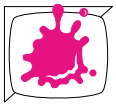
Proceedings of the

28th Central European Seminar on Computer Graphics

May 1 - 4, 2024
Smolenice, Slovakia

CESCG 2024



Institute of Visual Computing & Human-Centered Technology
TU Wien

Faculty of Mathematics, Physics and Informatics
Comenius University Bratislava

Institute of Mathematics
Slovak Academy of Sciences

Department of Computer Graphics and Interaction
Czech Technical University in Prague

Partners

Edited by Martin Ilčík, Jiří Bittner, Zuzana Berger Haladová and Michael Wimmer © 2024
ISBN: 978-3-9504701-5-4

Impressum

TU Wien
Institute of Visual Computing & Human-Centered Technology
Favoritenstraße 9-11 / E193-02
1040 Vienna
Austria

ISBN 978-3-9504701-5-4

Welcome to CESCG 2024!

This book contains the proceedings of the 28$^{\text{th}}$ Central European Seminar on Computer Graphics, short CESCG, which continues a history of very successful seminars.

The long history of CESCG has started in 1997 in a medium-sized lecture room in Bratislava, bringing together students from Bratislava, Brno, Budapest, Graz, Prague, and Vienna. The idea found wide appraisal and the seminar moved to the beautiful castle of Budmerice, where it was held for eight consecutive years, constantly growing in size and attraction. It was just in the 10$^{\text{th}}$ anniversary year 2006 that CESCG had to take a detour to move to Častá-Papiernička Centre, while it was back in Budmerice castle since 2007. Budmerice castle ultimately closed down for public in 2011. After spending that year in Viničné, in 2012 we moved to the beautiful castle in Smolenice. During the COVID pandemics lock-downs in 2020 and 2021, CESCG switched to a virtual mode at Discord and YouTube.

Who are the CESCG heroes who made this year's seminar happen? In no particular order – because many people were involved equally – we would like to thank the organizers from Vienna: Annalena Ulschmid, Diana Marin, **Michael Wimmer** and Max Höfferer. Special thanks goes to **Martin Ilčík** for extensive event management, keeping the seminar alive for over 15 years. **Jiří Bittner** from ČVÚT in Prague and Zuzana Berger-Haladová from Comenius University took care of the scientific process. We are very thankful to further CESCG organizers from Bratislava, Martin Madaras, Lukáš Hudec and **Andrej Ferko**, always an inspiration to CESCG. Simon Sadeger will produce professional promotion videos and Viktoria Pogrebacz will take care of recording the talks in full length.

The main idea of CESCG is to bring students of computer graphics together across boundaries of universities and countries. We focus on sustainable academic and research development in the field of Computer Graphics in Europe. Our mission is to support undergraduate talents in their future careers. We have 17 participating institutions and a tight time schedule of 18 student papers and 2 posters. For the first time at CESCG, a PhD Colloquium will host 8 interesting work-in-progress graduate talks.

We welcome groups from Bratislava (Comenius and STU), Slovakia; Brno (Masaryk and VÚT) and Prague (Charles and ČVÚT), Czech Republic; Budapest (BUTE), Hungary; Cambridge (University), United Kingdom; Graz (TU) and Vienna (TU and VRVis), Austria; Stuttgart (University) and Saarbrücken (Saarland University), Germany; Paris (ESIEA), France; Sarajevo (University and SSST), Bosnia and Herzegovina; Zürich (University), Switzerland. Martin Ilčík from TU Wien led the virtual workshop on "Scientific Storytelling" right after the students received the first feedback to their papers. The keynote talk "Different Perspectives of Data Visualization" will be given by Barbora Kozlíková from Masaryk University University in Brno.

We have assembled an International Program Committee (IPC) of 15 members, allowing us to have each paper reviewed by two IPC members during the informal

reviewing process. Students also cross-reviewed their papers. We would like to thank the members of the IPC for their contribution to the reviewing process:

Wanda Benešová	Martin Ilčík	Renata Raidou
Zuzana Berger Haladová	Tomáš Iser	Selma Rizvić
Jiří Bittner	Barbora Kozlíková	Dieter Schmalstieg
Daniel Cornel	Rafal Mantiuk	László Szécsi
Adam Herout	Radosław Mantiuk	Michael Wimmer

The reviewing process was further supported by: Vlastimil Havran, Jiří Hladůvka, Áron Samuel Kovács, Martin Káčerik, Miroslav Macík, Ivo Malý, Maath Musleh, Daniel Pahr, David Sedláček, Annalena Ulschmid.

For the first time in CESCG history, this edition will feature a dedicated PhD Colloquium where 8 graduate students will discuss their research ideas:

- Tensor Decomposition for Fast Weather Prediction Data Rendering
  Julian Croci. University of Zürich
- Working In Mixed Realities: 3D User Interfaces and Interaction Patterns to Integrate Practices in Different Perceptual Spaces
  Marina Lima Medeiros. VRVis Research Center, Vienna
- Human Emotion Recognition in VR Systems for Remote Telepresence and Collaboration for Training and Education Purposes
  Mamadi Dioubaté. ESIEA Graduate School of Engineering, Paris
- Procedural Modeling of Traversable Hierarchically Organized Layouts with Interoperability between Different Levels of the Hierarchy
  Emir Cogo. University of Sarajevo
- An Algorithm for Stochastic Progressive Refinement of Large Meshes
  Martin Čavarga. Comenius University, Bratislava
- Graphical Software Tool for Creating Standardized Cause-Effect Graph Specifications
  Ehlimana Cogo. University of Sarajevo
- Learning Parametric Primitive Segmentation on 3D Point Clouds
  Lizeth Fuentes. University of Zürich
- Domain Expert Centered Interface Design for AI-Infused System Development
  Martin Dubovský. Slovak Technical University, Bratislava

With the 20$^{\text{th}}$ anniversary of the seminar in 2016, Martin initiated the CESCG EXPO project. This year we restarted the EXPO with companies and research institutions specialized on visual computing presenting their innovative products in an interactive exhibition. The exhibitors are:

Akular, Bratislava	Procedural Design, Vienna
Canon, Bratislava	Shadow Map, Vienna
Cognex, Bratislava	VR Group, Brno
Escape Motions, Piešťany	WildRealm, Bratislava
Nanographics, Vienna	

For 2020 Martin initiated the ACADEMY project to offer tutorials and lectures by international experts to stimulate knowledge exchange in a way similar to a summer school. The COVID pandemics interrupted these plans, so the zero-year offered just a reduced set of workshops virtually. The first real ACADEMY took place 2022, offering a day with 5 tutorials right after the seminar. The next year the ACADEMY finally got its intended shape with 12 tutorials and lectures given in 4 parallel tracks over the course of 2 full days. This year we prepared a similar schedule with 8 tutorials in 2 parallel tracks spanning over 2 days:

GPGPU

- CUDA and Applications to Task-based Programming
  Michael Kenzel. Saarland University

XR

- Recent Trends in Augmented Reality
  Dieter Schmalstieg. University of Stuttgart

Reconstruction

- A Hands-on Introduction to Photogrammetry
  Wallace Wainhouse. Epic Games, Bratislava

Music

- Making Music with Code (Infinite Music for Games and More)
  Peter Mindek, Nanographics, Vienna

- Responsive Music Programming with D♭
  Martin Ilčík, Procedural Design, Vienna

AI

- Into the World of Generative AI
  Lukáš Hudes and Maroš Kollár, STU Bratislava

Geometry

- Voronoi Diagrams and Their Applications
  Martin Maňák, University of West Bohemia, Plzeň

Physics

- Cloth Simulation for Video Games
  Annalena Ulschmid, TU Wien

The organization of such a large event requires additional funding. We are very thankful to the partners of CESCG 2024 for supporting us financially and by donating prizes for awarding the best student results:

– **KAUST**, King Abdullah University of Science and Technology,
– **VRVis**, Research Center for Virtual Reality and Visualization,
– **Canon**, digital imaging solutions,
– **SISp**, Slovak Society for Computer Science,
– **Escape Motions**, developer of innovative visual tools,
– **IEEE Women in Engineering**, promoting women engineers and scientists,
– **Procedural Design**, adaptive content creation,

Please note that the electronic version of these proceedings is also available at https://cescg.org/library/.

April 2024,

Martin Ilčík
Jiří Bittner
Zuzana Berger Haladová
Michael Wimmer

Table of Contents

Computer Vision

Rendering and beyond

Posters

Partners of CESCG 2024

Keynote Talk

Different Perspectives of Data Visualization

Barbora Kozlíková

Masaryk University
Brno, Czech Republic

Abstract

In this talk, I'll focus on presenting different aspects that play a role in data visualization. On several use cases, mostly from the biomedical visualization, I will demonstrate the main challenges that the visualization researchers are facing. We will also touch related aspects, such as visualizing uncertainty and building a trust in visualization. I'll also share experiences with interdisciplinary research and collaboration with experts from other fields.

Bibliographical Details

Barbora currently holds the position of an Associate Professor at the Faculty of Informatics, Masaryk University, Brno, Czech Republic. She is heading the Visitlab research group focusing on diverse topics in visualization. Her main research interests are visualization and visual analysis with diverse application areas, with the largest focus on biochemistry. In the past 15 years, she has been intensely collaborating with protein engineering experts and together they developed the CAVER and CAVER Analyst tools for exploration of protein structures and their tunnels, where the temporal aspect of proteins plays a crucial role.

Computer Vision in Medicine (addendum 2023)

Segmentation of Whole-Slide Images with Context-Aware Vision Transformers

Michal Franczel*
Supervised by: Lukáš Hudec†

Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Bratislava / Slovak Republic

Abstract

Histological examination is a crucial component of breast cancer diagnostics. Analysis of whole-slide images (WSI) is a time-consuming process due to their hierarchical nature and size, resulting both in slower diagnostics and a lack of annotations. Recent advances in vision transformers have demonstrated potential within the field of computer vision. However, their properties with hierarchical gigapixel images, where contextual information is crucial, remains underexplored. In this paper, we propose a solution employing semi-supervised learning based on a self-supervised pretraining and supervised fine-tuning paradigm, utilizing these advancements. Our approach modifies vision transformer encoders within the segmentation network to incorporate contextual information from lower magnification levels through late feature fusion. The multi-scale model variant outperforms its single-scale counterpart, improving the dice score by 6.2%. Furthermore, we examine the properties of features learned by masked image modeling (MIM) and establish that vision transformers trained with MIM can effectively learn morphological phenotypes from unlabeled histopathological images, thereby validating its use as a pretraining technique in this domain.

Keywords: Whole-Slide Images, Breast Cancer, Deep Learning, Segmentation, Semi-Supervised Learning, Vision Transformers, Medical Imaging

1 Introduction

Breast cancer is one of the leading preventable causes of death, accounting for more than 13 percent of all new cancer cases and 28.7 percent of all cancer discoveries in women in the European Union as of 2020. While the number of new cases has increased over time, the number of deaths has decreased [9]. This can be explained not just by increasing the quality of treatment but also by increasing the rate of early disease diagnosis [7]. Histological analysis is a vital, yet time-consuming and difficult, component of breast cancer diagnosis. As part of the pathological examination of the breast, a biopsy is performed. The extracted tissue is sliced, and the slice is then stained most commonly with hematoxylin and eosin (H&E). Subsequently, it is placed on a glass slide, which is scanned with a motorized microscope. Slides are scanned at multiple magnification levels, resulting in z-stacks. This enables pathologists to switch between these magnification levels, simulating classical microscopy. Pathologists analyze a variety of structures, not only at the level of cellular morphology but also at the level of larger breast structures, utilizing both contextual information from lower levels and detailed information from higher levels of magnification.

Deep learning has been an essential part of computer vision, and convolutional models have been successfully applied to the field of computational pathology. However, in recent years, transformer-based vision models have gained prominence, obtaining state-of-the-art performance across numerous general vision tasks. The properties of these transformers have, however, not yet been thoroughly examined within the field of histopathology, especially when dealing with segmentation of hierarchical images. Due to their size, Whole-Slide Images (WSI) must be split into smaller patches. These patches, at the highest magnification level, may not contain the necessary information for segmentation models to make accurate predictions, as they lack coarser-grained tissue features and their spatial organization. Additionally, the complex hierarchical nature of WSI results in a lack of annotated datasets in terms of both quantity and quality, especially when dealing with pixel-level annotations.

In this work, we first assess the utility of vision transformers on the BCSS dataset and then propose modifications to the segmentation network so that it uses multiple magnifications and passes contextual information in a top-down manner. We split methods of feature fusion into three categories: early, intermediate, and late fusion. Prior work on convolutional encoders used variations of early and late feature fusion with linear and LSTM layers. We examine early and late feature fusion with the use of

*xfranczel@stuba.sk
†xhudecl@stuba.sk

the Swin Transformer [8] encoder and a modified Upernet [15] architecture, using both linear layers and cross-attention mechanisms. Additionally, we expand these two types of fusion through intermediate fusion, merging features between individual blocks of the encoder. We find that vision transformers, even though they lack the inherent biases of convolutional networks, can achieve similar accuracy within the domain of histopathology. With the introduction of late feature fusion, we surpass the accuracy of single-scale single architectures, increasing the dice score by 6.2%.

Additionally, to address the issue of the lack of annotations, we explore the use of self-supervised pretraining methods based on masked image modeling (MIM) utilizing the BRACS [3] dataset. Through qualitative analysis of learned features, we find that transformer-based models pretrained with MIM on the BRACS dataset can learn useful representations of various types of tissues, confirming that it can be used as a pretraining step within the domain of histopathology.

2 Related Work

The segmentation task can be interpreted either as a pixel-wise segmentation or a patch-wise segmentation, where an image is split into patches, which are then classified and combined to create a coarse segmentation mask. Numerous approaches to pixel-wise segmentation have been proposed, with the most prevalent being the convolutional U-Net [12], featuring an encoder-decoder architecture with a bottleneck and skip connections. Variants of this architecture have also emerged, including U-Net++ [18], which employs a densely connected decoder subnetworks, and R2U-Net [1], which incorporates recurrent modules within both its encoder and decoder stages.

These architectures, frequently employed in cell and organ segmentation tasks, do not take advantage of the hierarchical structure of whole-slide images. To address this limitation, a number of context-aware methods have been introduced for the classification and segmentation of histopathological images. Sirinukunwattana et al. [13] studied the impact of providing contextual information to the prediction algorithm. They approached the problem of image segmentation as a patch-level classification and compared three types of architectures: single-scale architecture, which operates at a single image resolution; early fusion, which fuses information from multiple resolutions before passing it through a neural network; and late fusion, which uses separate networks for different magnifications and combines the output to make a prediction. Out of the three groups described, a single-scale design performed significantly worse than architectures that used contextual information. Feng et al. [6] proposed an end-to-end framework that generates predictions at multiple magnification levels and combines them using a voting process, adopting the late fusion approach. This approach was also used by the multi-scale classification model proposed by Wetteland et al. [14] to classify small patches, combining them into segmentation mask of an entire WSI. One major advantage of the late fusion approach is its ability to utilize contextual information from multiple resolutions, enhancing prediction accuracy. However, the main disadvantage is the increased computational complexity compared to single-scale architectures.

Chen et al. [5] proposed the Hierarchical Image Pyramid Transformer (HIPT), a three-stage architecture that performs bottom-up aggregation for slide-level representation, akin to hierarchical attention networks in long document modeling. The model allows for self-supervised pretraining methods to pretrain each aggregation layer separately, which can then be fine-tuned with slide-level labels for cancer subtyping and survival prediction tasks in the TCGA. Since the attention is computed only within local windows, learning long-range dependencies is tractable. Even though this method of bottom-up aggregation is not useful for image segmentation, it may be useful as a pretraining step.

3 Data

Breast Cancer Semantic Segmentation (BCSS) [2] dataset contains regions of interest derived from 151 WSIs stained with H&E, collected from histologically confirmed cases of breast cancer. Pathologists graded regions from 21 difficult slides that were annotated by trained non-pathologist research participants. Masks, pixel-level annotations with 21 classes, were the resulting annotations. Within the scope of our work, we opted for the use of the modified version of this dataset with labels reduced according to TIGER Challenge.

BReAst Carcinoma Subtyping (BRACS) [3] dataset is a breast carcinoma subtyping dataset containing 547 H&E-stained whole-slide images and 4539 extracted regions of interest from these WSIs. Both WSIs and ROIs were annotated with lesion categories by the consensus of three pathologists. Benign, malignant, and atypical lesions are further subtyped into seven distinct categories. Even though this dataset is one of the largest in its category, it does not include annotations at the pixel level. However, it can be used for unsupervised training.

4 Method

One of our objectives is to make use of current developments in transformer-based models. We decided to exploit current breakthroughs in semi-supervised learning, focusing on the paradigm of self-supervised pretraining and fully-supervised finetuning, as training these transformer models requires vast quantities of data. Thus, as shown on Figure 1, training is divided into two stages, with each of these stages being performed at the slide level using a

patch generator. This generator generates a tissue mask and patch stacks for the entire WSI. The resulting data is subsequently used for training. Within the first stage of our experiments, we compare various architectures that either utilize input from a single magnification level or use custom architectures that make use of three separate magnifications. In the second stage of our experiments, we focus on self-supervised pretraining with the BRACS dataset. Since the dataset contains 547 WSIs, training is difficult given the available computational resources. Therefore, we have chosen to sample WSIs, and from these images, we have chosen to sample 150,000 patches. In this phase, we will evaluate method of masked image modeling.

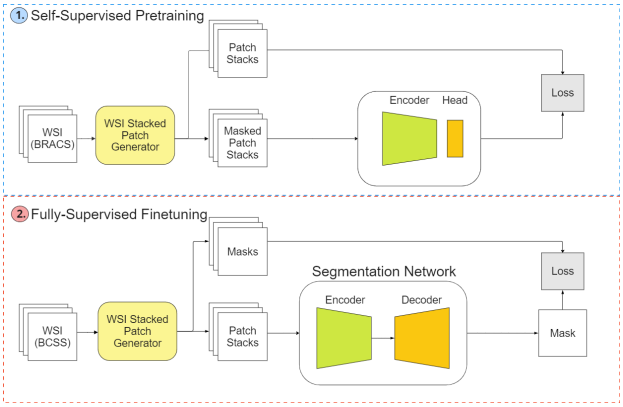


Figure 1: Overall two-stage architecture of proposed framework with self-supervised pretraining using MIM and fully-supervised context-aware finetuning

4.1 Data Preparation

Upon reviewing slides included in the dataset, we have discovered that a large part of all WSIs consist of background material without any histopathological relevance. That is why, firstly, a tissue mask is constructed employing simple thresholding and morphological operations. As some of the slides were labeled and contained artifacts, we created a mask for artifacts we observed by applying thresholding to the converted LAB image and deleting them from the tissue mask.

Patches are generated based on the pixels per micron (PPM) parameter of the slide, so that the generator can be used on varying datasets. Patches at the greatest magnification level containing tissue proportions below the threshold are discarded. When dealing with the BCSS dataset, patches with masks that include background levels above threshold in their annotation at the greatest magnification level are eliminated. When multiscale patch stacks are required, the location of the patch at the highest magnification level is determined first, and then the locations of patches at lower magnification levels are calculated, with higher magnification being at the central position. Stain normalization is the final phase in the preparation process, and we have decided to use the Macenko method of normalization [10]. This method is frequently employed as a preprocessing step, as it estimates hematoxilin and eosin concentrations from color space distributions and normalizes input images based on these concentrations, given some target image. Preprocessing flow is illustrated on Figure 2.

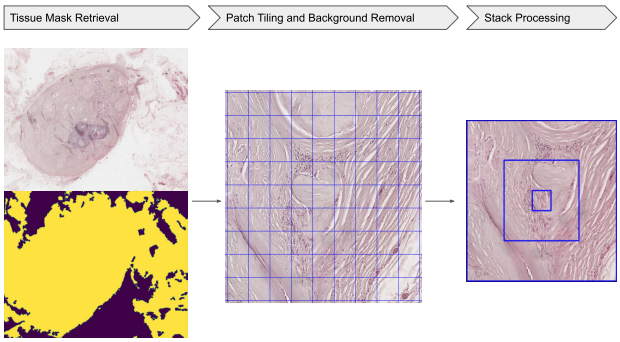


Figure 2: Three stages of data preparation: tissue mask retrieval with tresholding, patch tiling and background removal, and the addition of contextual patches from lower magnifications

4.2 Training Configuration

As per established and recommended training parameters, the final training configuration for fully supervised training uses the ReduceLROnPlaeau scheduler for convolutional networks and the cosine scheduler for transformer-based networks. As for the optimizer, AdamW is used, with betas set to 0.90 and 0.999, epsilon 1e-8, and a base learning rate of 5e-4. Since we have observed that this dataset is unbalanced and comprises disproportionately greater stroma and tumor types, we used Dice Cross-Entropy Loss, which includes squared versions of targets and predictions in the denominator,

$$\text{Loss} = \left(1 - \frac{2 * \sum_{c=1}^{C} p_c \hat{p}_c}{\sum_{c=1}^{C} p_c^2 + \sum_{c=1}^{C} \hat{p}_c^2}\right) - 0.5 * \sum_{c=1}^{C} p_c \log \hat{p}_c$$

where, C is the number of classes or categories, p_c is the true probability of class c, and $\hat{p}_c$ is the predicted probability of class c. Since the background class label in this dataset reflects unannotated regions and offers no semantic relevance in terms of training, it is not included in the calculation of loss.

To improve model resilience and reduce susceptibility to color pertubations, we employed augmentations. Vertical and horizontal flips, random rotations, and Contrast Limited Adaptive Histogram Equalization (CLAHE), random brightness, and contrast were employed. Additionally, we attempted to address the issue of class imbalance by oversampling and undersampling patches based on the classes present within its segmentation mask.

As for patch extraction, we used patch size 224 and overlap 112, as images of this scale contained sufficiently complex structures both within images and masks. Using our patch extractor, we used scale 1 as an input for single scale experiments and 1, 4, and 8 for multi-scale experiments. Within the scope of this work, the term *scale* refers to the downsampling factor relative to the highest level of magnification.

In the case of fully supervised training, the BCSS dataset was divided into 133 training samples and 18 validation samples. To prevent training on validation data that can be caused by overlap, patches were generated after this partitioning. During the training and validation process, metrics were computed patch-wise.

4.3 Single-Scale Architectures

First, we experimented with architectures based on U-Net [12] in order to establish a baseline and test configurations for data generation, as well as the properties of transformer-based networks and their usefulness within the domain of histopathology. For these U-Net-based designs, both encoder and decoder blocks based on transformers and convolutional blocks were utilized.

After these experiments, we focused on more complex and recent architectures, which generally perform better in multi-class semantic segmentation, with one of them being SegFormer [16] and the other being Upernet. Experiments on the SegFormer architecture were conducted with only small deviations from the original publication. The Upernet architecture was changed from multi-task to single-task with a Swin Transformer backbone. Backbone, which we used as an encoder for Upernet, remained the same. All the previously mentioned single-scale architectures maintain their original parameters, with alterations limited to their training parameters and minor implementation-related deviations. The transformer variants of these architectures employ the base size of the Swin encoder.

4.4 Multi-Scale Architectures

After the first iteration of experiments with a single scale, we focus on experiments combining multiple scales. Vision transformers have some different properties than convolutional neural networks, with the most important example being that they lack their intrinsic biases. Additionally, features of these two methods vary significantly [11], with global features being present at much earlier network stages. Simultaneously, various new layers and architectural elements were introduced in vision transformers and transformers in general, some of which do not have their counterparts within convolutional architectures. We try to improve the prediction accuracy of single-scale models evaluated in previous iterations using various methods, which we have divided into three categories:

1. *Early fusion*, where a single encoder is used and images from three different scales are combined before the first encoder stages
2. *Intermediate fusion*, where either three encoder branches are used and features are combined between stages from top to bottom, or a single encoder is used sequentially
3. *Late fusion*, where three images are passed through branches separately and fusion is performed on features passed before passing them to the encoder

We chose the Upernet-based architecture for these experiments involving multiple magnifications since it performed significantly better than other segmentation networks.

4.4.1 Early Fusion

The first method involves using a single encoder and merging its input before its first stage, concatenating channel dimensions. The number of blocks used was the same as for the single-scale encoder, but we increased the number of heads in the first two stages to 9 to match the complexity of the network to the increased complexity of the input.

