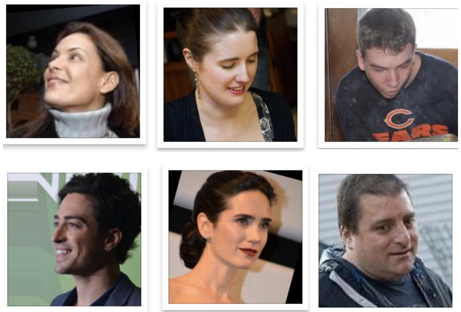


Figure 3: Examples of nonfrontal facial images.

On CACD

Based on the estimated angles, CACD was divided into frontal and nonfrontal subsets consisting of 15,145 and 3,026 images, respectively. Both subsets were randomly split into training/test (85%/15%) sets, and the training process was repeated 5 times with different random separation and the final outcome is the average of 5 times outputs. The quantitative results are summarized in Table 2. The results show that the accuracy of age estimation from frontal facial images is significantly better than that for nonfrontal images.

Subset	MAE
Frontal	4.59
Nonfrontal	5.65

Table 2: Performance (MAE) comparison on the frontal and nonfrontal subsets of CACD [6]

4.2 Testing on facial video datasets

Two new facial video datasets were constructed to evaluate our model in terms of age estimation performance. We collected 18,282 and 18,944 frames from two twelve-minute facial videos of Asian and European subjects, respectively. It should be noted that each facial video dataset was collected from the same person, and these datasets were used only for evaluating the age estimation models; currently, there is no facial video dataset available to be used for training the whole model. We first trained DRFs on AFAD and CACD, representing Asians and Europeans, respectively. Then, we tested the two trained models on the facial video datasets with simultaneous head pose estimation. Examples of the test images are shown in Fig. 4. We performed age estimation only for faces with head poses within 30 degrees, and we compared the results with the results for all images without head pose restrictions. Several other models were also trained on AFAD and CACD and then tested on the facial video datasets for more comprehensive comparisons.

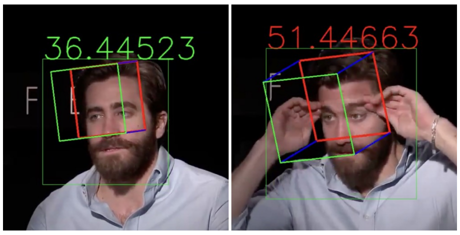


Figure 4: Examples from the facial video datasets with age and head pose estimates. The numbers represent the predicted age. Green and red colors indicate that the sum of the head pose rotational angles is less than and greater than 30 degrees, respectively.

On the Asian facial video dataset We trained a DRF on AFAD and tested the model on the Asian video dataset with head pose restrictions. We compared the results of our method with those of other outstanding age estimation models, and the quantitative results are summarized in Table 3. All models were trained on AFAD with the same training strategy to ensure fair comparisons. On the task of facial video estimation, our method achieves the best MAE of 5.12, and the variance is reduced by 0.62 compared to the best existing method.

Method	MAE	Variance
AlexNet [15]	6.19	6.92
DEX [16]	6.72	8.65
DRF [7]	5.96	4.12
Our method	**5.12**	**3.50**

Table 3: Accuracy (MAE) and variance results for comparison with state-of-the-art methods on the Asian facial video dataset

On the European facial video dataset We trained a DRF on CACD and tested the model on the European video dataset with head pose restrictions. We compared the results of our method with those of other outstanding age estimation models, and the quantitative results are summarized in Table 4. All models were trained on CACD with the same training strategy to ensure fair comparisons. On the task of facial video estimation, our method achieves the best MAE of 5.56, and the variance is reduced by 1.53 compared to the best existing method.