Our second method for early fusion, Upernet T3 is comprised of three encoder blocks, which process input triplets sequentially in a top-down manner, passing contextual information to higher magnifications. The patch at the lowest magnification level is processed by the first encoder. The input for the first stage of the next encoder is the second magnification level, and the first stage of this encoder produces a feature map that is prepended to the feature maps produced by the first encoder. This information is then fed into the feature pyramid network (FPN), which produces a single, combined feature map. This map is then fed into the remaining stages of the second encoder. After that, the same operation is carried out with the second and third encoders, respectively. The feature maps produced by the third encoder are subjected to processing with pyramid pooling module (PPM) and FPN Fuse blocks, which ultimately produce a segmentation mask.

4.4.2 Intermediate Fusion

The first model architecture, which we implemented with respect to intermediate fusion, marked as Upernet T6, was intermediate fusion with the use of cross-attention, visualized on Figure 3. We used three separate Swin encoders. Images taken at different scales are passed through encoder stages, with each stage being followed by a cross-attention stage. The cross-attention stage is composed of two cross-attention blocks followed by layer normalization, where the first cross-attention blocks takes an input low smallest magnification as context and intermediate magnification as an input and the second one takes highest magnification as an input and result of previous cross-attention as context. The intuition behind this idea was

that, using purely attention-based mechanisms, we could pass information between the encoder stages of these three branches in a top-down manner.

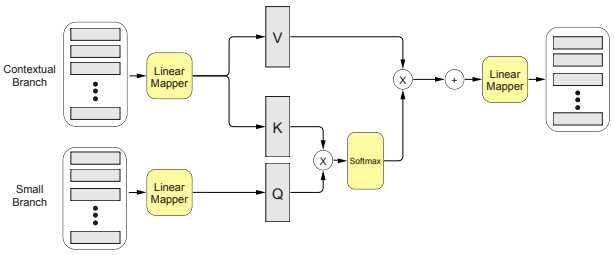


Figure 3: Cross-attention module

The second architecture utilizing intermediate fusion, with a designated label of Upernet T7, functioned in a similar manner, but instead of cross-attention, we fused class (CLS) token with patch tokens of higher magnification using convolutional layers. First, since Swin does not contain an explicit learnable CLS token, we compute it using an embedding layer, which takes input patch tokens as input, passes them through layer normalization followed by adaptive average pooling and a linear layer, producing a single class token representing all patches combined. This token is concatenated with patch tokens from higher magnification, and dimensions are reduced back to their original size using point-wise convolution. This way, we attempted to fuse features between three encoder branches by passing an aggrieved CLS token to lower magnifications.

4.4.3 Late Fusion

As for late fusion, we experimented with two methods as well. Our first method, named Upernet T2, was composed of three swin encoders, each used for different magnification level, all of them of same size. Since there are three encoder branches, the outputs of these stages need to be merged. For this, three outputs are first rearranged so that the height and width dimensions are reshaped into a single dimension, representing all tokens. Then we concatenate these tokens and pass them through an MLP block with the GELU activation function, which reduces their number to their original number. Finally, they are rearranged back to their original shape, and the resulting feature map is passed through the same PPM and FPN Fuse blocks. This architecture is shown on Figure 4.

The second method, named Upernet T5, utilizes feature merging instead of linear layers with a single cross-attention block, merging feature maps that are outputs of the last encoder stages. First, images from three selected scales are passed through all three encoders. Subsequently, the feature maps from the last encoder stages are passed through a self-attention block. Cross-attention is done twice, with the objective of passing contextual information from the lowest to the highest magnification level. The feature map produced by the second cross-attention module is then passed through the FPN Fuse block, result of which is then passed through the PPM block together with other features from the lowest magnification level. We hypothesized that the application of the self-attention mechanism in this manner may prove useful for passing high-level features at lower magnifications.

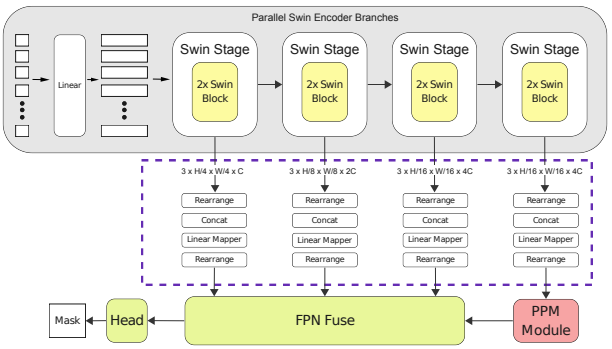


Figure 4: Architecture of Upernet T2 with three encoder branches and late future fusion

4.5 Masked Image Modeling

The effects of masked image modeling within the domain of histopathology have not yet been thoroughly studied. Thus, we focus on self-supervised training using masked-image modeling with iBot [17], which obtained state-of-the-art performance on various vision tasks outside of the medical domain.

iBot, similarly to DINO [4], employs two views created by augmenting the input image. Since it was not originally trained on medical images, we changed several parameters of these augmentations in order to accommodate the medical domain. iBot first applies local and global transforms, which produce local and global crops from the input image. After a visual evaluation of augmented crops, we changed the range of global crops to be between 0.7 and 1, from the original 0.14 and 1, and the range of local crops to be between 0.2 and 0.4, increasing the original values of 0.05 and 0.4, since such small patches did not contain sufficient information. Additionally, we removed color jitter and random grayscale from both local and global transforms and reduced the probability of solarization and Gaussian blur to 0.1. Experiments were done on ViT backbones on three different scales. For highest magnification, we used patch size 16, since we found it generally delivered better self-supervised results than with patch size 8. On intermediate and lowest magnification, we used patch size of 4, since we found, that smaller patch sizes on lower magnification levels provided necessary increase in resolution of attention maps.

5 Results

Within our single-scale experiments, we have found convolutional U-Net to be worse performing in comparison with transformer-based U-Net with respect to dice score. However, it obtained better IoU and per-class accuracy. Thus, we evaluated, that Swin-based U-Net performs on par with its convolutional counterpart, but with better inference and training speeds. Even though Segformer is a model with high throughput and efficiency, it performed significantly worse than Upernet, which outperformed all other evaluated models, as is shown within Table 1.

Table 1: Quantitative single-scale model evaluation

Models	mIoU	mDice	Per-Class Acc
Conv U-Net	29.44	36.57	36.27
Swin U-Net	28.58	36.91	35.72
Segformer	36.80	44.15	43.73
Upernet	46.03	56.05	55.1

Similarily to single-scale models, we have obtained several interesting results with multi-scale models. First, attention-based feature merging performs better when implemented as intermediate feature fusion rather than late fusion. Second, it was overcome by a method combining linear layers with late fusion. However, despite the fact that Upernet T6 with three encoder stages performed worse than Upernet T2 of the same size, its counterpart with a single encoder outperformed Upernet T6 of the same size. Thirdly, both methods of early fusion performed worse than their late and intermediate counterparts. Lastly and most importantly, all three best performing models outperformed their best performing single-scale counterpart. This result is attributable to the use of multiple contextual magnifications, which allowed the model to be trained on a specific dataset. Table 2 displays the respective results of the five architectures with the highest performance.

Table 2: Quantitative multi-scale model evaluation

Models	mIoU	mDice	Per-Class Acc
Upernet T3	40.8	47.92	48.24
Upernet T2*	46.12	54.67	53.96
Upernet T6*	48.15	56.52	55.68
Upernet T6	49.42	58.34	57.96
Upernet T2	52.13	62.25	61.51

* Single encoder used sequentially instead of three parallel encoders

5.1 Masked Image Modeling

Our qualitative evaluation involved the analysis of attention maps within the last transformer block. We used three different scales to measure how well these attentions worked on all three transformers that were trained with iBot. These were the same scales that were previously used for multi-scale experiments: 1, 4, and 8.

We started our experimentation at the highest magnification. Since the pretrained model was focusing only on white regions and ignoring regions with tissue compartments, we found the results to be underwhelming after training with patch size 8 and examining the attention maps of the model. However, when we increased the patch size to 16, we noticed that the attention heads began to concentrate more on the different kinds of tissue and small structures within the images, particularly cells. Following a qualitative analysis of attention, we discovered that heads 4, 7, and 12 acquired the ability to tell surrounding tissue from cells, which is particularly evident in images of tissue devoid of bubbles or other white regions. Within Figure 5 are visualized attention maps from heads 4, 3 and 2 of the last block of the encoder network, attending cells, stroma and fatty tissue, respectively.

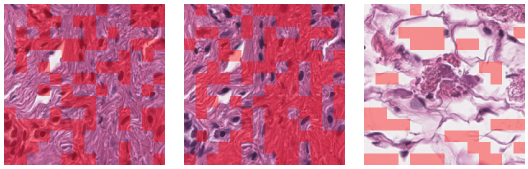


Figure 5: Visualized attention maps of cells (left), stroma (center) and fatty tissue (right) from the last encoder block with the highest magnification used

Secondly, we trained the same model with a scale of 4, with the model patch size set to 8. Even though the model did not focus on cellular structures as much as with higher magnification, upon analyzing various tissue types, we observed that it rather focused on differentiating between tissue compartments. Within fatty tissue, we can see that the attention map of heads 0, 1, 3, and 9 focused on membranes, and heads 2, 4, 5, 7, and 8 focused more on the fat itself, which resides within these membranes. Structures recognized by head 6 were not as apparent. However, it seems like model focused more on darker structures, including cells and darker tissue material where membranes meet. Within the darker patches that do not contain fat, we observed that attention within heads 0, 1, 3, 6 and 9 focused on stroma tissue connecting various other compartments, and attention within head 11 focused on darker regions of an image. On Figure 6, we can observe attention maps attending connective tissue and darker regions within fatty patches, as well as darker regions and stroma within patches containing tissue.

Finally, we trained a Swin transformer, which could be utilized for additional fine-tuning experiments, following the same approach as with ViT. We processed randomly sampled images and extracted features from the final layer of the encoder network. After applying T-SNE dimensionality reduction, we clustered the points using K-Means

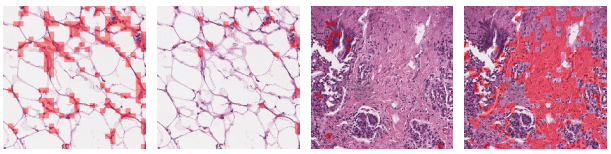


Figure 6: Visualized attention maps of connective tissue, dark regions and stroma from the last encoder block with patches from lower level of magnification

and visualized two components in Figure 7, with images representing the cluster centers. Figure 8 presents points accompanied by their respective images. The formation of clusters containing visually similar images is observed, with the primary basis for similarity being their visual characteristics.

6 Conclusions

First, we focused on the development of a full data processing pipeline that prepares whole-slide images for either training or analysis. Following a preliminary analysis of available datasets, we focused on evaluating single-scale models on the BCSS dataset and comparing convolutional networks with transformer-based networks, which have gained popularity in recent years. For the subsequent experiments involving multiple contextual magnifications, we chose the top-performing architecture based on an evaluation of these models. Experimenting with various variations of Upernet, we discovered that our multi-scale modification, Upernet T2, which is based on the late fusion of features between three backbones, outperforms its single-scale counterpart. We conclude, based on these results, that contextual usage improves the performance of the model.

Finally, we examined the effects of self-supervised learning with masked image modeling and its applicability in the medical field. After analyzing vision transformers pretrained with iBot, we conclude that masked image modeling is applicable to this domain and that models trained with masked image modeling may prove useful in future experiments.

Further research is required in the area of self-supervised pretraining. We found that this pretraining method is not appropriate for lower magnifications, but we hypothesize that this pretraining process could be generalized to multi-scale image processing by changing the training process so that all three encoders would learn representations together rather than separately. This requires changing the iBot head to accommodate the use of three backbones and changing the loss function so that all three encoders can be trained. Furthermore, we used only small models and a limited number of pretraining samples, since our evaluation of the self-supervised algorithm concentrated primarily on the analysis of feature maps. In order to evaluate larger Swin models, which should perform better with self-supervised pretraining, more training samples are required.

Weakly supervised learning could be incorporated into the fully supervised stage of our training pipeline. Since the BCSS dataset contains whole-slide images, but only ROI-level annotation, it is possible to use surrounding regions progressively as a method of weakly-supervised learning. As the segmentation model is trained from the labeled data during training, it can also produce segmentation masks for nearby regions, resulting in their weak labels, which can subsequently be added to the training set. Since the areas around labeled patches have some commonalities, we hypothesize that the model could use this shared information to generate reliable weak predictions. However, as training goes on, the factor of expansion must decrease because patches farther away do not benefit from proximity to the labeled patches.

References

[1] Md. Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M. Taha, and Vijayan K. Asari. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *CoRR*, abs/1802.06955, 2018.

[2] Mohamed Amgad, Habiba Elfandy, Hagar Hussein, Lamees A Atteya, Mai A T Elsebaie, Lamia S Abo Elnasr, Rokia A Sakr, Hazem S E Salem, Ahmed F Ismail, Anas M Saad, Joumana Ahmed, Maha A T Elsebaie, Mustafijur Rahman, Inas A Ruhban, Nada M Elgazar, Yahya Alagha, Mohamed H Osman, Ahmed M Alhusseiny, Mariam M Khalaf, Abo-Alela F Younes, Ali Abdulkarim, Duaa M Younes, Ahmed M Gadallah, Ahmad M Elkashash, Salma Y Fala, Basma M Zaki, Jonathan Beezley, Deepak R Chittajallu, David Manthey, David A Gutman, and Lee A D Cooper. Structured crowdsourcing enables convolutional segmentation of histology images. *Bioinformatics*, 35(18):3461–3467, 02 2019.

[3] Nadia Brancati, Anna Maria Anniciello, Pushpak Pati, Daniel Riccio, Giosuè Scognamiglio, Guillaume Jaume, Giuseppe De Pietro, Maurizio Di Bonito, Antonio Foncubierta, Gerardo Botti, Maria Gabrani, Florinda Feroce, and Maria Frucci. BRACS: A Dataset for BReAst Carcinoma Subtyping in H
amp;E Histology Images. *Database*, 2022, 10 2022. baac093.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021.

[5] Jia-Mei Chen, Yan Li, Jun Xu, Lei Gong, Lin-Wei Wang, Wen-Lou Liu, and Juan Liu. Computer-

aided prognosis on breast cancer with hematoxylin and eosin histopathology images: A review. *Tumour Biol.*, 39(3):1010428317694550, March 2017.

[6] Yanbo Feng, Adel Hafiane, and Hélène Laurent. A deep learning based multiscale approach to segment cancer area in liver whole slide image, 2020.

[7] S. W. Fletcher, W. Black, R. Harris, B. K. Rimer, and S. Shapiro. Report of the International Workshop on Screening for Breast Cancer. *J Natl Cancer Inst*, 85(20):1644–1656, Oct 1993.

[8] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021.

[9] Bettio M, Negrao De Carvalho R, Dimitrova N, Dyba TA, Giusti F, Martos Jimenez MDC, Neamtiu L, Nicholson N, Randi G, Rooney R, Voti L, Crocetti E, and Voithenberg L. Dataset collection: European cancer information system. 2023.

[10] Marc Macenko, Marc Niethammer, J. S. Marron, David Borland, John T. Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E. Thomas. A method for normalizing histology slides for quantitative analysis. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1107–1110, 2009.

[11] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *CoRR*, abs/2108.08810, 2021.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[13] Korsuk Sirinukunwattana, Nasullah Khalid Alham, Clare Verrill, and Jens Rittscher. Improving whole slide segmentation through visual context - a systematic study, 2018.

[14] Rune Wetteland, Kjersti Engan, Trygve Eftestøl, Vebjørn Kvikstad, and Emiel A. M. Janssen. A multiscale approach for whole-slide image segmentation of five tissue classes in urothelial carcinoma slides. *Technology in Cancer Research & Treatment*, 19:1533033820946787, 2020.

[15] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding, 2018.

[16] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *CoRR*, abs/2105.15203, 2021.

[17] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan L. Yuille, and Tao Kong. ibot: Image BERT pre-training with online tokenizer. *CoRR*, abs/2111.07832, 2021.

[18] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. *CoRR*, abs/1807.10165, 2018.

7 Appendix

Figure 7: T-SNE projection with K-Means clustering and patches from cluster centers

Figure 8: T-SNE projection of features from 10,000 images, with images as points

Visualization and UX

Visual Analytics for Graph Deep Learning: Case Study Neuron Correspondence

Sophie Pichler
Supervised by: Dr. Astrid Berg

VRVis Vienna

Abstract

Many deep learning applications are based on graph data in order to explore relationships or to analyze structures. Labeling this data is expensive and often requires expert knowledge. For the application of graph clustering to neuron data, the SOTA method GraphDINO generates self-supervised graph embeddings combined with the downstream task of clustering these embeddings. We observe on a particularly challenging neuron dataset that this method does not lead to satisfying clustering results. Therefore we use the graph embeddings generated by GraphDINO as an initial starting point to improve the network and to guide the network training. To achieve this, we developed the visual analytics framework *NetDive*. The user can analyze the graph embeddings and label single neurons that are falsely clustered. This annotation information is then used to train a semi-supervised model. To this end, we developed a network architecture, titled *GraphPAWS*, that assembles components of GraphDINO and of the semi-supervised network architecture *PAWS*. The model training can be started from within the visual analytics application NetDive and the resulting graph embeddings are available in NetDive as soon as the retraining is completed. We demonstrate how we iteratively improve the model performance using NetDive and GraphPAWS and evaluate our model against the self-supervised SOTA for our dataset.

Keywords: Visual analytics, Graph embeddings, Graph transformer

1 Introduction

Many deep learning applications are based on graph data, e.g. in the fields of anomaly detection in networks, relationship analyses in social networks and in neuroscience. The use case investigated in this paper is to cluster unlabeled spatial graph data that represents unregistered drosophila melanogaster larval level 1 neurons to reproduce meaningful cell types, addressing the objective of neuroscience to understand the correlation between nerve cells, also named neurons, and behavior [17, 1, 23]. The cell type is an annotation that a neuron receives based on predefined features. Depending on the feature set, the cluster groups vary. Key features explored in the literature are morphology, genetic markers, the neuron position within the nervous system, connectivity and intrinsic electrophysiological signatures [8]. We aim to cluster the neurons solely based on their morphology and aim to find correlations to meaningful cell type assignments as the SOTA method GraphDINO does not produce satisfying results.
This use case embeds in the broader challenge of clustering data without initially having labels to train the deep learning model with a supervised objective function. Experts initially do not know what the network should learn, but want to be able to steer the training while gathering new knowledge about the resulting clusters. This leads us to design a pipeline that addresses these problems by steering the training of the graph network through incremental analysis of the generated embeddings and by incorporating the new knowledge back into the training process.

Contribution: We develop a semi-supervised network architecture *GraphPAWS* that adopts the graph encoder of the self-supervised deep learning architecture GraphDINO and the processing of support samples of the semi-supervised deep learning architecture PAWS. The support samples are sparse annotations for the input data and the count of support samples is variable. The resulting graph latent embeddings are visualized in a visual analytics (VA) web application we title *NetDive* that we developed to analyze the embeddings, to iteratively add new support samples if needed and to retrain a model with this new information. For the evaluation we use manually labeled neurons to compute the performance analytically and we combine this with visual inspection and comparative analysis enabled by the VA application. The GraphPAWS architecture is applicable to a broader range of graph clustering tasks and NetDive is partly data type agnostic and therefore applicable also to embeddings of other input data types.

2 Background and Related Work

Our work combines graph representation learning with VA to utilize the human in the loop to incrementally improve

the network performance. We use the labels generated within the VA application as support samples for the semi-supervised network architecture, whilst we make use of *contrastive learning* to process the unlabeled input samples.

2.1 Contrastive Learning

Contrastive learning belongs to the most successful self-supervised learning methods. It involves the derivation of *supervisory signals* from the input data to guide the learning process [14].

Contrastive learning techniques optimize the model output by embedding the latent representations of variations of the same input sample close to each other, while increasing the distance between the embeddings of different input samples. The pairs of samples that are either attracted or repelled by each other are titled *positive pair* or *negative pair* respectively [14]. Phuc Le-Khac et al. [12] explain, that contrastive learning is not about learning from individual samples, but instead from *comparing* multiple samples. Positive pairs are generated by applying data augmentations to an input sample to get variants of input data that are considered *similar*.

The original non-augmented input sample is called *anchor view* and the augmented variant is referred to as the *positive view*. Negative pairs are generally formed by comparing the anchor view with all the other input samples. If contrastive learning is solely based on positive views [21], the model architecture needs to ensure that the latent representations do not collapse to a single node in the embedding space. This phenomenon is called *node collapsing*. Another force needs to increase the space between different samples.

A contrastive model includes an *encoder* that maps the input view x $\in$ X to a representation vector v $\in R^d$ and a *transform head* $h(v;\Phi_h) : V \rightarrow Z$, where Φ represents the model parameters, that are either used to aggregate features from multiple representation vectors or to reduce the dimensionality of a feature representation vector [12].

Prominent models that use contrastive learning to learn image representations are SimCLR [6], MoCo [10], BYOL [9], SwAV [4], PIRL [20] and DINO [5]. GraphCL [24] and GraphDINO [21] are examples for contrastive models that process graph data.

2.2 Visual Analytics for Latent Embeddings

There are in general two user groups in the field of visual analytics (VA) for deep learning [3]: model-driven users that compare model performances and data-driven users that study properties of the underlying data. A crucial criteria for VA applications is *global check and local check*.

Addressing this, a popular approach to compare embedding spaces is the comparison of local neighborhoods of individual objects in combination with a global comparison of the embeddings [11, 3]. The global embedding comparisons are typically implemented using scatter plots that are interlinked with detail views of selected objects [11]. Therefore dimensionality reduction algorithms are used to map the high-dimensional data in 2D or in 3D. The most common dimensionality reduction algorithms are PCA, tSNE, and UMAP [19]. Boggust et al. [3] discovered, that users prefer deterministic dimensionality reduction algorithms and that they distrust t-SNE and therefore use PCA dimensionality reduction as the default setting for the global projection. The visual analytics tool EmbComp [11] implements a binning feature for the scatter plots to manage the scale of big datasets. The scatter plots can be investigated using single object selection or multiple object selection using for example a rectangle selection tool [11, 15].

The investigation of local neighborhoods is built upon varying metrics. EmbComp visualizes point-wise comparison metrics and distribution comparison metrics. The metrics are visualized in bins which can be selected by the user to select the corresponding objects. The Embedding Comparator [3] visualizes metrics corresponding to the local neighborhoods of a selected datapoint with a histogram of scores, with color-encoding in the global embedding plots and with local neighborhood dominoes, i.e. multiple small visualizations. These small visualizations can be filtered and linked views enable the comparison between visualizations. The Embedding Comparator highlights datapoints with least and highest similarities to address the concern of users stating that they make object selections in an unprincipled way and might miss important correlations between the embedding spaces. Emblaze [19] states, that the Embedding Comparator lacks in finding relevant neighborhoods and addresses this issue in their application. The novel approach of Emblaze is comparison of embedding spaces using Star Trail augmentation. The trails connect the embeddings of the same object in different embedding spaces and the transition between the spaces can be animated using a slider. The connection lines, i.e. Star Trails, between the object embeddings quickly reveal datapoints that vary the most between multiple embedding spaces.

While partly being data type agnostic, the Embedding Comparator, EmbComp and Emblaze as well as many other lines of research regarding visual analytics for embeddings focus on NLP use cases.

In the field of graph embeddings the tools EmbeddingVis [13], CorGIE [15], GEMvis [7] and BiaScope [18] were developed, which are focused on node embedding.

CorGIE [15] encodes the graph nodes and trains a GNN to embed the nodes in the latent space. The user can interact with the node embeddings and select clusters of nodes using a rectangle selection tool. The selection leads to a topology space and feature space analysis. Regarding the topology space, the k-hop neighbors, i.e. the neighbors that are reachable by walking along a path with k topological hops, of the selected nodes are visualized within a visualization of the original graph. The user can evalu-

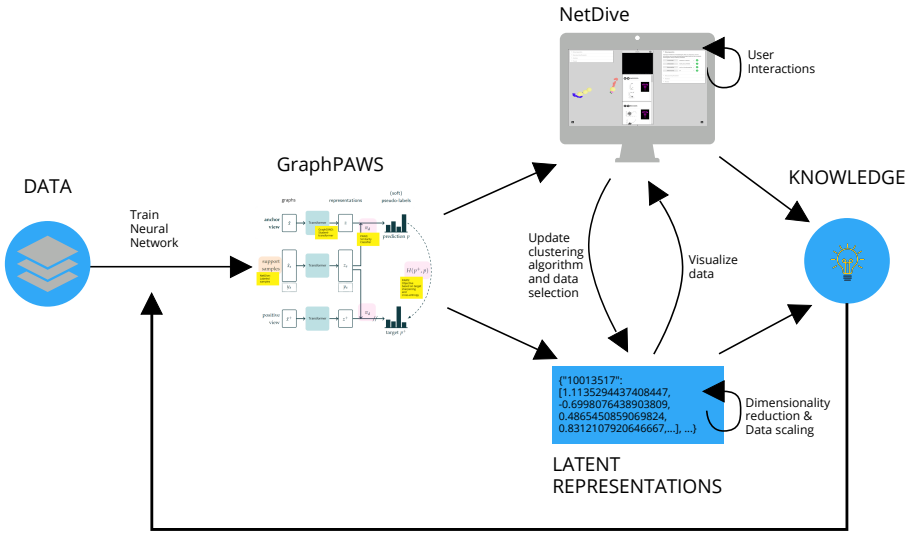


Figure 1: Our pipeline including our visual analytics tool NetDive and our model architecture GraphPAWS.

ate whether the node embeddings corresponds to the topological closeness. The feature space analysis panel shows histograms of feature value distributions of the selected nodes.

GEMvis [7] also interlinks a visualization of the original graph and the node embeddings. The selection of nodes can be applied regarding predefined node metrics. Chen et al. define 9 node metrics, including the node degree and node eccentricity and the node closeness. The metric values for each node are depicted in parallel coordinate plots. The user can interact with these plots to select the according nodes in the original graph and in the embedding space.

While the aforementioned applications EmbeddingVis, CorGIE, GEMvis and BiaScope are developed for node embeddings, we implement an application for whole graph embeddings. Advanced applications exist to leverage VA to compare and analyze the embeddings generated with deep learning models. We add the component of dynamically adding new labels to retrain the model while exploring the latent space that the input graphs are embedded in. We focus the usage of VA for artificial intelligence (AI) for the specific case, in which ground truth is difficult to gather and can only be provided to nudge the training in the right direction. We furthermore integrate detail views specific to the use case of exploring graph embeddings.

3 Methodology

Figure 1 depicts the pipeline that we set up to incrementally gain new knowledge in order to cluster graph data.

The preprocessed data serves as input data to train, validate and test the GraphPAWS model. GraphPAWS is discussed in Section 3.1. The model outputs latent representations of the input graphs. We store the latent representations on the filesystem and the visual analytics application NetDive, discussed in Section 3.2, accesses the data and provides the user with visualizations and user interactions to explore the latent embeddings and the associated neurons. This leads to new knowledge that the user can leverage to retrain the GraphPAWS model.

3.1 GraphPAWS

Our architecture GraphPAWS adopts the graph transformer components of GraphDINO and the semi-supervised architecture of PAWS.

3.1.1 GraphDINO: Self-Supervised Learning for Graph Data

The GraphDINO network [22] implements self-supervised contrastive learning based on transformer networks to find similarities between graphs based on the graph topology and spatial node information. GraphDINO is an adaptation of the DINO network for image data [5].

The GraphDINO model builds upon a student-teacher architecture that is used to generate latent representations of an input graph x. Both the teacher and the student process variations of x. The variations x_1 and x_2 are subsampled to a fixed number of nodes. Graph x_2 is passed to the teacher encoder and graph x_1 is augmented before being passed to the student encoder. The augmentations that are used are subsampling, rotation, node jittering, subgraph deletion, cumulative jittering and a random translation of the soma depth.

The student and the teacher network are identically initialized transformer networks that use the normalized Laplacian for the positional encoding. The outputs of the student and the teacher network are the latent representations z_1 and z_2 respectively. The multi-layer perceptron (MLP) implements a normalization layer and a linear layer to translate the latent representations z_1 and z_2 to p_1 and

p_2. The objective of the network is to decrease the loss that measures the similarity of p_1 and p_2, while not resulting in node collapsing.

GraphDINO uses a cross-entropy loss to measure how similar the latent embeddings of a sample p_1 and p_2 are.

3.1.2 PAWS: Semi-Supervised Learning for Image Data

PAWS [2] implements a semi-supervised deep learning architecture based on contrastive views and support samples to assign one-hot encoded pseudo labels to input images.

PAWS implements three processing streams. Similar to GraphDINO, it processes an input sample and an augmented version of the input sample in order to train invariances that lead to network generalization. PAWS implements the image augmentations random crop, horizontal flip, color distortion and blur. Additionally a mini-batch of labeled support samples is processed in the third stream. The support samples are annotated samples that function as prototype samples for a cluster. PAWS assigns a pseudo label based on the similarity of the latent embeddings of the anchor view and the positive view in relation to the support samples. PAWS expects each mini-batch to be composed by an equal number of instances for each sampled class.

The objective function uses the cross entropy function to measure the similarity of the pseudo-labels of the anchor view and the positive view. To avoid node collapsing, PAWS uses sharpening in the objective function. The sharpening function increases the confidence of the probability distributions, i.e. decreases the entropy.

Additionally the objective function adds a regularization term, titled *mean entropy maximization (ME-MAX)* that aims to increase the entropy of an unlabeled training-batch, to ensure that each label is getting predicted. More concretely, distributions like $[[1.,0.,0.],[1.,0.,0.],[1.,0.,0.]]$ are penalized and distributions like $[[1.,0.,0.],[0.,1.,0.],[0.,0.,1.]]$ are favoured.

3.1.3 GraphPAWS: Semi-Supervised Learning for Graph Data

Our architecture GraphPAWS, depicted in Figure 2, is an adaptation of GraphDINO and PAWS. We adopt the PAWS architecture that processes an anchor view $\hat{x}$, a positive view $\hat{x}^+$ and a support sample mini-batch $\hat{x}_s$. We replace the encoder of PAWS with the GraphDINO graph transformer. The encoder generates the latent embeddings z for input $\hat{x}$ and respectively z_s and z^+ for the support sample mini-batch and the positive view. The support samples are fed into the similarity classifier to compute the pseudo-labels p and respectively p^+ for the anchor view and the positive view. While PAWS expects the support sample mini-batch to be balanced, we implement weight-balancing in the GraphPAWS adaptation of the similarity

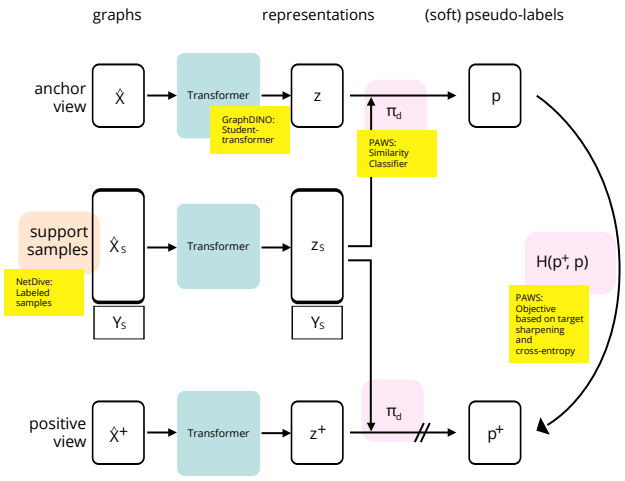


Figure 2: The semi-supervised GraphPAWS architecture for graph data.

classifier to compensate for class imbalances. This gives the user more flexibility while annotating neuron graphs.

The objective function is denoted with $H(p^+, p)$ in Figure 2. It corresponds to the objective function implemented by PAWS [2], which is based on cross entropy. We additionally train on the mean-squared error (mse) objective function.

The regularization term ME-MAX, implemented in PAWS, is added both to the cross-entropy and the mse objective function. We add a second regularization term that we title *One-Hot-Enforcement*, which enforces one-hot encodings of the embedded vectors. While ME-MAX operates over a batch of samples, One-Hot-Enforcement is applied to single training samples and averaged over a batch.

Equation 1 depicts the objective function in relation to the hyperparameters λ and γ that determine the relevance of the regularization terms *ME-MAX* and *One-Hot-Enforcement*,

$$\text{loss} + \lambda * \text{ME-MAX} + \gamma * \text{One-Hot-Enforcement}. \quad (1)$$

After training the latent embeddings are evaluated by latent space clustering through GMM or k-means and evaluated against ground truth labels.

3.2 NetDive

NetDive is developed for model engineers that design and refine the model and for domain experts to explore the graph data and to choose a model from the pre-trained model database. NetDive consists of a backend server to read and write data from and to the filesystem and a React frontend that communicates with the backend. On demand, i.e. using the refresh buttons in the user interface, the backend applies dimensionality reduction to the requested pre-computed latent embeddings. The frontend

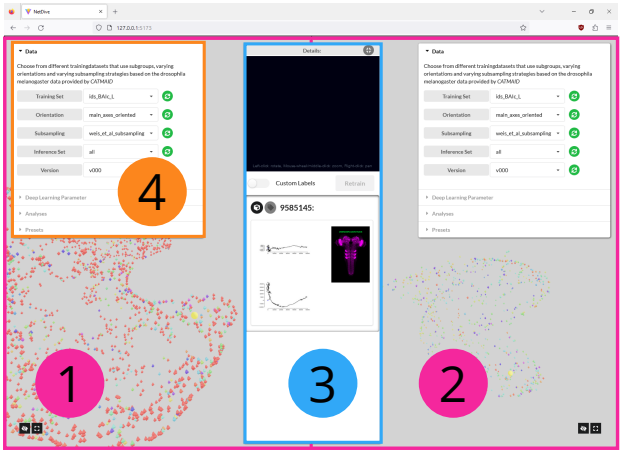


Figure 3: NetDive layout: (1) First 3D View, (2) Second 3D View, (3) Parameter Panel, (4) Detail View.

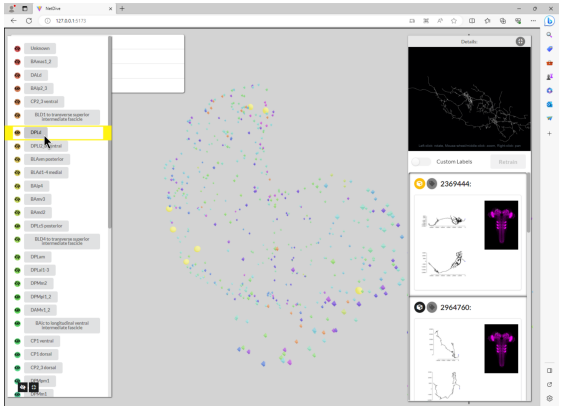


Figure 4: Selecting a lineage using the legend panel.

visualizes the dimensionality reduced latent embeddings in three dimensions. We chose to visualize the data in three instead of two dimensions to achieve a clearer cluster separation. This can come with the downside of distortion and occlusion. As clustering is not dealing with issues of length and angle preservation, and the points representing our use case of neurons are not covering a lot of space, which limits the issue of occlusion, we accepted these downsides. The scatterplots are implemented with THREE.JS. We use the clustering algorithms k-means and GMM to display the cluster predictions by color coding the datapoints in the scatterplots. The clustering algorithms are applied in the backend to the k latent dimensions that GraphDINO outputs before applying the dimensionality reduction. The prediction labels support the user to evaluate the quality of the latent embedding space.

3.2.1 User Interface

Figure 3 depicts the layout of NetDive. The user interface (UI) consists of three panels. Two views, annotated with (1) and (2), and a detail panel annotated with (3). The two view panels provide the user with parameters embedded in an accordion menu, annotated with (4), to select a model and analyses values to load and explore the latent embeddings in form of scatterplots in 3D generated by the corresponding selected model.

The user can choose between UMAP, t-SNE and PCA to reduce the 32 latent dimensions to three dimensions. Following Boggust et al. [3] we set PCA as default value. The datapoints in the scatterplots are color coded. The color of the datapoints is either assigned based on a selected ground truth or based on the cluster predictions generated with the selected clustering algorithm. The implemented clustering algorithms are k-means and Gaussian Mixture Models (GMM). The user can select the number of clusters to generate. The color codes can be used to toggle between the ground truth and the predicted clusters in order to detect similarities and dissimilarities. The color codes further aid with the comparative analyses using the two views, annotated with (1) and (2) in Figure 3.

Expandable legends, depicted in Figure 4, list the cluster labels that represent the latent embeddings. The labels are associated with the selected color codes. In Figure 4 the Hartenstein lineages, discussed in Section 4.1, are selected as a ground truth and the color codes correspond to the ground truth. The user can hide/show and select all datapoints with specific color codes within the legends.

View (1), view (2) and the detail panel are linked, implementing the concept of multiple linked views (MLVs), displayed in juxtaposition. When a user selects single or multiple latent embeddings, the corresponding datapoint in the other view is highlighted, if present, and the detail panel depicts a scrollable list of tiles depicting information about the selected node(s). The tile headers show datapoint identifier, a button to select the according datapoint for a 3D graph rendering, which is displayed on top of the detail panel, and a button to add or update the datapoint label. The tile content shows pre-rendered images of the selected datapoint.

3.2.2 Relabeling and Retraining

After initially activating the relabeling feature using the *Relabeling* slider in the details panel, all datapoints are rendered gray and the color coding is disabled.

The user can then select embedded points and relabel them in the relabeling modal. The modal, i.e. the overlay, is depicted in Figure 5. The user can either create a new cluster group and add the selected id for the embedded graph to that group or they can add it to an existing cluster group.

The new labels are forwarded to the network training. The training is triggered within NetDive and processed using a *subprocess call*. The default hyperparameters correspond to hyperparameters of the currently loaded model and the user can update the network hyperparameters for the training within NetDive in the retraining modal. After

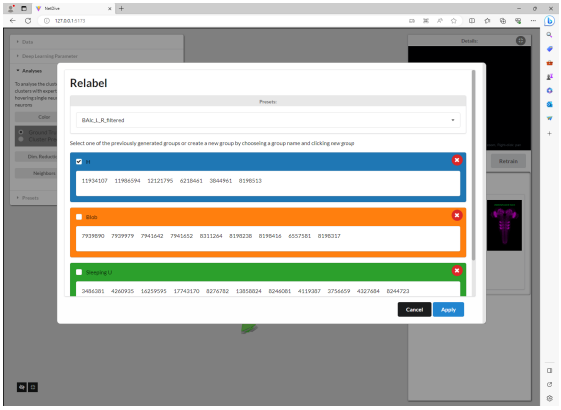


Figure 5: Relabeling neurons in NetDive.

the training is completed, the embeddings generated with the new model for the chosen inference set will be available in NetDive.

NetDive aims to make the analyses and the iterative training fast and intuitive.

4 Experiments

We conduct a series of experiments to evaluate our dataset (Section 4.1) on the self-supervised SOTA architecture GraphDINO (Section 4.2) and on our architecture GraphPAWS (Section 4.3).

4.1 Data

The drosophila melanogaster neuron graphs extracted from CATMAID, a platform for a collaborative reconstruction and annotation of data, are represented as undirected, acyclic graphs in three dimensions with the root node representing the soma, i.e. the cell body of the neuron.

We obtained the set of all available 7297 drosophila melanogaster larval neurons from CATMAID and restricted our analysis to a subset of 2970 neurons that was annotated by Michael Winding [23], as this subset contains more reliable neuron traces. We reduced this subset to 2541 neurons by removing all neurons that do not contain exactly one node annotated as *soma* and by removing all neurons with less than 200 nodes. CATMAID provides the neuron graphs as SWC files and we keep the (x,y,z) for each node of the neuron graph.

For the ground truth we generated a file that stores multiple ground truth cell type labels for each neuron id. The annotation files includes manually labeled cell types that we hand-crafted based on visual inspection. Furthermore it includes expert annotation by Dr. Volker Hartenstein [16]. Dr. Volker Hartenstein analyzed lineages, that describe neurons deriving from the same stem cells called *neuropblasts*. He states, that neurons within a lineage do not only share the same stem cell, but are also alike regarding the morphology. The datasets we define in out experiments are based on these lineages annotated by Dr. Hartenstein. Lineage BAlc neurons are located in the lateral surface of the antennal lobe and lineage CM4 in the postero-medial brain cortex.

We conduct our experiments on two subsets of the drosophila melanogaster dataset. We use the lineages BAlc and CM4 that Dr. Volker Hartenstein specified and divide these lineages in visually similar subgroups. Some lineages are visually coherent, whilst the lineages BAlc and CM4 fall into visually distinguishable groups. Lineage BAlc consists of 26 neurons, 13 in each brain hemisphere, and divides in three cluster groups. CM4 contains 66 neurons, 33 in each brain hemisphere, and divides in four cluster groups.

4.2 Self-supervised Training

We train GraphDINO for the learning rates $\in \{0.001, 0.0001, 0.00001\}$. While Weis et al. train on batch size $\in \{32, 64, 128\}$ we train on batch size $\in \{16, 32\}$ due to the smaller training dataset. We train with a 60-20-20 training-validation-test split on dataset *BAlc*.

We evaluate with 4 fold cross-validation for k-means and for GMM on the validation data by averaging over 100 k-means / GMM adjusted random index (ARI) scores per fold. ARI measures the similarity between two clusterings. We use the ARI computation of the python library *sklearn*, which outputs values between -0.5 for especially discordant clusterings and 1.0 for identical clusterings.

4.3 Semi-supervised Training

We trained 896 models using a grid-search on GraphPAWS for the dimensions loss function, ME-MAX influence λ, One-Hot-Enforcement influence γ, batch size and learning rate. We used the values ['cross_entropy', 'mse'] for the loss, the values [0, 0.1, 0.5, 1] for λ and γ, the values [0.001, 0.003, 0.006, 0.0001, 0.00006, 0.00003, 0.00001] for the learning rate and the values [4, 8, 16, 32] for the batch size. We ran the hyperparameter search for 100 epochs. Accordingly to the self-supervised GraphDINO training, we train with a 60-20-20 training-validation-test split on dataset *BAlc*.

We evaluate with 4 fold cross-validation for k-means and for GMM on the validation data by averaging over 100 k-means / GMM ARI scores per fold. We list the top performing models and eliminate the models that suffer from node collapsing and from an incapability to learn. We therefore analyzed the feature distributions of the latent embeddings and the loss curve. Figure 6 depicts a loss curve with downwards trend on the top right side, indicating that the model learns, while the loss curve on the top left side does not decrease. On the bottom left Figure 6 depicts the feature distributions of the latent embeddings

that have marginal standard deviations, indicating that all latent embeddings collapse to the same representation. On the bottom right side the features are well distributed.

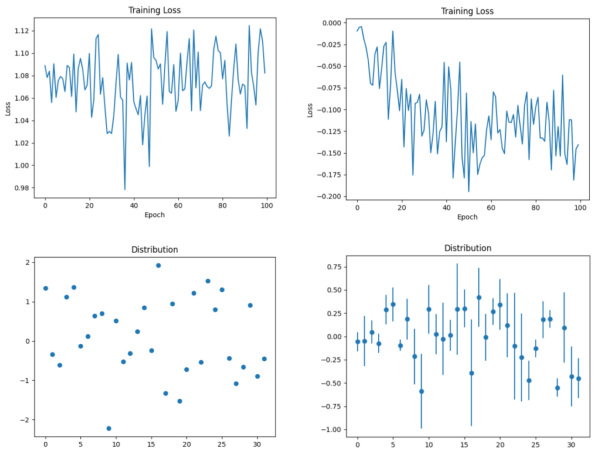


Figure 6: Example loss curves and feature distribution plots of a model suffering from node collapsing (left) and a model showing decreasing loss trend and distributed features (right).

5 Results and Evaluation

The GraphDINO model with learning rate 0.0001 and batch size 16 has the overall best ARI performance on k-means clustering with a score of 0.495.

The GraphPAWS model with learning rate 3e-05, batch size 32, gamma 1 and lambda 1 has the overall best ARI performance on GMM clustering with a score of 0.527. We repeated the training for these hyperparameters five times. The resulting ARI scores based on GMM clustering varied between 0.379 and 0.591. We have to consider this variance of performance when we evaluate the results.

Table 1 compares the ARI scores of the optimal self-supervised trained model with the optimal semi-supervised trained model.

	Self-Supervised Training	**Semi-Supervised Training**
Loss	cross entropy	mse
Learning Rate	0.0001	3e-05
Batch Size	16	32
Gamma	-	1
Lambda	-	1
ARI	0.495 (k-means)	0.527 (GMM)

Table 1: Results of optimal self-supervised trained model and semi-supervised trained model. The models are trained on lineage BAlc.

After determining the optimal GraphPAWS model we use the same model to train on a different lineage, i.e. lineage CM4, to demonstrate the usage of NetDive to iteratively explore the data and to feed new knowledge into the training.

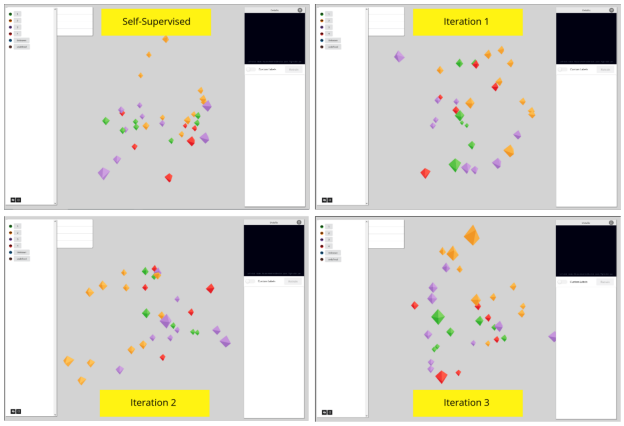


Figure 7: Neuron graph embeddings after each iteration denoted in Table 2.

For the NetDive evaluation instead of performing cross-validation, we train on the whole dataset and evaluate using the manually annotated CM4 samples that were partly also used as support samples during the semi-supervised training.

Initially we simulate the case that no labeled data is available and therefore train with the self-supervised GraphDINO model. We load the dimensionality reduced latent representations of the CM4 neuron graphs into NetDive and analyze the embeddings.

We explore how the ground truth clusters differ from the predicted clusters and label samples that are most distant from the visual cluster centroids. The ground truth we generated is for evaluation purposes only and is not available in a real use case. We then retrain a model on the GraphPAWS architecture with the annotated samples. We do this in three iterations and we add additional support samples in each iteration.

Figure 7 depicts the embeddings after each iteration, colored based on the CM4 ground truth. We cannot recognize a clear subdivision into clusters. We must therefore be cautious in assessing the positive trend in the improvement of ARI scores, reported in Table 2. Table 2 denoted the ARI scores based on k-means and GMM clustering after each iteration and lists the neuron ids of the support samples used for each iteration.

6 Discussion and Conclusion

In this paper, we established a workflow to address the problem of clustering graph data without initially having a ground truth for training whilst giving the user the possibility to guide the training process with minimal effort.

After the grid search that we performed in order to find the optimal GraphPAWS hyperparameters, we had to elim-

	Self-Supervised [22]	Iteration 1	Iteration 2	Iteration 3
ARI: **GMM**	0.152	0.145	0.143	0.225
ARI: **kmeans**	0.117	0.211	0.191	0.317
#Support Samples	-	4: 1 per class	8: 2 per class	12: 3 per class

Table 2: ARI scores of incremental training with NetDive. The ARI computation is based on manual ground truth for evaluation purposes only. The training is performed onlineage CM4.

inate models that had good ARI scores but which suffered from node collapsing and a lack of learning capability. These models sometimes had high ARI scores per coincidence.

While we used feature distribution visualizations and the loss curve plots to evaluate the models, these effects should also be visible in NetDive, as the embedding space would not divide in distinct clusters.

The results we achieved with GraphPAWS are not yet convincing. As documented in Section 5, we see an improvement reflected in the ARI scores (Table 2), but this effect is not clearly reflected in the NetDive clustering (Figure 7).

In order to address this, it would be an interesting future work, to further investigate the model optimization of GraphPAWS using NetDive, as we see indicators, that the pipeline that involves labeling support samples and restarting the training is intuitive and effective. We suspect that training on bigger datasets would eliminate outlier models and reduce the variance of performance for models trained on identical hyperparameters, reported in Section 4.3. Furthermore we want to experiment with fine-tuning the model after adding new support samples, instead of training new randomly initialized models, and therefore reduce training times. We also want to employ alternative subsampling strategies to reduce the input graphs to a fixed amount of nodes by evenly distributing the resampled nodes.

The NetDive user interface can be improved by adding simulations that visualize the cluster changes over time during training with color updates. It is also possible to add more characteristics of the neurons in the details Section and provide interaction techniques like brushing and linking over a feature space visualization for neurons to understand correlations between clusters and the cluster contents. We can extend the spatial representations and use the properties size and opacity of each data point to encode additional information besides the cluster label, e.g. the certainty of the cluster assignment in the opacity and the variance over a sequence of models in the size of the data point.