Method	MAE	Variance
AlexNet [15]	6.93	7.15
DEX [16]	7.17	8.22
DRF [7]	6.39	5.84
Our method	**5.56**	**4.31**

Table 4: Accuracy (MAE) and variance results for comparison with state-of-the-art methods on the European facial video dataset

5 CONCLUSIONS

In this paper, a combined system of age estimation and head pose estimation is proposed to solve the problem of age estimation based on faces in videos or webcam streams, where different head poses may lead to intolerable errors on the estimated ages. Experimental results show that with a head pose restriction such that age estimation is performed only for facial images with head poses within a specified degree threshold to ensure value refinement, our method achieves promising improvements in accuracy and stability for age estimation from video.

The main contributions of this paper are as follows: (1) We are the first to couple age estimation and head pose estimation for age estimation in videos. (2) Our method shows significantly improved performance in age estimation on facial video datasets compared to other state-of-the-art methods in terms of both accuracy and variance.

DATA AVAILABILITY

Links to datasets used in this paper.

CACD: `https://bcsiriuschen.github.io/CARC/`

AFAD: `https://afad-dataset.github.io/`

6 REFERENCES

[1] A. Lanitis, C. Draganova, and C. Christodoulou. Comparing different classifiers for automatic age estimation. *IEEE Transaction on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1):621-628, 2004.

DOI: 10.1109/TSMCB.2003.817091

[2] A. Lanitis, C. J. Taylor, and T. F. Cootes. Toward automatic simulation of aging effects on face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):442-455, 2002.

DOI: 10.1109/34.993553

[3] C. Shan, F. Porikli, T. Xiang, and S. Gong, editors. Video Analytics for Business Intelligence. *Studies in Computational Intelligence. Springer*, 2012.

[4] M. Stefan Ìczyk, and T. Bochen Ìski. Mixing deep learning with classical vision for object recognition. *Journal of WSCG*, 28(1-2): 147-154, 2020.

DOI: 10.24132/JWSCG.2020.28.18

[5] Z. Song, B. Ni, D. Guo, T. Sim, and S.Yan. Learning universal multi-view age estimator using video context. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 241-248, Nov 2011. 1 DOI: 10.1109/ICCV.2011.6126248

[6] B. Chen, C. Chen, and W. H. Hsu. Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. *IEEE Trans. Multimedia*, 17(6):804-815, 2015. DOI: 10.1109/TMM.2015.2420374

[7] W. Shen, Y. Guo, Y. Wang, K. Zhao, B. Wang, and A. Yuille. Deep Regression Forests for Age Estimation. In *IEEE CVPR*, pages 2304-2313, 2018.

DOI: 10.48550/arXiv.1712.07195

[8] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao. Using ranking-CNN for age estimation. In *IEEE ICCV*, pages 5183-5192, 2017.

DOI: 10.1109/CVPR.2017.86

[9] Ruiz. N, Chong. E, Rehg. J.M. Fine-grained head pose estimation without key-points. In *CVPR workshops*, pp. 2074-2083, 2018.

DOI: 10.48550/arXiv.1710.00925

[10] W. Pei, H. DibeklioÄlu, T. BaltruÅ¡aitis and D. Tax. Attended End-to-End Architecture for Age Estimation From Facial Expression Videos. In *IEEE Transactions on Image Processing*, volume 29, pages 1972-1984, 2019.

DOI: 10.1109/TIP.2019.2948288

[11] Z. Ji, C. Lang, K. Li and J. Xing. Deep Age

Estimation Model Stabilization from Images to Videos. In *International Conference on Pattern Recognition*, 2018.

DOI: 10.1109/ICPR.2018.8545283

[12] Simonyan. K, Zisserman. A. Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556, 2014.

DOI: 10.48550/arXiv.1409.1556

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

DOI: 10.48550/arXiv.1512.03385

[14] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua. Ordinal regression with multiple output cnn for age estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4920-4928, 2016.

DOI: 10.1109/CVPR.2016.532

[15] K. Chang, C. Chen, and Y. Hung. A ranking approach for human age estimation based on face images. In *ICPR*, 2010.

DOI: 10.1109/ICPR.2010.829

[16] K. Chang, C. Chen, and Y. Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. In *CVPR*, pages 585-592, 2011.

DOI: 10.1109/CVPR.2011.5995437