Regarding the evaluation we want to perform user studies with experts in the field of neuroscience to see how users outside the domain of deep learning can use visual analytics to refine pre-trained models and which features they are missing in the current NetDive setup.

References

[1] Merriam-Webster neuroscience. www.merriam-webster.com/dictionary/neuroscience. Accessed: 2023-04-12.

[2] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, October 2021.

[3] Angie Boggust, Brandon Carter, and Arvind Satyanarayan. Embedding Comparator: Visualizing Differences in Global Structure and Local Neighborhoods via Small Multiples. *27th International Conference on Intelligent User Interfaces*, pages 746–766, March 2022.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, pages 9912–9924, Red Hook, NY, USA, December 2020. Curran Associates Inc.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, October 2021.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML'20*, pages 1597–1607. JMLR.org, July 2020.

[7] Yi Chen, Qinghui Zhang, Zeli Guan, Ying Zhao, and Wei Chen. GEMvis: a visual analysis method for the comparison and refinement of graph embedding models. *The Visual Computer*, 38(9-10):3449–3462, September 2022.

[8] Marta Costa, James D. Manton, Aaron D. Ostrovsky, Steffen Prohaska, and Gregory S.X.E. Jefferis. NBLAST: Rapid, sensitive comparison of neuronal structure and construction of neuron family databases. *Neuron*, 91(2):293–311, July 2016.

[9] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent a new approach to self-supervised learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, pages 21271–21284, Red Hook, NY, USA, December 2020. Curran Associates Inc.

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. Computer Vision Foundation / IEEE, 2020.

[11] Florian Heimerl, Christoph Kralj, Torsten Moller, and Michael Gleicher. embcomp : Visual Interactive Comparison of Vector Embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 28(8):2953–2969, August 2022.

[12] Phuc Le-Khac, Graham Healy, and Alan Smeaton. *Contrastive Representation Learning: A Framework and Review*. October 2020.

[13] Quan Li, Kristanto Sean Njotoprawiro, Hammad Haleem, Qiaoan Chen, Chris Yi, and Xiaojuan Ma. EmbeddingVis: A Visual Analytics Approach to Comparative Network Embedding Inspection. *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 48–59, October 2018.

[14] Xu Liu, Yuxuan Liang, Chao Huang, Yu Zheng, Bryan Hooi, and Roger Zimmermann. When do contrastive learning signals help spatio-temporal graph forecasting? In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. ACM, November 2022.

[15] Zipeng Liu, Yang Wang, Jurgen Bernard, and Tamara Munzner. Visualizing Graph Neural Networks with CorGIE: Corresponding a Graph to Its Embedding. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2022.

[16] J. K. Lovick, K. T. Ngo, J. J. Omoto, D. C. Wong, J. D. Nguyen, and V. Hartenstein. Postembryonic lineages of the drosophila brain: I. development of the lineage-associated fiber tracts. *Dev Biol.*, 2013.

[17] National Research Council (US) Committee on Research Opportunities in Biology. *Opportunities in Biology*. National Academies Press (US), Washington (DC), 1989.

[18] Agapi Rissaki, Bruno Scarone, David Liu, Aditeya Pandey, Brennan Klein, Tina Eliassi-Rad, and Michelle A. Borkin. BiaScope: Visual Unfairness Diagnosis for Graph Embeddings. *2022 IEEE Visualization in Data Science (VDS)*, pages 27–36, October 2022.

[19] Venkatesh Sivaraman, Yiwei Wu, and Adam Perer. Emblaze: Illuminating Machine Learning Representations through Interactive Comparison of Embedding Spaces. *27th International Conference on Intelligent User Interfaces*, pages 418–432, March 2022.

[20] Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically interpretable reinforcement learning. Technical report, 2018.

[21] Marissa Weis, Laura Hansel, Timo Lüddecke, and Alexander Ecker. *Self-supervised Representation Learning of Neuronal Morphologies*. December 2021.

[22] Marissa A. Weis, Laura Pede, Timo Lüddecke, and Alexander S Ecker. Self-supervised graph representation learning for neuronal morphologies. *Transactions on Machine Learning Research*, 2023.

[23] Michael Winding, Benjamin D. Pedigo, Christopher L. Barnes, Heather G. Patsolic, Youngser Park, Tom Kazimiers, Akira Fushiki, Ingrid V. Andrade, Avinash Khandelwal, Javier Valdes-Aleman, Feng Li, Nadine Randel, Elizabeth Barsotti, Ana Correia, Richard D. Fetter, Volker Hartenstein, Carey E. Priebe, Joshua T. Vogelstein, Albert Cardona, and Marta Zlatic. The connectome of an insect brain. *Science*, 379(6636):eadd9330, March 2023.

[24] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Adapting Design Methodology to Enhance Physician Workflow in IBD EHR Systems

Lucia Ondovčíková*
Supervised by: Miroslav Laco†

Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava / Slovakia

Abstract

Electronic Health Records have the potential to enhance service delivery in the healthcare sector, yet their complexity frequently leads to inefficiencies and user resistance. A major challenge in this context is the identification and interpretation of correlations between disease-specific parameters, requiring domain expertise. In response to these challenges, a collaborative research effort with domain specialists is underway to develop a Domain Expert-Oriented Design of a User Interface for the registry of Inflammatory Bowel Disease (IBD) patients. This tool aims to be both a database and a basic statistical tool, designed in the physician's natural working environment. The development of this tool is guided by a modified Patient Experience Design framework that is tailored to the specific requirements of healthcare professionals and uses a customized version of the Double Diamond Design model to improve data collection from clinicians and enhance effective interdisciplinary collaboration. This research initiative represents a potentially important contribution to the field of IBD, especially given the absence of existing tools offering comparable functionality.

Keywords: User Experience, Domain Expert Centered Design, Electronic Health Records, Inflammatory Bowel Disease

1 Introduction

User Experience (UX) [3, 11, 13] in healthcare impacts the efficiency and effectiveness of patient care, particularly in managing complex diseases like Inflammatory Bowel Disease (IBD). A well-designed UX is essential for improving diagnostic precision and optimizing clinical workflows, leading to better health outcomes [6]. However, the treatment and management of IBD are often challenged by the limitations of existing Electronic Health Record (EHR) systems, which usually struggle with usability issues due to their complex designs, potentially impacting the decision-making processes of healthcare professionals.

This article aims to address these challenges by designing a user interface for the EHR system, specifically tailored for IBD management, involving domain experts during the design process. It focuses on overcoming existing problems by emphasizing explainability, which ensures that the system's data is easily understandable and interpretable by healthcare professionals, thereby facilitating more informed decision-making. The intention behind this methodology centered on domain experts is to enhance the management and understanding of IBD, ensuring an optimal UX for the professionals involved in this domain.

2 Related work

To ensure that the result of our work is effective in delivering a positive user experience, it is important to select an appropriate design methodology. Various approaches, including the one introduced by Sedlmair et al.[14], provide a foundational framework for the development of domain-specific methodologies. In this context, our work aims to explore the integration of a Domain Expert Centered Design methodology, inspired by the principles of Patient Experience Design. Secondly, we need to highlight the importance of data visualization consistency within EHRs. Finally, we analyze the available solutions that have partially attempted to address these issues.

2.1 Patient Experience Design

The standard double diamond approach [2, 1, 10], commonly used in domain-specific areas like medicine, proves insufficient, leading to the development of the Patient Experience Design (PXD). This methodology represents a major change in how healthcare services are designed and delivered, focusing primarily on the needs of the patient. It marks a significant shift from traditional healthcare models, where patient interactions were often secondary to clinical procedures and operational efficiency.

Lisa K. Meloncon defines PXD as a systematic approach aimed at exploring the relationships between technology and human activities in healthcare [12, 9]. As

*lucia.ondovcikova@gmail.com
†miroslav.laco@gmail.com

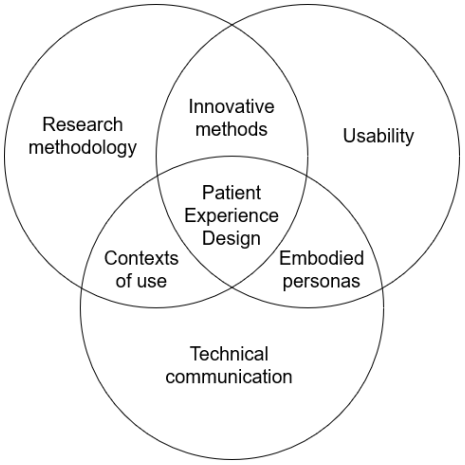


Figure 1: Knowledge domains of PXD with primary concepts [12].

demonstrated by Meloncon in Figure 1, PXD brings together 3 main domains of knowledge: Research methodology, Usability, and Technical communication. The author details them in her study as outlined below [12]. **Research methodology** refers to the systematic methods and approaches used to collect information and insights that are subsequently applied in the next phases of development. **Usability** domain emphasizes the need to create patient experiences that are intuitive, efficient, and satisfying. **Technical communication** encompasses the communication of complex medical information to patients in a way that is easily understandable for them. The author further explains that the overlapping areas are the primary concepts of PXD that are drawn from each domain of knowledge [12]. Understanding the **Context of use** is important for designing experiences that are relevant and tailored to the patient's needs. **Embodied personas** emphasizes a more detailed understanding of personas - including the physical and emotional state of the users. Usage of **Innovative methods** (use of new technologies, unconventional approaches to problem-solving) can affect the resulting experience with the product.

2.2 Design and Visualization Techniques in EHR Systems

The usability and effectiveness of EHR systems are heavily dependent on how data is visualized and interacted with by domain experts [8]. Kenichiro Fujita et al. [6] highlight the importance of identifying key attributes in these records for effective processing and management. The most common attributes are patient identifiers, timestamps of interactions within the EHR system, and the specific primary data types relevant to the current system, which are subsequently broken down into more specific elements, as shown in Figure 2. Decomposing complex concepts into smaller, more manageable parts, facilitates a deeper understanding of the data and enhances their visualization.

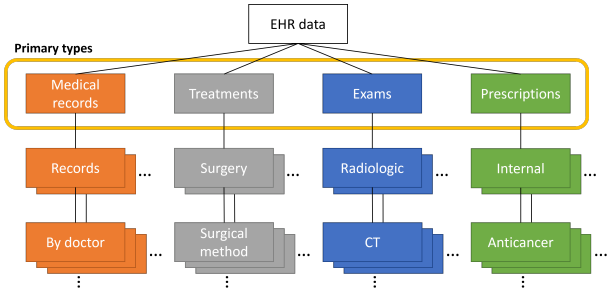


Figure 2: The example tree structure of EHR data types and primary types [6].

Based on the identified attributes, the author proposes the following screen design principles for effectively displaying EHR data: Displaying single patient data in one view, Summarizing data for an overview and providing details on demand, Displaying data in a time-series format, Categorizing data by primary type, and Displaying more data simultaneously when the above principles are met.

By adopting specific principles such as categorization by data type, time-series display, and summarizing data with detailed views on demand, the author further proposes three distinct screen designs, as shown in Figure 3. All of them use color-coding as a visualization technique, categorizing information by EHR primary types. **Design 1** organizes data in a time-series format. Each data point shows a title and expands to reveal details upon interaction. **Design 2** presents data in a matrix layout with two axes - time-series and primary type. This design incorporates all the principles and displays the most data. **Design 3** arranges data in time-series, where selecting a point displays detailed data.

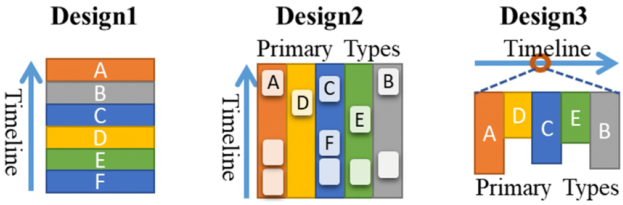


Figure 3: Screen designs using proposed principles [6].

2.3 Research Registries in IBD EHR Systems

Research registries have become essential in advancing medical knowledge, providing comprehensive databases for systematic study. Regarding IBD EHR systems, the only publicly accessible tool is the UR-CARE demo

database[1]. The UR-CARE database [4], initiated by the European Crohn's and Colitis Organisation (ECCO), aims to enhance patient care and support Inflammatory Bowel Disease (IBD) research, standing out as a dedicated platform in this area. Developed through the collaboration of IBD specialists and contributors from national research database projects throughout Europe, UR-CARE facilitates data collection on IBD symptoms and other key patient data. It covers disease characteristics, diagnostic tests like endoscopy, treatments, and lab results. Additionally, UR-CARE enhances user experience by automatically calculating various indicators, such as the Simple Endoscopic Score for Crohn's Disease, from the data provided.

Aside from the patient's health card management, UR-CARE (demo version) consists of 3 other components: Filters, Statistics, and Inboarding website. In the Filter section, users can create filters using existing patient list or use the already created ones, to aggregate interesting information into different groups according to chosen attributes and set conditions. Subsequently, in the Statistics section, these custom filters are utilized to conduct a deeper analysis of the most important attributes. The statistics can be visually represented using descriptive statistics or one of the three available types of graphs: bar chart, pie chart, and evolution chart. However, the data representation is not focused on usability, which makes it difficult for physicians to make well-informed decisions.

3 Contribution

In this paper, we presented a new approach to the prototyping process called Domain Expert Centered Design, arising from the modified Patient Experience Design [12]. Our year-long case study with medical professionals demonstrated the efficiency of this approach, underscoring the importance of a flexible methodology. An important component of our method was the emphasis on contextualizing information during the communication process, which significantly enhanced the development workflow and led to better feedback from domain experts.

As a contribution, we have created an intuitive user interface for a research-oriented Electronic Health Record (EHR) system, designed with a focus on IBD, using our proposed framework. This system stands out for its straightforward data organization and user-friendly design, which aligns closely with physicians' daily routines, aligning closely with their workflow. While the current system contains only core functionality, its potential for evolution into a comprehensive assistant tool for doctors presents an exciting avenue for future development.

[1]Available online, accessed on 10/03/2024: `https://perseed.eu/urcare/index.html`

4 IBD EHR System Design Proposal

The collaboration between domain experts in IBD, encompassing both clinical and surgical expertise, and IT professionals is essential. Domain experts require analytical skills to uncover hidden patterns and achieve effective data matching for research insights, while IT specialists do not have specific domain knowledge and depend on domain experts to provide it. This collaborative approach led to the creation of a research-oriented IBD EHR system.

4.1 Design Methodology

This system is designed using a Domain Expert Centered Design methodology, integrated with an expanded Double Diamond design process [1], to ensure the system is as intuitive as possible. The main goal is to develop a prototype of the system aimed at assisting doctors in monitoring the state of a patient's disease more effectively, while also enabling the aggregation of patient data for further research of IBD. The topics discussed in Section 2 Related Work primarily focus on Patient Experience Design, yet this approach is almost equally applicable to physicians. It is common for individuals to articulate a full idea more effectively when the information's context is visually represented, as is the case here. Hence, we decided to alter the traditional Double Diamond framework [2, 10, 1]. The main idea is that information gathered during each session would be subsequently analyzed and incorporated into the prototype. This means that the process does not follow a predefined order of phases but instead transitions between them as required. This adjustment not only enhances the development process in terms of quality but also elevates the domain expert experience design.

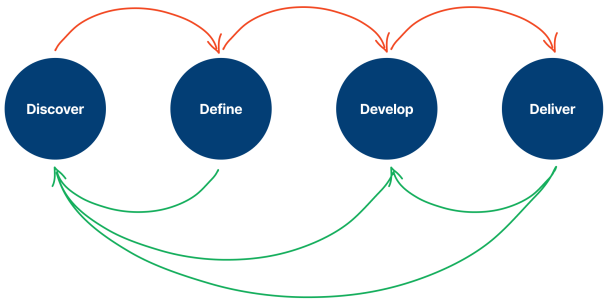


Figure 4: Modified diagram of the Double Diamond. The red color represents the original Double Diamond, and the green color represents extra steps.

During the first phase, **Discover**, the UX team tries to develop a mutual understanding and a shared language between them and medical experts through collaborative interviews and prototype demonstrations [5], which is more challenging than the traditional prototype development, due to the necessary knowledge and expertise in the medical domain. The **Define** phase focuses on information processing. As described before, it's common to find that not

all necessary information is available at this stage. As a result, the gathered information is processed, with previously collected data, and any gaps are addressed in the next meeting. During this phase is also created a specific type of persona called an embodied persona [12]. It enables a deeper and more precise identification of the specific needs and objectives related to the problem, as described in Section 2.1. Collaborating with a domain expert requires expertise in a given area that the average person does not have, leading to the creation of a second persona, the IT development team. The UX team then serves as a mediator, facilitating the exchange of information between the domain experts and the IT team. This mediator approach is not typically observed in the traditional development process, highlighting a unique aspect of our methodology. The rest of the second phase remains the same as in the Double Diamond methodology. In the third phase, **Develop**, the goal is to visualize and refine the ideas generated in the previous meetings. Unlike the traditional approach, prototyping does not follow the usual sequence of prototyping. In this specific case, the process begins with basic layout sketches and quickly progresses to more detailed wireframes-mockup hybrid prototypes. The reason for moving directly to hybrids in this phase is to enable doctors to provide effective feedback. Seeing more detailed screens allows them to comment specifically on what aspects of the design align with their workflow, what differs from their current practices, and what elements might be unnecessary or unhelpful. Feedback obtained from the following consultations is systematically integrated into the design revisions, which is the **Deliver** phase. These changes have to be integrated rapidly, in order to provide better feedback at the next meeting with domain experts. Following multiple iterations of the prototype, when both sides are satisfied, the process progresses to usability testing, which would be conducted according to traditional methodologies.

4.2 IBD EHR System

The collaborative approach described in the previous Section led to the creation of an intuitive user interface for the IBD EHR system. The system proposal incorporates insights gathered from the Section 2 Related Work. The IBD data are systematically categorized as illustrated in Figure 5. Categories arise from the actual process of examination in the IBD domain. Consultations are regular examinations of the patient's overall health, during which physicians may request additional examinations to obtain a current comprehensive overview. Decisions regarding treatment or surgical operation are made during these consultations, hence their inclusion within the consultation category in the tree structure. Examinations like endoscopy or magnetic resonance imaging are also part of the patient's overall condition, but they do not fall under the consultation itself.

In the prototype, the data is placed under the "Consultation and examinations" tab. Within this section, data is presented in two ways: a table and a timeline. For domain experts, the timeline view is more important as it provides a visual representation of the patient's overall condition over time. Our design, based on Design 3 [6] (more in Section 2.2) with slight modifications, plots examination points along the time-series axis. These points are coloured based on examination type, helping doctors in visualization. Clicking on a point reveals detailed examination information, organized by primary types using tabs. The interesting feature of our design is that the timeline also incorporates additional sub-levels of IBD data, including current treatment, surgery, and a specific index for Crohn's disease (pCD). This type of extended timeline assists physicians in decision-making and planning further treatment strategies.

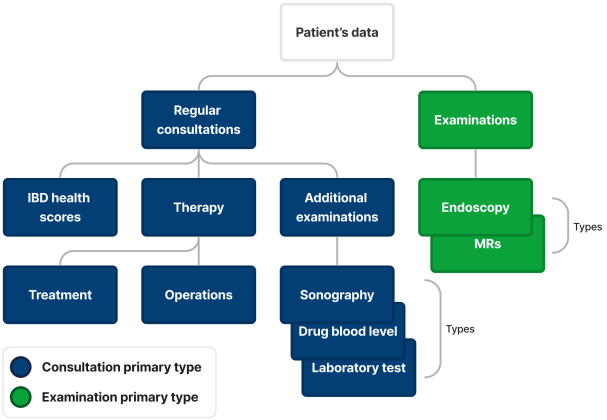


Figure 5: Primary types of IBD data in our work.

Simplifying the process of adding examinations is made more straightforward and user-friendly. In the "Consultation and examinations" section, simply selecting "Add new exam" enables the physician to input essential data without navigating through multiple steps, as in the UR-CARE registry [4]. Notably, our system stores only relevant data to prevent system overload. The research part of the register differs mostly in the study part. Filter creation operates on a similar principle to UR-CARE [4], where doctors specify a parameter and its expected value. In the study part of our IBD registry researchers choose their focus - examining relationships between two variables in a group or comparing a variable across groups. They use an interface for dynamic variable adjustments and a graphical representation, limited to continuous graphs for consistency in study evaluation. In contrast, the UR-CARE system [4] is limited to using just one filter per study, enabling users to choose and graph multiple variables. These are then displayed sequentially, which may affect their readability.

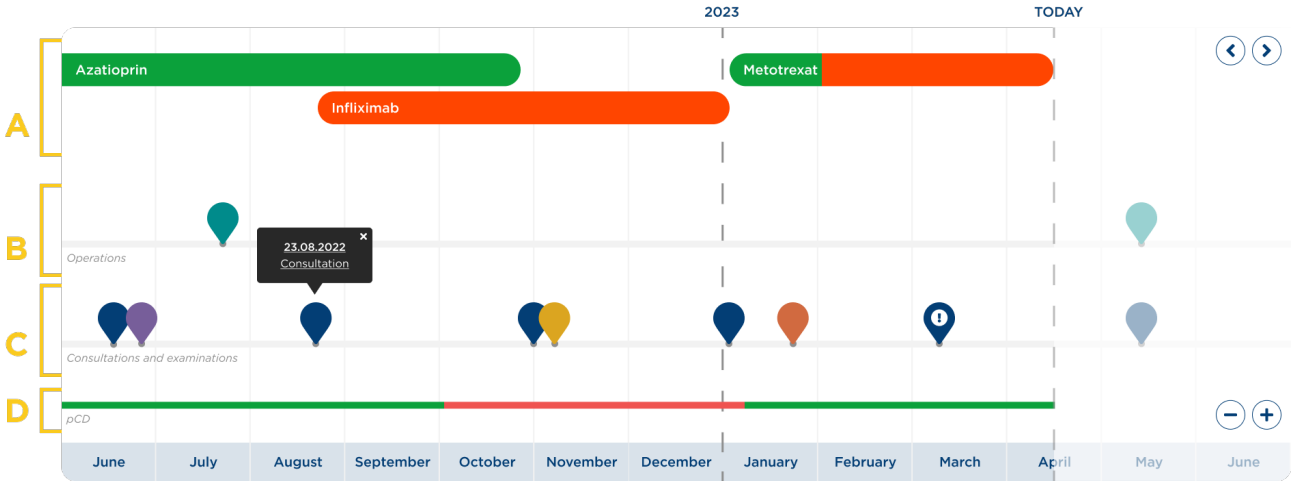


Figure 6: Representation of a patient's IBD data in our work using time-series. The C section represents primary data, coloured by examination type. A Section represents treatment, the B section is for operations and D is the specific index for Crohn's disease. A, B and D sections offer complementary information to the C section.

5 Case Study

The methodologies and design process we proposed were validated through a case study lasting over a year, during which the design process for the research-oriented IBD EHR system was applied to create the user interface design and specifications for iterative agile development.

5.1 Domain-specific Observation

The primary objective of the observation was to gain essential knowledge for the project's development. This included establishing a shared language, identifying the needs and goals of domain experts, and ensuring that before each meeting, the UX team had analyzed information from previous discussions and outlined key points for the upcoming session.

Focus groups with medical practitioners were scheduled twice a month. In the beginning, these sessions primarily focused on doctors' approach to patient examinations, diagnostic processes, and treatment strategies, highlighting the important need for personalized care strategies. The shared language was not immediately clear, it required about 2-3 meetings for the UX team to understand the complexities of the disease and also for domain experts to understand our collaborative approach, including data structuring, providing factual information, staying on topic, etc. The sessions were not strictly moderated to encourage a natural discussion flow. This approach was adopted after realizing that structured moderation limited the number of insights gained from doctors. Allowing the conversations to develop more freely led to more comprehensive feedback from the medical professionals.

These interactions also provided insights into the tools utilized in their practice, specifically the UR-CARE tool and the hospital's primary patient record system. It was identified that exporting data from the system is difficult, and the readability of medical reports is not appropriate. Important information for doctors, such as patient's medical history, treatment details, and screening examinations, are either inadequately presented in the current system. The system currently does not support efficient viewing of screening examination snapshots or adding notes to them. Furthermore, the meeting also highlighted the need for better organization and accessibility of laboratory parameters within the system, as well as the integration of medical reports from various sources.

5.2 Domain-specific Ideation

The domain-specific ideation phase has its goals in defining Personas, Information Architecture, User Scenarios, and User Flows in order to formalize and better understand the knowledge transferred from the domain experts from the observation phase.

Persona

The persona of Dr. Alice was derived from direct interactions with three experienced gastroenterology physicians. These interactions provided insights into the daily operational challenges and technological "pain points" experienced in the field. Unlike common persona creation, a domain-specific approach requires a deeper understanding of the researched field, including technical aspects, specialized tasks, and the specific goals and challenges faced by users within that context.

User Scenarios

Correct application of the proposed methodology in the Observation phase required identifying user scenarios that match the experiences of the doctors we spoke with. This section enumerates only the most important scenarios through motivators that have been identified, covering

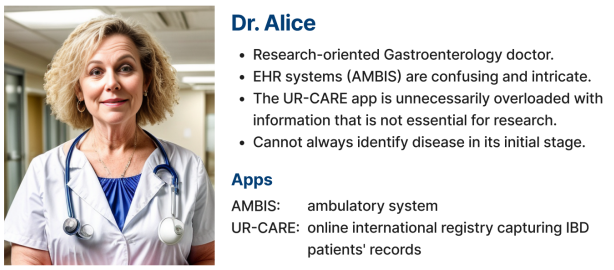


Figure 7: Persona of Dr. Alice.

all aspects of the proposed application.

1. Dr. Alice has examined the patient and needs to record the examination results in the patient's record.
2. Dr. Alice has to add new test results to a patient's record for further examinations.
3. Dr. Alice wants to review and analyze a patient's health trends for better medical assessment.
4. Dr. Alice needs to create a filtered group of patients that meets her specific criteria for a more targeted analysis in her research.
5. Dr. Alice wants to visually analyze the relationship between different health parameters (using a chart) for a specific patient group in her IBD study.
6. Dr. Alice needs to use the data for further analysis outside the IBD research system. She wants to be able to work with all relevant data from her study, including charts, patient lists, and other related information in an external statistic tool.

Information Architecture

The design of Information Architecture (IA) was inspired by the UR-CARE registry (for consistency), enhanced using insights gained from the observation phase. Our proposed system consists of 3 main sections - User management, Electronic Health Records, and the Statistics section. The core part of the IA is the IBD EHR system itself, which can be seen in Figure 8. The User management part of the IBD EHR system is not included, as it is out of the scope of our research.

Since the doctors were already familiar with the UR-CARE application, we tried to keep this structure in the new one and modify it according to our needs. The statistic part has the same structure as UR-CARE, it is divided into 2 sections: one for filtering a group of patients and another for descriptive statistics of studies. This division is based on the insights from the Observation phase, where a domain expert's initial step in new research involves defining the study group. Within the registry, the patient management section is preserved, while the method of adding patient examinations and other medical records differs in many aspects. Based on the insights from observation, we

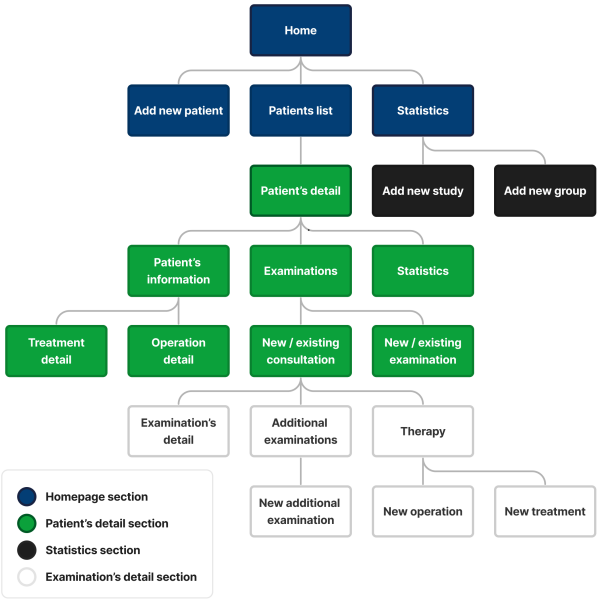


Figure 8: Information Architecture of research-oriented IBD EHR system. Colors in the legend represent different subsections of the Information Architecture.

organized the data and discovered that the entire data input procedure revolves around patient examinations. For their research, it's important to attach a timestamp to each piece of data, enabling them to efficiently filter and explore additional correlations. Consequently, the primary tier is the examination, which is then subdivided into specifics, subsequent related examinations, and the patient's therapy.

User Flows

After the scenarios were specified, the next step involved defining User Flows to ensure precision in the subsequent prototyping stage. These flows were based on already created User scenarios, requiring minimal additional specifications. Within the EHR part, beyond managing patient information, understanding the complex process of patient examinations was essential to ensure the system matches the real patient examination process. During the development of User Flows for the Statistics module, it was also important to consider findings obtained from user research conducted in the Observation phase. They indicated that doctors prefer initially to define a patient group for their studies, followed by the need to monitor specific data in the selected group. These insights were subsequently converted into User Flows.

5.3 Prototyping

As outlined in Subsection 4.2 IBD EHR System Design Proposal, the prototyping phase started with low fidelity prototypes. They were not drawn on paper, but using an online collaborative whiteboard. The primary purpose of these low fidelity prototypes was to define the layout of the

application and to illustrate the flow of information within the proposed system.

In the subsequent phase, we started creating wireframe-mockup hybrids, which was based on insights from meetings with physicians and brainstorming sessions inside the UX team. The prototype presentation revealed significant gaps in our initial designs (mainly for domain knowledge misalignments), leading to the rejection of approximately 60% of the very first design decisions. However, this process was also valuable, because of providing important insights from practitioners who highlighted the differences between our proposed designs and their real-world practice, guiding necessary refinements. Design validation sessions with doctors became more frequent, and all changes were consulted continuously. Each session systematically progressed through the defined user scenarios using visualized interactive high-fidelity prototypes for feedback gathering, which was rapidly incorporated into the next iteration of the prototype.

Through a series of meetings and iterative revisions, we developed our first workable mockups, containing mainly the application's core features. Scenarios that were successfully validated, were transferred to the IT team for development, including functional specifications, to ensure delivery of the research-oriented IBD EHR system to the domain experts in a reasonable amount of time. Meanwhile, the UX team continued to work on designing the interface for the remaining user scenarios.

6 Discussion

Considering our methodology, it becomes evident that our approach differs from the traditional double-diamond model [2, 10, 1]. From the very beginning, we were overwhelmed with plenty of information that required immediate and ongoing processing, which continued to grow and evolve throughout subsequent meetings. Moreover, our process incorporated prototyping at a much earlier stage than what is typically observed in the classical double-diamond model [2, 10, 1]. This unique approach to our project is illustrated in Figure 4, which illustrates how we navigated back and forth between different phases as the situation demanded.

To bridge the language gap between domain experts and the UX team, we established a shared language [12]. This common language allowed the UX team to act as a mediator between domain experts and the IT team, facilitating effective communication and collaboration across disciplines. An additional aspect related to this was the necessity of validating the prototype through in-person meetings rather than remotely. This approach allowed the UX team to more accurately interpret feedback to the IT development team, leveraging their understanding of the established common language.

Another interesting aspect observed during the case study is that in the traditional Double Diamond design approach [2, 10, 1], there's a large accumulation of information during the Discover phase. This information is then processed to what is essential in the Define phase. Typically, in the Development phase, there is a second, smaller peak in information volume as the design becomes more concrete. In our case, as we can see in Figure 9, the reduction and refinement of information in our case did not occur during the first two phases, but rather during the prototyping stage. It was exactly as described in Section 5 Case Study, because, at this point, the doctors were able to visually interact with our concepts of their daily workflows and processes and provide us with concrete feedback, pointing out what aspects were incorrect or beneficial.

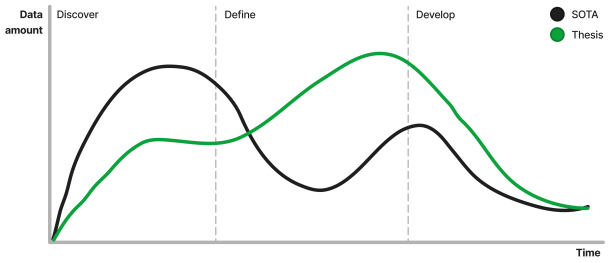


Figure 9: Figure illustrates the volume of data collected over time during prototype development. The black line represents the traditional Double Diamond approach, while the green line shows our domain expert-centered approach.[2]

7 Conclusion

In this paper, we present outcomes from a year-long collaboration with three specialists in IBD treatment research, focusing on creating a user interface for an IBD EHR system designed around their specific needs and workflows. Using a modified double diamond design approach, we emphasized the importance of incorporating domain experts throughout the design cycle. A key aspect of the development process involved showing the context of the information being communicated at any given time. Once this concept was applied to the defined design process, the development process smoothed out, resulting in improved quality of feedback. The structure of the proposed EHR system is more straightforward and user-friendly for physicians than the state-of-the-art IBD EHR system, which was also confirmed by the physician: *"Finally, managing my patient health records will become much easier and less error-prone with the system being aligned with my typical workflow. Also, I'll no longer need to manually extract the patients' data from the system and use*

[2]Graphical representation of the Double Diamond process, emphasizing the relationship between data volume and time, sourced from IDEO Tools, `https://www.ideo.org/tools`. The black line represents the State-of-the-art development of data amount (retrieved from the article) and the green line represents the author's.

various complex tools for my IBD-related research studies". The statistical section has been refined to remove unnecessary details and simplify operations, making it more suitable to the specific research practices of our domain experts.

While the system is not yet fully developed, with numerous potential enhancements remaining, one of the most beneficial additions would be transforming the system into an assistant tool for doctors. This would not only speed up their daily tasks but also assist in diagnosing and identifying disease parameter correlations. Plans also include conducting usability testing on fully functional prototypes. Additionally, we intend to extend the proposed system integrating explainability support to reveal hidden trends in the IBD EHR database without directly incorporating AI algorithms[7].

References

[1] Eleven lessons : managing design in eleven global companies. 2007.

[2] Monica Bordegoni, Marina Carulli, and Elena Spadoni. *Prototyping User eXperience in eXtended Reality*. 01 2023.

[3] L. Buley. *The User Experience Team of One: A Research and Design Survival Guide*. Rosenfeld Media, 2013.

[4] Johan Burisch, Javier P Gisbert, Britta Siegmund, Dominik Bettenworth, Sandra Bohn Thomsen, Isabelle Cleynen, Anneline Cremer, Nik John Sheng Ding, Federica Furfaro, Michail Galanopoulos, Philip Christian Grunert, Jurij Hanzel, Tamara Knezevic Ivanovski, Eduards Krustins, Nurulamin Noor, Neil O'Morain, Iago Rodríguez-Lago, Michael Scharl, Julia Tua, Mathieu Uzzan, Nuha Ali Yassin, Filip Baert, and Ebbe Langholz. Validation of the 'United Registries for Clinical Assessment and Research' [UR-CARE], a European Online Registry for Clinical Care and Research in Inflammatory Bowel Disease. *Journal of Crohn's and Colitis*, 12(5):532–537, 02 2018.

[5] Steven Dow, Blair MacIntyre, Jaemin Lee, Christopher Oezbek, Jay David Bolter, and Maribeth Gandy. Wizard of oz support throughout an iterative design process. *IEEE Pervasive Computing*, 4(4):18–26, 2005.

[6] Kenichiro Fujita, Katsumi Onisi, Tadamasa Takemura, and Tomohiro Kuroda. The improvement of the electronic health record user experience by screen design principles. *Journal of Medical Systems*, 44, 12 2019.

[7] Weina Jin, Jianyu Fan, Diane Gromala, Philippe Pasquier, and Ghassan Hamarneh. Euca: the end-user-centered explainable ai framework, 2022.

[8] Suchitra Kataria and Vinod Ravindran. Electronic health records: A critical appraisal of strengths and limitations. *The journal of the Royal College of Physicians of Edinburgh*, 50:262–268, 09 2020.

[9] Molly Kessler, Lee-Ann Breuch, Danielle Stambler, Kari Campeau, Olivia Riggins, Erin Feedema, Sarah Doornink, and Stephanie Misono. User experience in health & medicine: Building methods for patient experience design in multidisciplinary collaborations. *Journal of Technical Writing and Communication*, 51:004728162110444, 10 2021.

[10] Magda Kochanowska, Weronika Gagliardi, and with Ball. *The Double Diamond Model: In Pursuit of Simplicity and Flexibility*, pages 19–32. 01 2022.

[11] Christian Märtin, Bärbel Christine Bissinger, and Pietro Asta. Optimizing the digital customer journey—improving user experience by exploiting emotions, personas and situations for individualized user interface adaptations. *Journal of consumer behaviour*, 22(5):1050–1061, 2023.

[12] Lisa K. Meloncon. Patient experience design: Expanding usability methodologies for healthcare. *Commun. Des. Q. Rev*, 5(2):19–28, aug 2017.

[13] Ella Patterson and Emre Erturk. An inquiry into agile and innovative user experience (ux) design. 10 2015.

[14] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

A New Visualization Framework for Simulink 3D Animation

Tereza Hlavová
Supervised by: Jan Houška

Department of Computer Graphics and Interaction
Czech Technical University in Prague
Prague / Czech Republic

Abstract

This paper focuses on visualization and interaction with a 3D scene in the Simulink 3D Animation tool for MATLAB. Our goal is to improve its visual quality, physics simulation capabilities, and performance. We propose a new rendering component using Three.js, a JavaScript 3D graphics library. We describe an implementation of the rendering component and its addition to the software. Compared to its predecessor, the new renderer supports some of the new visual features of the X3D format version 4.0, mainly physically based rendering (PBR), image-based lighting (IBL), and improvements in simple collision detection. We demonstrate the improvements and changes using official examples from Simulink 3D Animation.

Keywords: MATLAB, Three.js, Simulink 3D Animation, physical simulation, physically based rendering

1 Introduction

In the last decades, 3D graphics have come a very long way with new breathtaking improvements in visuals promised and realized every year. In contrast, a MATLAB software tool Simulink 3D Animation (SL3D) has been missing an up-to-date look, not receiving similar visual improvements in years. Features we wanted to focus on include physically based rendering (PBR), image-based lighting (IBL), and casting of shadows. PBR aims to represent an interaction between light and the surface of an object more accurately than empirical local illumination methods [12]. Under such model, objects with defined materials should look consistent under any lighting setup in the scene, which was not the case for the empirical model that SL3D used. IBL produces realistic reflections and ambient lighting from images and makes the objects appear as if they belong to a given environment. Casting of shadows can help better understand locations of lights and the scene and relative locations of 3D objects.

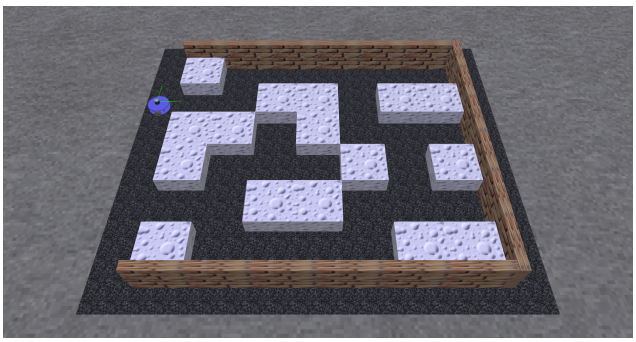
MATLAB development teams have been encouraged to transition their user interface components of Simulink and MATLAB tools using Java or other third-party technologies to web technologies. The main version of SL3D uses OpenGL 1.2.1 to render the scene and Java for its inclusion in the MATLAB graphical output displaying windows called figures. An experimental JavaScript-based branch of SL3D already existed before our work, so we analyzed its differences from the main version. The experimental version used a modified version of a JavaScript library called X3DOM for rendering 3D content [18]. X3DOM library does not support all the features that SL3D needs, mainly a *LinePickSensor* node needed for Simulink models imitating lidars, and therefore further modifications to the library were needed. Additions and modifications to the library are not trivial because it is declarative and the actual library functionality is mostly undocumented. The experimental version also performed poorly at stress tests frequently resulting at the scene not being loaded and object parameters not being updated in the scene.

We overviewed possible different directions of the development looking at three JavaScript 3D graphics libraries. We proposed a completely new implementation of its visualization component. We implemented a new renderer, interaction methods with the 3D scene and its inclusion to the rest of Simulink 3D Animation. Improvements in rendering quality are shown in Figure 1. The work was developed under HUMUSOFT s.r.o. for The MathWorks, Inc.

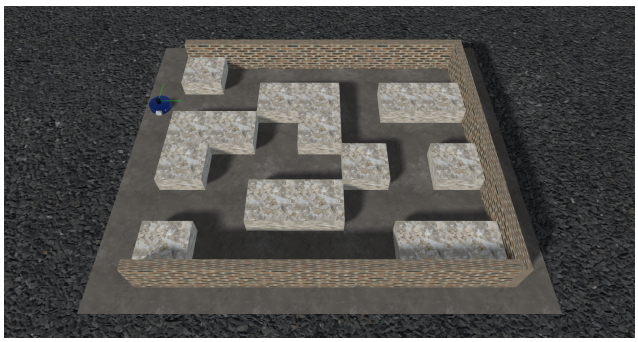
In section 2 we will describe SL3D and proposed modifications in more detail. The implementation is then overviewed in section 3 and the results are presented in section 4.

2 Background

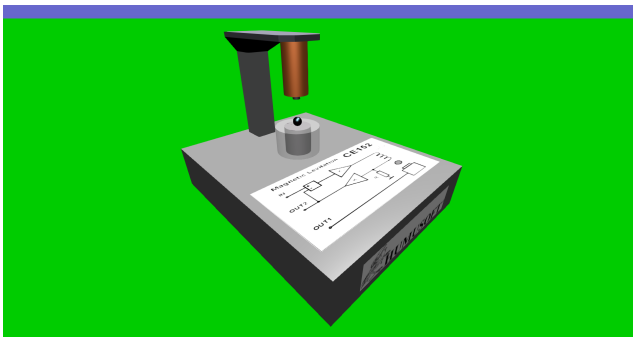
In this section, we will introduce the Simulink 3D Animation tool for MATLAB software, describe its use, implementation, and proposed goals and modifications for this work. MATLAB is a computing environment as well as a programming language developed by The MathWorks, Inc. It is widely used together with Simulink, a block diagram environment used to design systems with multidomain models, simulate before moving to hardware, and deploy without writing code [14].



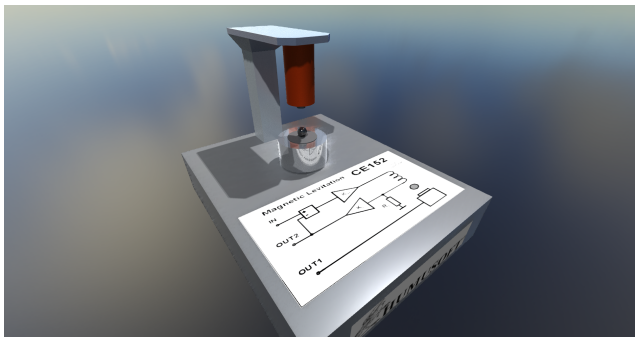
(a) *vrmaze* old example.



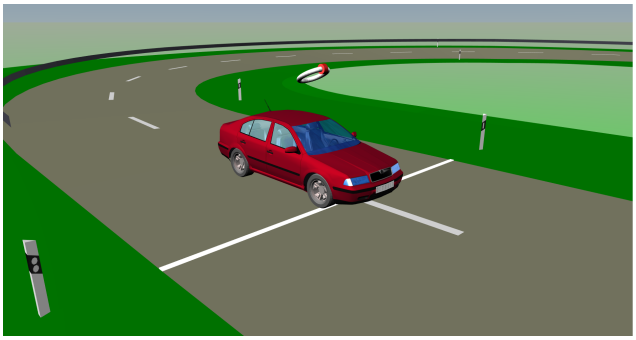
(b) *vrmaze* updated example.



(c) *vrmaglev* old example.



(d) *vrmaglev* updated example with SSR.



(e) *vr_octavia* old example.

(f) *vr_octavia* updated example.

Figure 1: Scenes from official Simulink 3D Animation examples, old lighting model versus using PBR, IBL and shadow casting.

2.1 Simulink 3D Animation

Simulink® 3D Animation™ is a tool under MATLAB software that links Simulink models and MATLAB algorithms to 3D graphics objects in virtual scenes [15]. With this tool, the user can load a 3D scene into a scene editor, modify its content, view the scene in a viewer, or connect attributes of the scene to those of a Simulink model to visualize the scene and its updates in a viewer.

SL3D in the MATLAB release version R2024a works with standardized scene formats VRML [3] and its successor Extensible 3D - X3D, version 3.3 [4]. It supports a list of features of the standard that are part of the Immersive Profile of X3D version 3.3. Both are declarative file formats describing 3D objects and scenes, as well as their behavior and user interaction. They are designed by the Web3D Consortium. The standards offer users a wide range of built-in scene node types for transformations, geometry, material definitions, and a prototyping concept used for creating new custom node types. To work with a 3D scene in Simulink, the tool offers library blocks. The blocks can write data into a 3D scene or read data from the 3D scene. This creates a connection between the fields of the scene nodes and the parameters in Simulink blocks.

The software tool can be divided into four main implementation parts:

- MATLAB interface,
- internal scene representation,
- canvases,
- editor/viewer.

The simplified software architecture outline can be seen in Figure 2. Most of the interface is implemented in MATLAB. The underlying functionality, scene storing and handling, is implemented in C++. There, the loaded 3D scenes are kept in an internal representation in classes loosely based on the OpenVRML Library [11]. C++ functions can be called from MATLAB functions with one main module called *vrclimex*. The other direction of communication is realized through callbacks. Users can either work directly with prepared functions or they are able to interact with the scene through virtual canvas classes, which maintain up-to-date modifiable scene properties.

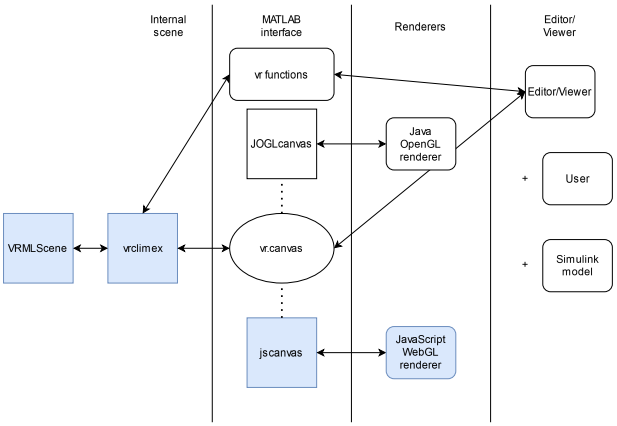


Figure 2: Simplified outline of Simulink 3D Animation architecture. The highlighted components were modified for the purposes of this work.

2.2 Proposed Modifications

The two SL3D versions take completely different approaches to the way the rendered images are produced and delivered to the user. The main branch encapsulates rendered frames directly to figures by performing traversal of the scene graph loaded into the internal scene representation. The experimental branch maintains its own separate renderer running in MATLAB's *HTML UI component*. This component internally uses a Chromium browser and displays HTML5 and JavaScript content. The *HTML UI component* runs independently from the MATLAB processing and offers a JSON-based communication channel without any synchronization guarantees. We performed stress tests on the experimental version before our implementation. In these tests, we periodically created a new canvas and checked for correctly loaded scene properties in the MATLAB canvas. Not a single instance out of 50 ensured the correct load of the chosen scene viewpoint. Testing an animation of a sphere object on a circular trajectory without any additional wait resulted in 1633 ignored animation steps out of 2000. Based on that we concluded that a communication protocol is needed to ensure the delivery of scene updates and user requests between the MATLAB interface and the renderer in the *HTML UI component*.

For the rendering component itself, we considered open-source 3D graphics libraries: the previously used X3DOM, X_ite [7], and Three.js [10]. The license of X_ite demands the source code of the software using it to be publicly available and thus is not fit for commercial use. X3DOM does not offer flexibility for the implementation of missing features or modifications in general. We decided to completely re-implement the experimental branch, adding a new renderer using the Three.js library. It offers control over scene building, animating, and rendering and thus also the flexibility that X3DOM lacks. Possible support of real-time physics simulation in SL3D could be also added this way in the future because Three.js is capable of including ammo.js [9], which is a direct port of Bullet Physics Engine [6] into JavaScript. X3DOM offers a physics simulation component [1] too, but there is no user documentation, no official examples and we were not able to produce working examples ourselves during testing.

Rendering VRML and X3D version 3.3 worlds only according to their standards would not use the wide variety of features Three.js is capable of. In order to massively enhance the visual capabilities of SL3D we suggested implementing new features also in the internal scene representation. A new X3D standard version 4.0 [5] was approved in December 2023. Its additions and changes might be crucial for this and future work. Possibly the most important addition is the inclusion of PBR through *PhysicalMaterial* node and shadow casting through *castShadow* field in a *Shape* node. We finished this work before the official finalization of the standard version when also an *EnvironmentLight* node was still a part of the standard as well before being removed for the finalized release, which included IBL.

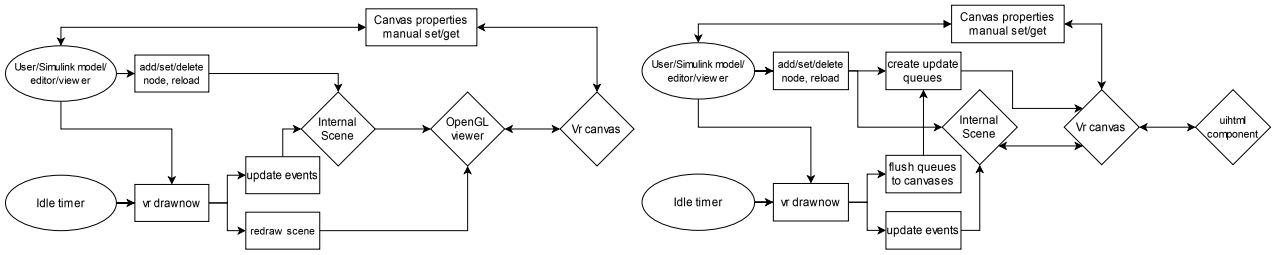
3 Implementation

This section will describe the principles of the implementation of the most important modification decisions.

3.1 Architecture Changes

After solving X3D format version control in the internal scene representation, new nodes and also new fields enhancing the existing nodes had to be marked accordingly. After these modifications, the software was capable of loading nodes and fields required for the Immersive profile of X3D 4.0 specification into the internal scene.

A message ID confirmation system was needed to ensure that no exchanged information was outdated. We implemented a new communication protocol between the renderer's code running in the *HTML UI component* and the rest of the architecture - mainly regarding updates from the internal scene but also requests from and to the MATLAB interface.



(a) Standard Java-based Simulink 3D Animation.

(b) Experimental Simulink 3D Animation, newly implemented.

Figure 3: Modification of scene and canvas properties propagation.

In the main branch, the propagation of scene modifications into the rendered frame was ensured by calling a *drawnow* function. It executes and flushes all scene modification calls. It is usually called by Simulink after every simulation step or by the editor upon user interaction. For updates in the scene event system, an internal idle timer also executes the calls on a periodic basis. This propagation of scene modification to the viewport in the main branch is visualized in Figure 3a.

For the experimental branch, we implemented a virtual canvas registration process for scene updates. The updates get stored in update queues. Upon a *drawnow* call, an update message is produced from all and sent back to all canvases registered for the scene. The scene modification propagation to the viewport in the new experimental version can be seen in Figure 3b.

The scene has to be exported for the renderer of the experimental branch because it is run separately. For messages that describe a scene or part of the scene, we used encoding into JSON format using a library called RapidJSON [2]. Scene nodes are mapped to objects, and field to object properties. The initial scene export into JSON format is built using a modified internal scene traversal which effectively prints out all information deemed important for the functionality of the external renderer. Scene modification updates created upon a *drawnow* call also use the same export functionality but are limited to the nodes they relate to.

3.2 JavaScript Renderer

As stated before, we implemented the Javascript renderer using the Three.js library. The main script starts after MATLAB calls its initialization function. It sets up the communication protocol control and constructs a rendering pipeline.

3.2.1 Scene Import and Updating

Three.js is not originally meant to work with VRML or X3D files. Its own scene representation and overall library functionality differ from the said file format standards. It was important for us to use as much already existing functionality of this powerful library to fit the standards, but even with those efforts many node types did not have matching Three.js equivalents, or at least not entirely.

The scene maintained in the renderer has to accept updates from the internally maintained one while also following the VRML mechanisms of reusing nodes. Thus we decided to use two main structures in the renderer. There is a scene graph, which is rendered by Three.js, and all its nodes are inherited from Three.js classes. Then there is a map of nodes that holds references to all instances under every node ID. Both structures are held in a scene script that is also responsible for managing scene navigation, user interaction, and rendering settings propagation to the scene nodes. It also provides a map of functions for node building and updating out of the JSON representation.

Similarly to the internal scene class inheritance hierarchy, the JavaScript renderer code defines a class for every supported scene node and the building function generates their instances. Each node always implements a constructor, a *clone* method possibly with a *copy* method, an *init* method for creation, a *set* method for non-default values during both scene import and scene updates, and a *delete* method for proper disposal of resources both locally and on the GPU. The classes either derive from native Three.js classes, enhancing them to fit X3D concepts, or were implemented anew.

The biggest difference between the X3D standard principles and the Three.js approach to the representation of the scene comes in the form of the X3D's *Shape* nodes. While the X3D standard specifies a *Shape* node capable of holding any kind of geometry node and appearance descriptions, Three.js provides different scene objects for different geometry types they are able to present. This forced us to divide some of the X3D node definitions from objects actually used in the scene and keep them both, which allowed for better control over shared resources and scene updates.

The most important classes that hold X3D definitions and manage separate Three.js objects are shape, all geometry classes, and appearance classes containing material classes and texture classes. All of them influence values in (usually multiple) Three.js objects that actually play a part in the scene rendering and manage re-referencing and reusing the resources, or their disposal when they are not

used anywhere anymore.

A source code example for a material update can be seen below. It implements base/diffuse texture addition to all required real materials in the Three.js shape objects:

```
// go through all registered x3d materials
// with base or diffuse texture
for (const x3dmat of array)
{
  const newmap = this._texture.clone();
  // go through all appearances using them
  for (const app of x3dmat._x3dappearances)
  { // apply appearance's texture transform
    app.setTextureTransform(newmap);
    const mats = app.getRealMaterials();
    // and apply it to all real materials
    // linked to that appearance's shape
    for (const mat of mats)
    {
      if (mat.map) // remove old texture
        mat.map.dispose();
      mat.map = newmap.clone();
      mat.needsUpdate = true;
    }
  }
  newmap.dispose();
}
```

The implementation of the text rendering in particular is non-trivial. This is due to the X3D standard being more flexible than the text rendering methods offered by Three.js (mainly the alignment, justification, direction, UTF-8 fonts, and material application requirements). We implemented it by rendering the text into texture from an HTML canvas element, which we deemed flexible enough. The texture is used as a mask on a plane shape which can have a material applied to it. An example can be seen in Figure 4. The downside of this approach is that the resolution for the rendered texture is pre-set. Making this dependent on the position of the camera could be a topic for future work.

The new renderer's supported nodes' mapping to Three.js classes can be seen in Table 1. Geometry gets already triangulated in the internal scene, originally for OpenGL, X3D geometry nodes missing in the table are exported for the renderer as pre-triangulated *IndexedFaceSet*. Supported sensors are also not included in the table, they do not derive from any Three.js class and are discussed in the following paragraphs.

Three.js has an inbuilt loader for models of glTF format [13]. We have also allowed the inclusion of such models in inline nodes using the loader. However, their properties cannot be modified, because the internal scene does not work with this format yet. The models are just inserted into the scene and are influenced by the transformation hierarchy.

Table 1: Overview of implementing X3D nodes using Three.js.

X3D Node	Three.js class	relationship
Networking Component		
Anchor	Group	inheritance
Inline	Group	inheritance
Grouping Component		
Group	Group	inheritance
Switch	Group	inheritance
Transform	Group	inheritance
Rendering Component		
IndexedFaceSet	BufferGeometry	inheritance
IndexedLineSet	BufferGeometry	inheritance
Shape Component		
Material	MeshPhongMaterial, LineBasicMaterial, MeshUnlitMaterial	encapsulation
PhysicalMaterial	MeshStandardMaterial, LineBasicMaterial, MeshUnlitMaterial	encapsulation
Shape	Mesh, LineSegments	encapsulation
Geometry3D Component		
Box	BoxGeometry	inheritance
Cone	BufferGeometry	inheritance
Cylinder	BufferGeometry	inheritance
Sphere	SphereGeometry	inheritance
Text Component		
Text	PlaneGeometry	inheritance
FontStyle	[none]	–
Lighting Component		
EnvironmentLight	Scene	parameter
DirectionalLight	DirectionalLight	inheritance
SpotLight	SpotLight	inheritance
PointLight	PointLight	inheritance
Texturing Component		
ImageTexture	Texture, CanvasTexture	encapsulation
TextureTransform	[none]	–
Navigation Component		
Billboard	Group	inheritance
NavigationInfo	Object3D	inheritance
Viewpoint	Object3D	inheritance
Navigation Component		
Background	Mesh/Scene	encapsulation /parameter

3.2.2 Sensor Nodes Functionality

Sensor nodes we have implemented in the new renderer so far include *ProximitySensor*, *TouchSensor*, *PlaneSensor*, *LinePickSensor*, and *PrimitivePickSensor*. When an enabled sensor is called to update on every simulation step, it evaluates its state and when needed, adds its output information to a message queue that gets sent over to MATLAB after all sensors are evaluated. The messages are then processed in the internal scene representation to be available for reading by the user or Simulink model. Only one can-

vas is chosen as the main and is responsible for producing and sending back possible sensor updates.

We have used simple collision detection and intersection computation functions provided by Three.js for the sensor activity evaluation. That includes ray-casting into the scene, or triangle/sphere and triangle/box intersections. A special case was *LinePickSensor*, which uses line geometry to detect hits with a chosen subtree of scene graph objects. The line geometry does not have to be made out of individual straight lines for every sensor but instead can be a general line geometry with many line segments. Thus we process every line segment individually during the update. The pseudocode for processing an individual segment can be seen below. *LinePickSensor* node in use is shown in Figure 5.

ALGORITHM 1
Sensor line segment processing algorithm

$startPoint \leftarrow parent$.localToWorld($startPoint$)
$endPoint \leftarrow parent$.localToWorld($endPoint$)
$length \leftarrow startPoint$.distanceTo($endPoint$)
$direction \leftarrow endPoint$.clone().sub($startPoint$)
$direction$.normalize()
$raycaster$.set($startPoint$, $direction$)
$raycaster.far \leftarrow length$
$intersects \leftarrow raycaster$.intersectObjects(...$target$)

Figure 4: Text rendering for the official example *vr_panel*.

3.2.3 User Interaction and Navigation

A viewpoint binding mechanism was implemented directly according to the X3D specification. We wanted to ensure quick and usable methods of navigation in the scene, so we decided to implement the following principles:

- The camera can orbit around a selected target in the scene.
- The camera can rotate around the center of its local coordinate system.
- The camera can zoom in and out on a selected target in the scene proportionally based on a distance to the target.
- The camera movement can be controlled by keyboard input.
- The camera is grounded under the WALK navigation type of X3D specification.

Apart from navigation and some of the sensors, user is also able to interact with the scene in edit mode. In this mode, the sensor functionality is postponed, and picking interaction is instead evaluated as node selection through ray-casting. The clicked nodes in the scene get highlighted and their fields are revealed in a world editor to modify as shown in Figure 6.

3.2.4 Rendering Pipeline

For post-processing management, we have used post-processing add-on to Three.js developed by Raoul van Rüschen [17]. Apart from the normal render pass, outline pass for highlights is added in edit mode. Subpixel morphological anti-aliasing pass can be added through canvas settings. A custom screen capture pass renders to a buffer on screenshot request. As an experimental feature, we have also added screen space reflections implementation from Realism Effects developed by 0beqz [8] in two additional passes - *velocityDepthNormal* pass and *SSREffect* pass.

4 Results

Most of the examples used for testing, comparisons, and showcase are the official examples of Simulink 3D Animation [16].

4.1 Communication Protocol

Before our work, the experimental X3DOM-based viewer did not use any message confirmation protocol and did not guarantee to offer up-to-date information on its virtual reality canvas properties. Furthermore, being independent on *drawnow* calls from the scene meant that modifications of multiple simulation steps were applied at the same time, resulting in the loss of visible changes and also an inability to reasonably measure performance.

Now, the implemented communication protocol does guarantee waiting on load events, and confirmation of messages when a renderer-dependent property value is requested. Due to the object nature of the new protocol, screen capture image data transfer from the renderer to the canvas was made possible and implemented as well. The same communication stress tests, which failed before the implementation, all passed with this new implementation.

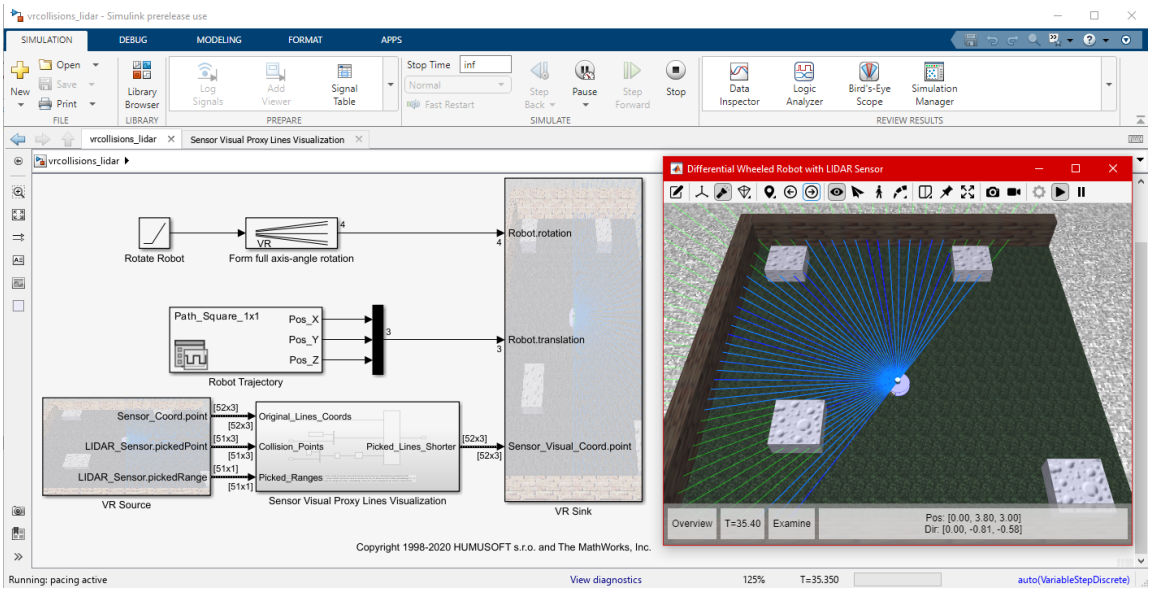


Figure 5: Simulink 3D Animation *vrcollisions_lidar* official example showing the functionality of the *LinePickSensor* node with the new renderer. There is a robot with many sensors on its body. If a sensor detects a collision with a wall, the point of collision is visualized by color of the sensor beam. The blue beam has not yet hit anything, the green part of the beam is occluded by a wall.

Figure 6: Example of highlights rendering for editing mode made with Outline rendering pass.

4.2 Supported Nodes

Current scene export and import for the experimental renderer supports most of the nodes the internal scene by itself does with the exceptions of points-related nodes, a *MovieTexture* node, a *PixelTexture* node, sound nodes, and the rest of the sensor nodes, which have not been implemented yet. Most of the scenes used by Simulink 3D Animation official examples are able to fully load with minor visual differences. Loading times of the scenes vary, the official examples taking a maximum of seconds, but scenes heavy on detail with millions of vertices do not get loaded in a reasonable time, similar to how the previous X3DOM-based viewer performed, or even the standard Java-based viewer in some cases does.

Upon modification and enhancement of the internal scene's supported node types and implementing support for given scene node representation in the JavaScript-based renderer, new visual features are now possible to use. This has allowed us to update old official Simulink 3D Animation examples, comparisons are shown in Figure 1. Performance measurements can be seen in Table 2. The example *vr_octavia* is simpler than the other two in number of animated objects and sensor activity. From the visible difference between FPS of Java-based viewer and JavaScript-based viewer in this example we can conclude that the bottleneck of the process is the implemented communication protocol and communication channel of *HTML UI component*. The Java-based viewer not limited by the communication management performs better.

Table 2: Results of tests done on the standard and the experimental version of Simulink 3D Animation. Each example was run under a Simulink profiler tool. Examples used for testing are from official software examples *vrcollisions_lidar*, *vr_octavia* and *vr_octavia_2cars*. Performance measurements are given in frames per second.

Example	**Java-based viewer**	**Three.js viewer**
vr_octavia	75	44
vr_octavia_2cars	44	44
vrcollisions_lidar	40	44

LinePickSensor node functionality needed for Simulink

3D Animation official examples of *vrmaze* and *vrcollisions_lidar* is fully implemented in the experimental version. It does not cover the full functionality specified by the X3D specification yet, but it does improve on the main version of SL3D. The intersection computation using Three.js ray-casting is more precise than that of the main version of SL3D where only intersections with bounding boxes and bounding spheres are implemented.

5 Conclusions and future work

Upon exploring the current implementation of Simulink 3D Animation, we proposed modifications to the communication protocol, supported nodes, and renderer itself. We implemented the changes, compared the software tool to its previous state, and showcased the results.

The experimental version of SL3D is now able to render the scenes of most of the official examples provided by the software. Its functionality was significantly enhanced as well as the spectrum of rendering features. Although throughout the implementation testing was done and the transition to a new renderer based on the library Three.js has been fairly successful so far, some issues might still be addressed during future development.

Missing implementation of certain node types will need to be implemented in the new renderer as well. For the purposes of using the Three.js visual capabilities to the fullest, Simulink 3D Animation will probably allow exporting of its own file format of scene description, which will be an enhanced variant of the X3D file format.

In the future, it could be interesting to integrate a full physics engine either into the renderer or even directly into the internal scene representation in MATLAB.

References

[1] Don Brutzman, Andreas Stamoulias, Athanasios G. Malamos, and Markos Zampoglou. Enhancing x3dom declarative 3d with rigid body physics support. 2014.

[2] A Tencent company and Milo Yip. *RapidJSON, Main Page.* https://rapidjson.org/.

[3] Web3D Consortium. *Information technology – Computer graphics and image processing – The Virtual Reality Modeling Language (VRML2) – Part 1: Functional specification and UTF-8 encoding.*, 1997.

[4] Web3D Consortium. *Information technology — Computer graphics, image processing and environmental data representation— Extensible 3D (X3D) — Part 1: Architecture and base components.*, 2013.

[5] Web3D Consortium. *Information technology — Computer graphics, image processing and environmental data representation— Extensible 3D (X3D) — Part 1: Architecture and base components.*, 2022.

[6] Erwin Coumans. *Bullet 2.80 Physics SDK Manual.* http://www.cs.kent.edu/∼ruttan/GameEngines/lectures/Bullet_User_Manual.

[7] CREATE3000. *X_ite X3D Browser.* https://create3000.github.io/x_ite/.

[8] github.com/0beqz. *Realism Effects for Three.js.* https://github.com/0beqz/realism-effects.

[9] github.com/kripken. *Ammo.js - Direct port of the Bullet physics engine to JavaScript using Emscripten.* https://github.com/kripken/ammo.js.

[10] github.com/mrdoob. *Three.js Official Documentation*. https://threejs.org/docs/index.html.

[11] Chris Morley and Braden McDaniel. *OpenVRML.* https://sourceforge.net/projects/openvrml/.

[12] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory To Implementation*. The MIT Press, 4 edition, 2023.

[13] The Khronos Group, Inc. *glTF™ 2.0 Specification.* https://registry.khronos.org/glTF/specs/2.0/glTF-2.0.html.

[14] The MathWorks, Inc. *MATLAB Documentation*. https://www.mathworks.com/help/matlab/index.html.

[15] The MathWorks, Inc. *Simulink 3D Animation Documentation*. https://www.mathworks.com/help/sl3d/classic-virtual-reality-world.html.

[16] The MathWorks, Inc. *Simulink 3D Animation Official Examples.* https://www.mathworks.com/help/releases/R2023a/sl3d/examples.html.

[17] Raoul van Rüschen. *Post Processing for Three.js.* https://pmndrs.github.io/postprocessing/public/docs/.

[18] X3DOM. *Official X3DOM Documentation.* https://doc.x3dom.org/gettingStarted/index.html.

Medical Imaging

Domain Expert in the Loop of Digitized Histopathology Education and Artificial Intelligence

Erika Váczlavová*
Supervised by: Miroslav Laco†

Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava, Slovakia

Abstract

In this paper, we propose a way to use a graphical user interface to present digitized multi-modal data in the field of medicine for specific domain experts. Our data consisted of digitized histopathology specimens, subject to expert examination. As the digitization of histopathology for educational purposes is only in its beginning stages, we explore how to present the data to experts in a way to encourages them to build up their confidence in digitized workflow. As part of this research, we are working on streamlining the workflow by designing assistance tools based on artificial intelligence (AI). While presenting the results of AI to specific domain experts in medicine, it is important to choose the right explainability of the results of black-box algorithms, and how to present the outputs in the user interface. We found out that the implementation of functionalities driven by artificial intelligence depends on the level of expertise of the domain expert. The differences are observed in a case study with cooperation from medical students and doctors, who got access to digitized multi-modal data with AI-powered functionalities in iteratively designed prototypes of the specialized system for education in the field of histopathology. We present outcomes from the aforementioned case study to serve as a base for the future development of specialized interfaces in the field of digitized histopathology.

Keywords: Histopathology, User Experience, Artificial Intelligence

1 Introduction

Histopathological specimens are samples obtained through biopsy or surgical procedures and subjected to histological processing. Histological processing involves the fixation of the sample, cutting it into thin sections, and staining with specific dyes that allow for microscopic tissue analysis. To digitize these glass specimens are used special scanners with whole slide imaging technology (WSI). WSI produces high-resolution digital images at multiple magnifications and focal planes. These types of images are highly suitable for educational purposes as the WSI is more interactive, it is easy to share them, and provides the opportunity to convey the same information to each student, which is not possible with glass slides, because none of them are identical. Hence, it is not surprising that WSI is increasingly being used in examinations[8].

During the examination of a slide, pathologists carefully observe and interpret the histological characteristics of the case within the context of clinical information. Through this process, they identify regions of interest, that are pertinent to the specific cases[10]. The whole process of an examination of slides and annotation is time-consuming and inefficient because areas of interest cannot be marked directly into glass slides and to determine the area of interest the specialist must go through the whole specimen in multiple zoom views.

Higher accuracy, capability, and efficiency are some of the many reasons why to transform the workflow to a digital one, through the digitization of specimens. By digitization, WSI images replace the glass slides. These WSI images are accessible by annotation tools provided in a digital platform. These tools typically provide a menu of markup shapes including measured lines, polygons, rectangles, circles, and free-form lines, which can be applied in a wide range of colors. Some systems allow text labeling of the annotation[10].

Another method to enhance the efficiency of histopathologist's work in annotating individual WSI images is by integrating AI algorithms into the process. These algorithms can automatically identify areas of interest within the images using different approaches, thereby accelerating the workflow of experts. Subsequently, experts would review the outputs of the artificial intelligence system and make adjustments as needed.

We aim to leverage the benefits of digitized image annotation processes into the teaching process at medical universities. Our endeavor involves developing a specialized tool equipped with diverse educational features, and functionalities supported by AI to the extent that its

*xvaczlavova@stuba.sk
†miroslav.laco@stuba.sk

results are presented according to the target audience.

2 Human-in-the-loop of Artificial Intelligence

Human-in-the-loop Artificial Intelligence (HITL), refers to a process addressing concerns of individuals regarding the negative impacts of the artificial intelligence revolution, such as output accuracy and interpretability. This process integrates the operation of artificial intelligence with the human factor based on domain knowledge. In the case of supervised learning, the AI can learn and make decisions based only on the supplied data, along with tags associated with the data, which we call annotations. AI's decisions are based on statistics and connections, abstracted at both lower and higher levels from the supplied data. But these decisions do not contain domain knowledge, which often does not appear in the data [4]. While training models of AI, human input is often important, which corrects the results and thereby helps improve algorithms. HITL also addresses ethical questions about ownership of knowledge on which artificial intelligence models are trained since the models they learn from data created by ordinary workers [13].

3 Artificial Intelligence and User Experience

Artificial Intelligence (AI) holds a pivotal role in improving human-computer interaction and optimizing user experience. However, the design and innovation of such interactions pose multifaceted challenges. AI's potential for introducing unforeseen errors can adversely impact both reputation and user experience in collaborative settings. Designing cooperation between humans and AI is particularly demanding [12].

In iterative prototyping and testing of user experience without the use of AI, it is possible to address and test further iterations of shortcomings. However, when prototyping and testing with AI features, this becomes challenging as the AI may introduce unforeseen errors. Another challenge for designers is setting user expectations regarding what can be expected from the AI. Since the AI lacks legal and ethical awareness, there is concern over incorrect outputs potentially causing frustration. Additionally, for user experience professionals, collaborating with artificial intelligence experts can be challenging due to the distinct domains involved. Moreover, by prioritizing explainability in AI, users can develop a deeper trust in the system, as it allows them to comprehend the inner workings and decision-making processes, thereby ensuring an optimal balance of complexity in presented results. [12, 5].

Various methods exist for presenting AI model output data. Designers must consider scenarios like true positives, false positives, true negatives, and false negatives. These are addressed in two result generation approaches. One prioritizes output precision, aiming for accuracy even if the output set is smaller, potentially overlooking some true positives. The other approach, called recall, aims for a broader set of outputs to maximize the presence of true positives, even if not all results are relevant or correct.[2].

4 State of the Art in Education Process

We focused on analyzing various educational and annotation tools for digitized multi-modal data that can be used in both teaching and practice in medical universities. In these tools, we look at functionalities that are useful in the study of pathology, as well as in the analysis of medical image data. In the realm of medical imaging and education, several tools have emerged, each with its own set of advantages and limitations.

QuPath stands as a platform for the analysis of medical image data. Its ability to handle diverse formats and provide a range of marking tools empowers users to annotate and manipulate areas of interest directly onto digital specimens. However, the absence of a comment feature and a somewhat complex user interface may pose challenges, particularly for those with limited computer literacy. In contrast, **AMBOSS** represents a commercially driven approach, offering a repository of educational materials in a sleek, user-friendly interface. Its virtual library and note-taking functionalities enhance the learning experience, allowing users to create and share annotations with ease. Nonetheless, its closed nature restricts the ability to modify or expand the content of medical knowledge for people in medical field study. Meanwhile, **The Human Protein Atlas** serves as a valuable supplementary resource, providing a wealth of high-resolution images showcasing protein distribution across various human tissues and cell lines. While its predefined pathways and detailed descriptions offer structured learning experiences, the inability to insert custom images or annotate specimens may restrict its utility for interactive study. In essence, each tool brings unique strengths to the table. However, navigating their respective limitations is crucial in harnessing their full potential for medical education and research in the modern era.

5 Our Approach

Recent studies recommend that for working with data in the field of medicine and health, it is necessary to develop new usability methods and theories on how to work with them [6]. Based on these recommendations, various new procedures began to emerge as to how to

perceive the user during the design of the system and also that this user needs to be specified more closely according to the domain area in which the design is being created. One viewpoint entails the adaptation of the conventional user-centered design principle, particularly within the medical domain, where it has been redefined as patient-centered design[6]. Based on the state-of-the-art in patient-centered design, we decided to modify this principle and work on histopathology-expert-centered design and medic-centered design.

5.1 Design Centered on the Domain Expert in the Field of Medicine

Before creating a functional system design, it is essential to consider all stakeholders involved in the creation process, ensuring that the trust of the domain expert, for whom the system is designed, is gradually established. All of these stakeholders are visualized in Fig.1. Additionally, the system should be designed in such a way that the domain expert can naturally utilize all its functionalities and extensions without hesitation after its creation.

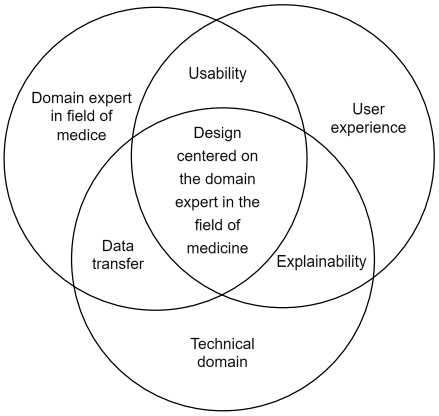


Figure 1: Visualization of 3 components and their cooperation in design centered on the domain expert in the field of medicine methodology. Each area overlay represents an area to focus on. Adapted from Meloncon et al. [6]

During the system development process, the role of this principle is to ensure that the domain expert generates data required for the technical aspects of the system, while the system provides data to the domain expert in an understandable format. The presentation format of the data is examined by a user experience expert, who explores how to create a reliable and usable system. Findings are obtained through interaction with the domain expert and translated into technical language for the development.
To create a design focused on the domain expert in medicine, it is important to build the entire collaboration thoroughly and approach the design as a continuously evolving relationship between stakeholders. To be able to design the cooperation and the system to the satisfaction of the domain expert, it is important to get the collaboration right from the initial stages steered properly.

5.2 Annotation Enhanced Educational Tool

As a second important contribution of this paper, we have designed an annotation tool that will be part of a comprehensive educational system in cooperation with domain experts from the medical university. The proposed prototype focused on functionalities related to annotating digitized histopathological specimens. In this prototype, individual images can be viewed and annotated using various tools. These tools are divided into those not supported by AI and those simulating real results of AI models.
The design was based on user needs of real users, which we generalized into 2 personas. These personas served to better understand the mental model of end users. Among these personas is an expert who teaches histology and pathology at the university and practices in the clinical sphere, aims to teach modern methods at the university, and provides feedback to students on their work while also sharing extra materials. The second persona is a student who seeks hands-on experience in annotation and desires access to materials even after classes to further educate themselves in the field of diagnosis determination.

5.3 Presentation of the Artificial Intelligence Outputs in Histopathology

While designing the user interface for the annotation tool, our focus was on deliberating upon the most suitable presentation of artificial intelligence. Two principal approaches were considered: automation characterized by AI-driven task execution devoid of human intervention, and augmentation which entailed AI providing recommendations to users of the annotation tool, who subsequently validated or dismissed its outputs within the context of our work[9].

We proposed three functionalities aimed at simulating AI results in various forms. Automation was represented by a tool that upon triggering the workflow, automatically highlighted all areas of interest on the annotated image. Augmentation was depicted through two tools: one gradually revealed areas of interest in the annotated image, requiring user confirmation or rejection with each annotation. The second tool offered the option of displaying hints, outlining regions on the image where areas of interest could potentially be found, without showing the actual annotation. Our contribution includes comparing the usability and the explainability of AI outputs using user experience methods such as usability testing and contextual interviews.

5.4 Reward Function and Explainability

A reward function is a component of reinforcement learning of AI algorithms. It defines the objective or goal that the AI is trying to maximize or minimize in order to receive rewards or punishments, guiding the AI's behavior toward achieving desired outcomes. This approach may be considered from the UX point of view when working with all NNtypes, not only reinforcement-learning-based NNs.

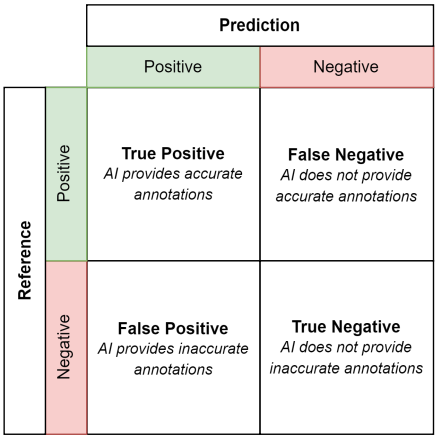


Figure 2: Reward function for AI used in annotation tool.

In Figure 2, errors arise in two cases. False Positives occur when AI provides inaccurate annotations, especially troubling in fields like education or medicine due to potential user impact. False Negatives happen when AI fails to provide accurate annotations, leading to increased manual work for users and posing challenges in maintaining focus during correction. We consider this as a concern for applications in the field of medicine where it is crucial for the user to receive accurate outputs. Therefore, our proposal offers the user more benefits in minimizing False Positives (where AI generates inaccurate annotations), hence it is appropriate to optimize the function for precision. We acknowledge that opting for this method entails a compromise, meaning our model will lean more towards abstaining from creating any annotation rather than producing an inaccurate one. We prioritize refraining from displaying any annotation to the user over presenting an inaccurate one. We propose to verify this approach and evaluate our proposal using pre-generated annotations as AI's results against manually created annotations.

6 Approach Validation and Testing

Based on contextual inquiry and observation of domain experts we prepared four user scenarios, which were, according to the good practice in the field of UX and usability testing, tested by five participants (more in [7]). These participants corresponded to the personas created in earlier stages of the project. The group of participants consisted of four students with low expertise and one experienced expert from the medical field. Despite varying IT skills, all possess sufficient knowledge to effectively annotate the data.
The usability testing was focused on evaluating the necessity and form of implementing AI. We measured quantitative outcomes such as task completion rate, error rate, and time needed for tasks. We also collected qualitative data from user feedback and suggestions. During this testing phase, we also examined the explainability of artificial intelligence outcomes and how to present them to end users, as represented by the testers. The entire testing process was conducted using the thinking-aloud method [1].

We defined four tasks. The tasks were created in cooperation with the domain expert. One of them was designed for the user not to utilize tools supported by AI. The remaining tasks simulated various ways of utilizing the outcomes of AI. The outcomes from AI were simulated by pre-created annotations, created by domain experts, and served to participants using the Wizard of Oz methodology [3, 11]. All tasks were about annotation in real digitized specimen. The specimen was from cardiac tissue and contains the endocardium. All tasks were based on the same annotations on the same data. The tasks were:

1. Please annotate the endocardium in the image using the drawing function - This task was designed because its results will serve as a baseline for evaluation.

2. Please annotate the endocardium in the image using the Annotation proposals function - This task was designed for observing the user behavior and the impact of augmentation on performance in simple tasks.

3. Please annotate the endocardium in the image using the Automated annotation function - This task was designed for observing the user behavior and the impact of automation on performance in simple tasks.

4. Please annotate the endocardium in the image using the Hints function - This task was designed for observing the user behavior and the impact of augmentation on performance in simple tasks.

7 Results

During the testing, we monitored various metrics, and after evaluation, we divided the results into quantitative and qualitative outcomes. Each relevant feedback obtained during testing helped us understand how to implement AI-supported features properly.

7.1 Quantitative Results

The typical procedure involves observing task completion success. In this case, participants were able to complete all tasks, with the assistance of a facilitator required only once. Task assignments were straightforward, and the prototype was designed to be trivial to users, as the end user is not proficient in information technology.

Task completion time

Participant (Expertise)	Manual annotation (Task 1)	Annotation proposal (Task 2)	Automated annotations (Task 3)	Hints (Task 4)
P1 (low)	145	98	70	**52**
P2 (low)	150	80	60	**40**
P3 (low)	180	96	**60**	69
P4(high)	185	75	50	**39**
P5 (low)	180	110	57	**42**
Average	168	91.8	59.4	**48.4**
Std	16.91	12.71	6.43	11.28

Table 1: Task completion time in seconds "Std" - Standard deviation

Time needed for one annotation

Participant (Expertise)	Manual annotation (Task1)	Annotation proposal (Task 2)	Automated annotations (Task 3)
P1 (low)	7	13	**6**
P2 (low)	8	7	**5**
P3 (low)	10	7	**5**
P4 (high)	12	**7**	10
P5(low)	11	10	**4**
Average	9.6	8.8	**6**
Std	2.07	2.68	2.10

Table 2: Time needed for one annotation in seconds;

In Table 1, we can observe the trend in user performance evolution within the proposed tool with the assistance of AI-generated results. As evidenced, tasks utilizing artificial intelligence were completed faster. These values must also consider a slight bias introduced by participants gaining experience with the tool and gradually acclimating to its use with each task. However, this bias is not high enough to preclude the assertion that the application of artificial intelligence in any form has increased work efficiency.

In Table 2, we can compare the time required for creating a single annotation in a digitized image. These results may be influenced by biases stemming from prototype limitations. However, this bias is not significant and applies to each annotation, so it does not need to be taken into consideration. In Table 2, it is evident that creating annotations without tools incorporating AI assistance takes longer than creating annotations with their use. Based on the numerical values, it is therefore most suitable to implement AI in the form of automation to reduce the time required for each annotation.

7.2 Qualitative Results

We conducted a qualitative evaluation based on participant observations during testing, and analysis of video recordings obtained during testing with the participant's consent to anonymously participate in the research, as well as the facilitator's questions or questionnaire inquiries.

Technical design of the prototype

All inquiries regarding the simplicity of application usage were responded to by participants with a positive sentiment. Considering that the prototype was designed so that participants meeting the parameters of our personas had no issues with its utilization, we deem it a suitable environment for testing the prototype with functions working with AI's results.

Automation

In the prototype, we represented automation through the functionality of displaying automatic annotations being simulated outputs of the AI for the given whole slide image instantly with the image itself as an overlay. All participants appreciated having a large number of annotations quickly using this approach. Regarding the facilitator's question about whether this functionality could pose any negative impact on their work or study, responses varied depending on the level of expertise.

Participants with lower levels of expertise stated that they appreciated such functionality as it speeds up their review of individual images during study or in their potential future work.

Participants with higher levels of expertise exhibit more skepticism towards this functionality. When using it, they are concerned that the system may offer incorrect annotations which they may not have time to verify and could potentially lead to errors. They also emphasize the importance, particularly in teaching contexts, of reviewing images to determine whether the area annotated is correct, which may not occur when a large number of annotations are displayed.

Augmentation

In the prototype, we represented augmentation of the manual annotation process with the AI outputs relevant to the given whole slide image through two different functionalities. One of them was Hints which displayed regions in the image where an area of interest might be located, prompting the creation of an annotation. Regardless of expertise, all participants appreciated this

functionality and claimed they would commonly use it.
Participants with lower levels of expertise would utilize this functionality during study sessions, where it would assist them in orienting themselves in the image and guiding them to create their own annotations.
Domain experts with higher levels of expertise liked this tool and claimed it would facilitate their manual and laborious examination of specimens. They also appreciated its potential for use in the educational process.

The second functionality representing augmentation is the Annotation proposal, where annotations gradually appear on the image, allowing the user to confirm or remove them with a click. After approval, they can continue to edit them. This functionality was perceived by all participants as faster than manual annotation but slower than automation. Regardless of expertise level, this functionality was perceived most positively, as domain experts felt they had control over individual annotations.

Augmentation or Automation
When asked which annotation functionality they would prefer, domain experts with lower levels of expertise agreed that automation seemed practical and fast. Conversely, domain experts with higher levels of expertise recommend augmentation, both in clinical practice and in teaching and study.

The level of trust in AI results
When asked whether they trust the system that recommends annotations, participants expressed skepticism. None of them confirmed that they would fully trust the system. However, this fact is positive because they would all verify the majority of annotations, thus reducing the risk of error.
The level of expertise of the domain expert, in this case, the participant, also influences their trust in the system. Participants with lower levels of expertise stated that if they had more knowledge in the respective field, they might be able to trust the system more. They also expressed that they would trust the system if they knew it didn't make errors frequently.
Participants with higher levels of expertise state that it's not possible to trust the tool 100 percent, but that's also true for humans. However, an individual who is still learning about the subject must have input on which to build. Such utilization of artificial intelligence would be credible only if an expert intervenes in the learning process to correct any misinformation provided to students. The use of such a system in clinical practice and trust in it would likely require time to build. The longer the tool is used, the more an expert would know which potential errors to focus on.
One of them stated:
"Even though I have the opportunity to intervene, the human mind tends to seek simpler paths. So, in that case, I wouldn't have trust in the system because ultimately I no longer trust myself. Comparing what I know with the information provided by the system can lead to a situation where two pieces of information confront each other. And now it's about which of those personalities is more confident to say 'but this is how it is,' even though it hasn't looked at, for example, 80,000 slides like artificial intelligence has. Because it will have a greater opportunity to feed its head than the human."

Explainability
During testing, we also asked our participants how their trust in the system could be supported. Their trust could be enhanced through explainability features, which would justify the individual outcomes of artificial intelligence. Explainability could assist domain experts in understanding highlighted areas and provide additional information necessary to confirm or decline AI results.
From the interviews, we learned that domain experts would prefer explainability in written form. This explainability should clarify the reasons why the annotation was created using medical terminology. Test participants did not prefer explainability in the form of percentages indicating AI's confidence in the annotation, nor did they favor heat-maps or other numerical ratings. Similarly, they did not prefer explainability in the form of a similar case shown in a tooltip.

8 Discussion

In the realm of the user experience in digitized histopathology, it is imperative to meticulously consider and accommodate the varying levels of expertise among domain-specific experts. This entails a conscientious approach to integrating the insights and contributions of experts from diverse domains, ensuring that each individual's specialized knowledge and proficiency are fully present the AI results in a proper way and leveraged to adapt the processes and outcomes within the digital histopathology framework.
Prioritizing expertise levels among domain experts in designing the frameworks for digital histopathology is fundamental for driving innovation and enhancing medicine study field improvements.

Assuming that the proposed tool will be utilized by experts with a lower level of expertise and also by experts with a high level of expertise, it is essential to design it in a manner that caters to the specific needs of both groups.
The limitation of the usability study was introduced by participants gaining experience with the tool and gradually acclimating to its use with each task. However, we claim this bias does not contradict the basal finding that the application of an AI-assisted approach in any form increases work efficiency when introduced in the tool after the user gets familiar with the manual annotation workflow.

8.1 Phases of Design Focused on the Domain Expert in the Field of Medicine

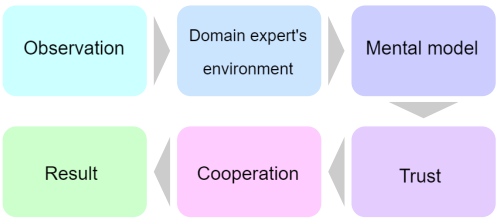


Figure 3: Visualization of phases of design focused on the domain Expert in the field of medicine.

Before creating a design focused on a domain expert in the field of medicine, it is important to build the entire collaboration thoroughly and approach the design as a continuously evolving relationship between stakeholders. To be able to design in a way that achieves satisfaction for the domain expert, it is important to grasp the collaboration correctly from the initial phases. All detailed steps of collaboration is depicted in Fig.3.

The first half of the collaboration ends with the creation and understanding of the mental model of the domain expert in the field of medicine. Only at this stage can we create specific personas and design to reference real user needs. This phase is preceded by observing domain experts. Observing domain experts also includes contextual interviews, obtaining a concrete picture of their needs. When it comes to a specific domain, such as in this case the domain of medicine, it is important to conduct observations and meetings with domain experts in their own working environment. Observing the home environment helps us understand the typical flow of activities, and the domain expert appears more confident in their familiar environment, with their behavioral model not being distorted by various external factors.

The second half of the collaboration begins with the most important phase, which is building trust. Trust from the domain expert in the field of medicine towards the technical domain expert is crucial due to significant differences in focuses. Trust needs to be built gradually through dialogue and openness. With such an approach, the domain expert gains confidence and begins to collaborate with technical domain experts as colleagues without the need to distinguish or underestimate either side. After gaining trust, it is necessary to reinforce it by involving the domain expert in the development process, making it clear that their opinion is important even in the technical domain. The involvement of the domain expert can take various forms, from the initial stage of prototype development during sketching, through testing or data creation, to feedback.

The final phase of the collaboration is the outcome. The outcome consists of multiple goals from each party involved in the collaboration. The outcome includes the developed prototype, product, and system, as well as the satisfaction of the domain expert in the field of medicine. The final phase may define various outcomes in different cases, but it is important for these outcomes to meet the goals and bring benefits to both domain experts in the field of medicine and the technical domain. Among the outcomes, we also include associated partial goals such as gained trust or expanded knowledge in the domain.

8.2 Application of Automation and Augmentation Based on the Level of Expertise in the Field of Medicine

It is crucial to focus on the implementation of intelligent features into tools used for educating domain experts in the field of medicine to streamline the work of medical professionals and generate a wealth of valuable study materials that will be more readily available to all students compared to current educational methods. Through collaboration, testing, and observation, we have found that how artificial intelligence is implemented into applications in the field of medicine should be on the expertise level of individual domain experts who will be using the proposed tools.

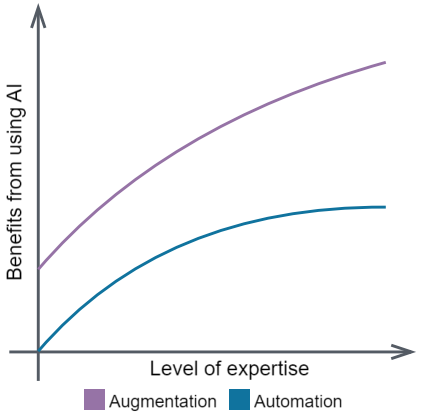


Figure 4: Visualization of the relationship between the benefits of using AI and levels of expertise in 2 types of AI implementation

There is a relationship between the expertise level of the domain expert and the number of benefits that can be derived from using AI-supported functionalities. As we can observe, the higher the expertise level, the greater the benefits provided by such functionalities. The reason is that experts with higher levels of expertise can critically evaluate the results of AI, whereas, without domain knowledge, there could be negative influences on the outcomes of AI.

As visualized in Fig.4, there are some differences between the methods of implementing AI and the benefits these methods yield to domain experts with varying levels

of expertise.
The first method of presenting AI outputs in the user interface, depicted in blue in the figure, is automation. This curve commences at the origin of the coordinate system, signifying that in the absence of domain knowledge, automation yields no discernible benefits either in the learning process or in clinical practice. This is because the user attempting to educate themselves through the application loses a crucial part of the learning process, namely analysis. The magnitude of benefits conferred by automation increases gradually with the accumulation of domain knowledge.

The second curve depicted in Fig.4, represented in purple, has its benefit value with near-zero domain knowledge obviously higher than the automation approach. As evident, this curve does start at a higher point, indicating that this method of AI implementation is suitable even in the educational process, where it can provide recommendations and help experts with lower levels of expertise navigate through digitized preparations. With increasing levels of expertise, the number of benefits that augmentation can bring also increases. In this method of AI implementation, domain experts have significant control over the AI results, which gives them a greater sense of comfort and increases trust.

9 Conclusion

In this paper, we targeted the identified research problem of domain experts in the loop of digitized histopathology education and artificial intelligence. We addressed these open research questions with our approach proposal including histopathology-expert-centered design and medic-centered design. We validated and examined the proposed approach in a case study in cooperation with domain experts from a medical university. Our main contribution is the phases of design focused on the domain expert in the field of medicine and the proposal of application of automation and augmentation based on the level of expertise in the field of medicine.
In our future work, we plan for medical faculty students to adopt the annotation tool in their education process as an extensive usability study while evaluating their interactions. The study on interactions will supplement the study on Design Focused on the Domain Expert in the field of medicine.

References

[1] Ted Boren and Judith Ramey. Thinking aloud: Reconciling theory and practice. *IEEE transactions on professional communication*, 43(3):261–278, 2000.

[2] Michael Buckland and Fredric Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994.

[3] Steven Dow, Blair MacIntyre, Jaemin Lee, Christopher Oezbek, Jay David Bolter, and Maribeth Gandy. Wizard of oz support throughout an iterative design process. *IEEE Pervasive Computing*, 4(4):18–26, 2005.

[4] Fabrice Jotterand and Clara Bosco. Keeping the "human in the loop" in the age of artificial intelligence: accompanying commentary for "correcting the brain?" by rainey and erden. *Science and Engineering Ethics*, 26:2455–2460, 2020.

[5] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

[6] Lisa K Meloncon. Patient experience design: Expanding usability methodologies for healthcare. *Communication Design Quarterly Review*, 5(2):19–28, 2017.

[7] Jakob Nielsen and Thomas K Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 206–213, 1993.

[8] Liron Pantanowitz. Digital images and the future of digital pathology. *Journal of pathology informatics*, 1, 2010.

[9] Sebastian Raisch and Sebastian Krakowski. Artificial intelligence and management: The automation–augmentation paradox. *Academy of management review*, 46(1):192–210, 2021.

[10] Chetan L Srinidhi, Seung Wook Kim, Fu-Der Chen, and Anne L Martel. Self-supervised driven consistency training for annotation efficient histopathology image analysis. *Medical Image Analysis*, 75:102256, 2022.

[11] Leandro Manuel Reis Velloso and Gil Barros. Recurrent techniques used in ux design: a report from user survey and interviews with professional designers. *Journal of Design Research*, 21(1):47–61, 2023.

[12] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. Re-examining whether, why, and how human-ai interaction is uniquely difficult to design. In *Proceedings of the 2020 chi conference on human factors in computing systems*, pages 1–13, 2020.

[13] Fabio Massimo Zanzotto. Human-in-the-loop artificial intelligence. *Journal of Artificial Intelligence Research*, 64:243–252, 2019.

Deep Learning-Based Segmentation and Classification of Histological Colon Cells

Patrik Kozlík*
Supervised by: prof. Ing. Vanda Benešová, PhD.†

Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava / Slovakia

Abstract

Deep neural networks have become important in the research of medical applications, particularly in histology. One of the important research areas is the usage of deep neural networks in the diagnostics of diseases such as Crohn's disease and Ulcerative Colitis. In these types of diseases, the correct detection of specific cell types and cell features is crucial. To integrate this domain-specific knowledge directly from pathologists, we propose a customizable system made of three connected modules: segmentation, filtration, and classification.

The segmentation module uses the AttentionUNET [11] architecture to find cell boundaries. The filtration module contains a stack of filters suitable to specific cell features, such as color, shape, and area. These filters are used to eliminate irrelevant cells from the predicted segmentation mask. The classification module uses the ResNet34 [4] architecture for multi-class classification tasks. Through experimentation involving custom loss functions and attention modules, we found that the filtration module is well-suited for elimination of irrelevant cells from the predicted mask. The segmentation module achieves a Dice score of 84.18% and an F1-score of 90.08%. However, the classification module exhibits an accuracy of approximately 72%, primarily due to limited annotated data. Nonetheless, our solution proves effective in scenarios with constrained training data, as the filtration module aids with process of prediction by filtering out irrelevant cells.

Keywords: Deep Learning, Cell Segmentation, Cell Classification, Computer Vision, Medical Image Analysis, Feature Extraction, Histopathology

1 Introduction

In recent years deep learning started to be involved progressively more in histology. *Histology images* contain a vast number of features and objects, which need to be precisely detected by doctors. An example of a histology image of the colon can be seen in fig. 1. These images are in a different resolutions in a range from 61440x73728 to 134400x82944 pixels. Because of these resolutions it is sometimes hard to detect abnormalities in them. Because of this, *deep neural networks (DNN)* came in handy. With the help of DNN, mostly convolution neural networks [10], pathologists can save days of work. Specific types of diseases, where deep neural networks can be used is *Ulcerative Colitis* and *Crohn's disease*. The symptom presented in these diseases is inflammation of the colon and small intestine. Starting and long-lasting inflammation can be detected in histology images. In the last 50 years occurrence of these diseases increased 10 to 15 times. This means it became a modern-age disease, which occurs more and more often [9]. Indeed, the complexity of diagnosing diseases in the colon arises from the diverse range of cell types and tissues present in this region. Unlike some other diseases that may affect more homogeneous tissues, conditions affecting the colon often involve interactions between various cell types and tissues, each with its own distinct characteristics and functions. Diagnostic and prediction of disease is extremely important. When a doctor can predict the development of a disease, he can establish less invasive treatment earlier. This can in some situations save a patient's life.

The training of neural network-based system requires a sufficient amount of annotated cell images. However, annotating these cells is time-consuming and can only be done by pathologists. Since creating annotations is not a common task for pathologists, it is crucial to expedite the annotation process. Only pathologists know the exact rules regarding which cells are most important and which features to focus on. This underscores the importance for pathologists to have the option to interact with and modify the annotation tool, as well as access an annotation tool capable of extracting cells based on specific features. Hence, they require an annotation tool capable of minor modifications by individuals who aren't AI professionals.

The contribution of this papers is to provide a system, which is able to perform segmentation and classification of specific areas of histology images chosen by pathologists, with the additional option of modifications of the system

*xkozlik@stuba.sk
†vanda_benesova@stuba.sk

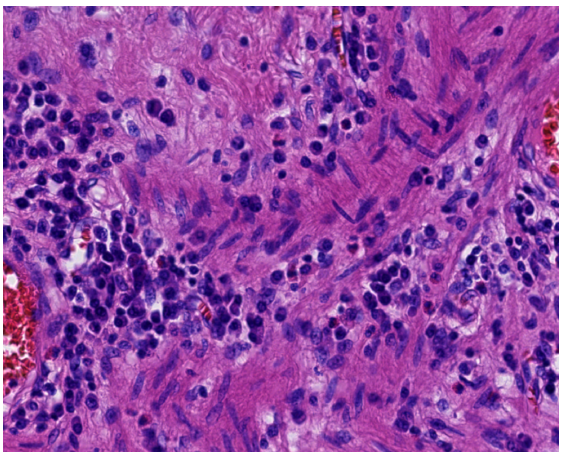


Figure 1: Example of histology image

by the user. Due to the scarcity of training data, the model is pre-trained on the CoNIC dataset [2]. The whole system is meant to be implemented into the custom annotation application as a tool for the acceleration of the pathologist's work. The advantage of this custom annotation application is that it would be easily expandable based on specific requirements.

Mentioned system is made of three connected modules: segmentation, filtration, and classification. These modules work as a pipeline. The main part is the filtration module, which provides necessary modifiability. Implementation of the module into the custom annotation application can increase the speed of histology image examination by a pathologist. Furthermore, it offers the option to implement active learning based on the visualization of predicted masks within the annotation application.

2 Related work

Numerous applications have been developed for cell segmentation and classification [17], yet only a limited number specifically target Crohn's disease and Ulcerative Colitis. That is the reason why, it is hard to compare this specific task to other state-of-the-art solutions. Most of the similar solutions focus on the incorporation of additional features with *attention mechanisms* [14] or adjustments to neural network architectures. This is a advantageous option when we want a precisely functioning black box that requires infrequent modification. However, it also increases the system's complexity, making optimization and management more challenging and model less explainable.

Yildiz et. al [18] work with the same CoNIC dataset [2] as used in this paper. However, the paper provides a whole different approach to cell segmentation and classification. The *UNET* architecture mentioned in the paper is used for multi-class segmentation which means they used one model to predict and classify segmentation mask. Validation was performed on five different subsets of images. The average metrics of these tests are *57,08% dice score* and *48,57% IoU*. At first glance, this approach may appear to offer a superior and simpler solution. However, a issue may arise. One model needs to learn how to perform segmentation and also classify individual pixels into correct classes. Handling complex data such as cells can indeed present challenges in this regard. The problem also can be seen in the paper, where the test accuracy metric showed that some of model is not that robust. This is beneficial in situations, when model needs to have fast performance and take up less disk space. Due to the previously mentioned challenges associated with the complexity of performing classification and segmentation on similar types of data such as cells, we opted to use separate models for segmentation and classification.

Iacucci et. al [8] provide neural network architecture adjusted to distinguish between remission and inflammation phases. This can be predicted from histology images based on specific types of cells located in tissue. The method which is used for this problem is a modification of *VGG16 architecture* by use of attention mechanism [8], which results in better focus on *neutrophils*. Input to the attention mechanism is the annotation of the corresponding whole slide image. This modified architecture provides *79% accuracy* and *71% F1-score* for classification. The implementation offers techniques for prioritizing specific types of cells over others.

Zeng et. al [19] provide different sights on the problem and different approaches. In this proposed solution architecture detects the position, shape, scale, and contour of cells. The architecture used in the paper is called *Residual-inception-channel attention-Unet (RIC-Unet)* [19]. It brings the advantage of residual blocks, which helps to extract more representative features. The Inception block [19], also present in this architecture, is renowned for its computational efficiency and its ability to manage large receptive fields. On top of this, an attention mechanism is added for better focus on regions of interest. The combination of these blocks creates the architecture of RIC-Unet. This architecture provides an *80,08% dice score* and *82,78% F1-score*. The biggest advantage of this architecture is its ability to extract features of single cells.

Aziz et. al [3] used different method for feature extraction. In this proposed solution is chosen five different neural network based architectures for feature extraction, following algorithms for feature selection and classification itself. Specific architectures used for feature extraction are VGG16, VGG19, Xception, ResNET50 and ResNET121. Proposed method outperforms state of the art solutions by 95,5% accuracy for multi-class classification and 99,49% accuracy for binary classification. It shows exceptional results, but one problem still persists. The system is challenging to interpret, making it difficult to ascertain which specific features are the most relevant and whether they align with the primary decisive features identified by pathologists.

The mentioned papers [8, 18, 19] are based on attention

mechanism [6]. However, these mechanisms can be modified only by AI specialists, due to their complexity. This could be beneficial in some situations, but when a model needs to be set up for each task specifically, it becomes crucial for the model to be *user-adjustable*.

3 Proposed method

The whole concept of developing an AI system for medicine, which we want to follow can be seen in fig. 2. The starting point is often a pathologist who presents the problem to a *UX expert*. This UX expert later introduces the problem to the AI expert in a more technical manner. Afterward, AI experts try to implement solutions and discuss results with pathologists through UX experts. This process is repeated a few times, creating a loop [13]. In our specific problem, pathologists need architecture, which can accurately classify selected classes of cells in colon histology images. These types of cells are often common in Crohn's disease and Ulcerative Colitis. To take advantage of communication with pathologists, we aim to create an architecture, which will be easy to modify. These modifications will enable the integration of additional information provided by pathologists directly into the architecture. The blue part of the diagram in fig. 2 represents the AI system of our proposed solution, which is described in the next chapter.

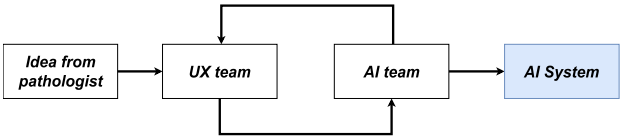


Figure 2: Human in the loop cycle for developing of new AI histology tool

3.1 System architecture

The whole system, which can be seen in fig. 3, can be divided into three main modules. Each of these modules is responsible for a different part of the process. The process starts with the extracted histology image patch, which is inserted into the first module. This module is called the *segmentation module*, which is responsible for predicting segmentation masks. The next module is called the *filtration module*. This module is the most flexible part of the system. It contains a stack of filters used for the filtration of the created segmentation mask. It serves as the second module to expedite the process. By filtering out irrelevant cells before passing them to the classification module, we can streamline the validation process and save time. The last module, called the *classification module* is applied for extracting of specific classes of single cells. By combining these modules, we obtain a classified and segmented mask for a specific patch, which is then filtered based on the required cell features. All of these modules are integrated and implemented within the PyTorch [15] framework to optimize the performance of the deep neural network-based application.

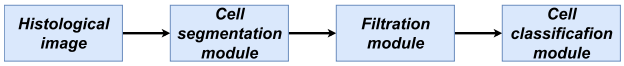


Figure 3: Architecture of system

3.2 Segmentation module

The first part of the system is the segmentation module, which can be seen in fig. 4. This module is responsible for the creation of a segmentation mask. The module is consists of two parts: *data preparation* and *cell segmentation*. In the data preparation part, data loading, splitting images into the same size patches, and data preprocessing are performed. The preprocessing stage involves resizing of data and the conversion of data into tensors. Following data preparation, cell segmentation is performed using the *AttentionUNET* [11] architecture, recognized as the state-of-the-art approach for cell segmentation. In our evaluation, we compared Ordinary UNET, AttentionUNET, and Residual AttentionUNET architectures. Despite similar dice scores averaging around 83%, the AttentionUNET architecture outperformed others in accurately segmenting cells and their edges. This determination was made through visual comparison by a pathologist.

The AttentionUNET architecture consists of a combination of two components: the conventional UNET and an attention module. UNET architecture [16] is a convolution neural network-based structure, based on encoder-decoder architecture enriched by skip connections for enhanced feature extraction. These connections aid in retaining fine details during upsampling by concatenating feature maps from corresponding encoder and decoder layers. Furthermore, an attention gate consisting of two inputs is also in the UNET architecture. The first input is from the layer before and the second is from the encoder part. These inputs are then concatenated and passed through ReLU, convolution layer, sigmoid, and then resampled. This model is trained based on transfer learning on *CoNIC* [2] dataset. CoNIC dataset is used due to small amount of annotated data provided by the pathologist.

3.3 Filtration module

The second part of the system, which can be seen in fig. 5, is the filtration module. This module is the main part of our proposed solution. The primary objective of this module is to offer pathologists options for interacting with the system and modifying the results based on their preferences. It aims to preserve the accuracy of the outcomes as much as possible. This module is separated into the *data preparation* and *cell filtration* parts. The data preparation

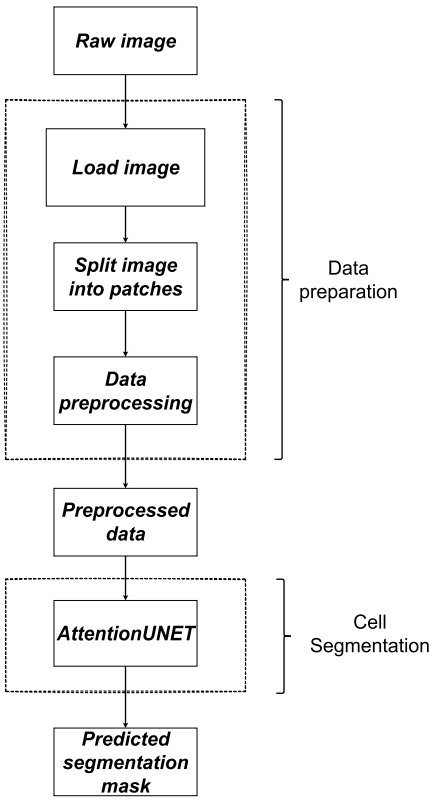


Figure 4: Components of the segmentation module

part is phase involves merging image and mask patches into a single mask and image. This facilitates the subsequent step, which is the usage of a *contour analyzer*. A contour analyzer, implemented using the OpenCV [5] Python library, is utilized to extract single cells from the mask. When we have contours and their positions, we can extract these contours from images. After the extraction of cell contours, we can easily perform filtration. In this stage of development are implemented three filters.

The first filter detects an *area of specific cells*. At first, the area is calculated over each contour. After that, the value of the area is compared to the threshold, and is decided whether the contour passes the test or not.

The second implemented filter is the *shape filter*. This filter also iterates over each contour of the mask and calculates the aspect ratio of elliptic approximation [7]. Using of this calculation solve problem of different cell rotations, because of which cannot be used simple ratio of width and height. The calculation creates value that is used to decide whether the contour is valid or not. Decision is also made by comparison of threshold and calculated ratio, like before.

The third filter is a *color filter*, which is the most complex one. For this filter three input data are crucial. Image, segmentation mask, and sample image. Sample image is one color image, which is set to specific color, based on observations of cells. The first step of the color filter is to extract contours from the mask. Once contours are obtained, specific color pixels of the contour need to be extracted from the image. The next step involves calculating the color ratio by leveraging the difference between two colors in the form of CIE Lab, specifically utilizing the *CIE 2000 method*, across both the image and color sample. This method is implemented with color-science [12] Python library. As the last step, all calculated pixels of the contour are added up and divided by all pixels of the contour. This creates an average color difference value, which is used for filtration.

Mentioned earlier, each of the filters is implemented to work with a specific threshold value. The value tells which cells need to be filtered and which need to be kept. Setting these thresholds directly inside of the annotation tool in which the presented system will be implemented provides the required variance for pathologists. The filtration module is not limited to the mentioned filters; it is designed to allow easy modification and addition of new necessary filters based on the *extraction of cell features*. The choice of these three specific filters is guided by the observations of pathologists during annotation sessions, which highlighted the most important cell features. Further identification of additional filters will involve additional sessions with pathologists.

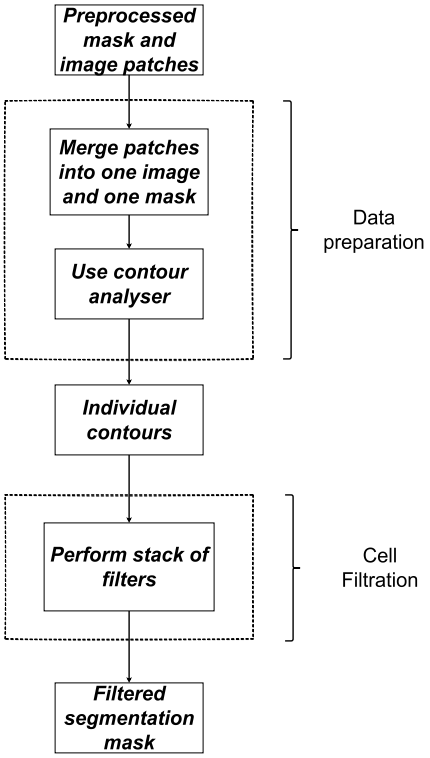


Figure 5: Components of the filtration module

3.4 Classification module

The final component of the system is the classification module (fig. 6) tasked with classifying single cells. These

cells must be categorized into six distinct classes: neutrophil, epithelial, lymphocyte, plasma, eosinophil, and connective. Module is divided into two main parts: *data preparation* and *cell classification*. In the data preprocessing part all patches need to be merged into one image, because of the contour analysis. Following this, single cells can be extracted and prepared for classification. These single cells are then passed to the cell classification part, which is made of *ResNET34* architecture [4]. Variations of ResNET and VGG16 architectures with single and multi-input configurations were explored. Despite the similarities in results across the models, attributed to insufficient training data, the ResNET34 architecture currently yields the best performance. However, as more training data becomes annotated, additional experiments with diverse architectures will be conducted.

ResNET34 architecture can be divided into three parts. The first part is residual blocks with skip connections. These residual blocks contain convolution, batch normalization, and ReLU activation function. Between each of these residual blocks are skip connections. The number of residual blocks defines what type of ResNET architecture it is. When increasing the number of residual blocks, the model is able to get more fine-grained features, but on the other hand, it needs bigger computational resources. Global average pooling is located after residual blocks. It is used for reducing of output's spatial dimension to 1x1, which is then fed to the last layer. The last layer is the fully connected layer, which is used for probabilities and prediction extraction. This model is used for the classification of cells and output labels for single cells.

Based on visualization of *class activation map (CAM)* [1], which can be seen in fig. 7, it is evident that the neural network model is sometimes not focusing on important areas. In fig. 7a can be seen CAM of the correct prediction and in fig. 7b can be seen CAM of the incorrect prediction. From the images, it can be observed that the model sometimes encounters difficulties in determining which pixels to focus on. Because of this problem, a custom attention module is added to ResNET34 architecture. This module is implemented into the architecture after the convolution layers extract weights from the segmentation mask, providing additional information to the model. This adds to the model information about the position of important cells and indicates which pixels are important.

3.5 Training process

The system contains segmentation and classification modules, which need to be trained. Both models are exclusively trained on the CoNIC dataset, which contains colon histology images. Therefore, because of the same type of data in the dataset, it could be used for *transfer learning*. The training process for each of them is slightly different. The segmentation model is trained with dice loss function and Adam optimizer for 100 epochs. On the other hand, the classification model is trained with cross entropy loss and with Adam optimizer, which was trained for 20 epochs. Both of the trainings were performed on Tesla V100-SXM2-32GB with 8 GPU.

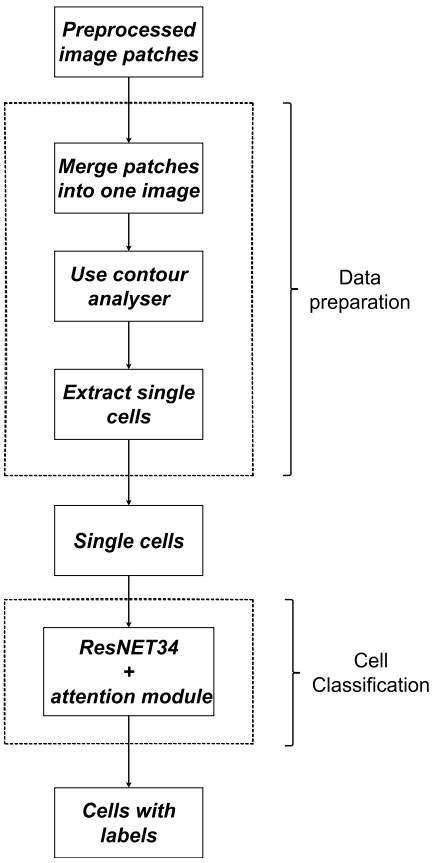


Figure 6: Components of the classification module

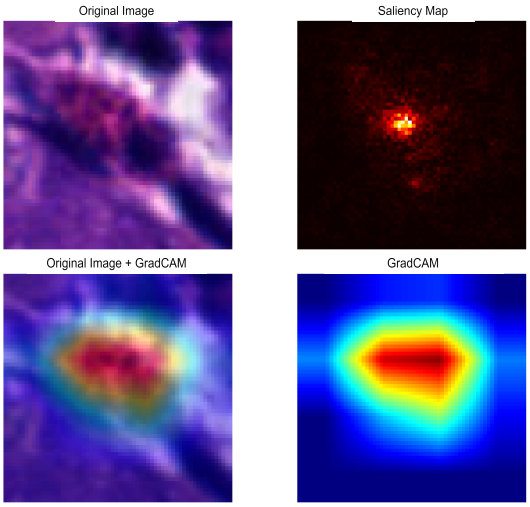
3.6 Evaluation methods

Evaluation methods used in the proposed solution can be separated into three parts: quantitative evaluation, visual inspection, and validation by pathologists.

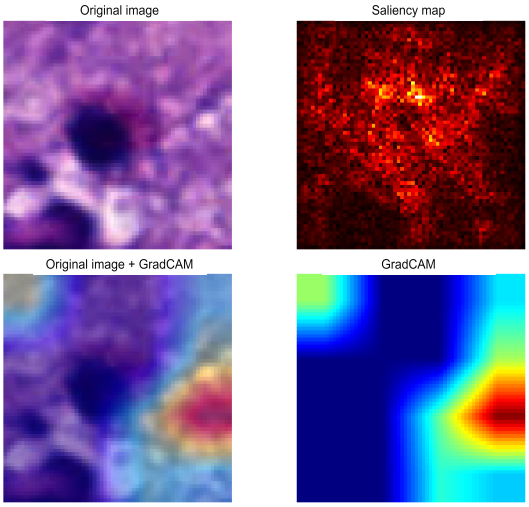
Numeric metrics evaluation can be divided into the segmentation and classification parts, as each task requires distinct metrics for evaluation. For segmentation, *dice score*, *intersection over union*, and *F1-score* are used. On the other hand, classification uses *precision*, *recall*, and *accuracy*. These numeric metrics interpret performance precisely with easily interpretable numeric values.

Visualization, on the other hand, represents graph plots of the training process or exact generated segmentation masks. These can be used as simple methods whether the model is performing well. Another visualization method used in the paper is class activation mapping, which is used for classification. This one is useful to check if the trained neural network is correctly focusing on the important and the most representative areas of the image.

The last evaluation metric used is direct validation by a pathologist. Pathologists review the generated predicted mask and provide feedback on its accuracy and quality.



(a) Class activation mapping of a correct prediction



(b) Class activation mapping of an incorrect prediction

Figure 7: Class activation mapping (CAM)

While this approach is the most informative and useful, its execution frequency is limited by the availability of pathologists' time. The evaluation process can be segmented into two tests: qualitative and quantitative. The first test involves feature extraction for qualitative analysis, where pathologists explain the most significant features during cell annotation. The outcome of this test provides information about important features. The second test is active learning. In this evaluation method, a proposed solution can be also directly integrated into the annotation application, expediting the validation process. For optimal results, specific cells that pose challenges for the classification model need to be provided to the pathologist in an iterative process. The outcome of this approach is a refined predicted mask for our model, which can subsequently be retrained on this problematic data.

4 Results

The results of experiments can be categorized into two parts, one is numerical metrics results and the second one is visualization of results. A combination of these two gives the best insight into the evaluation of computer vision solutions, especially the proposed one.

Numeric value metrics provide results of two different modules. These modules are previously mentioned segmentation and classification modules. Results of the segmentation module can be seen in a table 1. These results provide the difference in dice IoU and F1-score metrics between CoNIC dataset and the custom dataset. The *custom dataset* is made up of converted raw data, directly from the pathologist. Data is in a format of *ndpi*, which was converted and used to extract around *600 annotated cells*. These cells are then used as previously mentioned custom dataset. Metrics shown in the table 1 indicate promising results both on CoNIC and on the custom dataset. The fact that the model is trained only on CoNIC and yet, we can achieve sufficient results without additional training on custom data is great. The second evaluated module is the classification module. The results of this module also provide calculated metrics over CoNIC and custom dataset. The metrics used for this experiment are accuracy, precision, and recall. Based on these results, it can be observed that the model performance is not optimal, but it is satisfactory considering the limited availability of annotated data. Increasing of accuracy will be part of the active learning implementation. On the other hand, the filtration module cannot be evaluated like this because only a pathologist can say if cells in the filtration module are correctly removed.

Segmentation task			
Dataset	Dice	IOU	F1-score
CoNIC	84,18%	72,69%	90,08%
Custom Data	75,22%	66,74%	79,13%

Table 1: Results of segmentation module

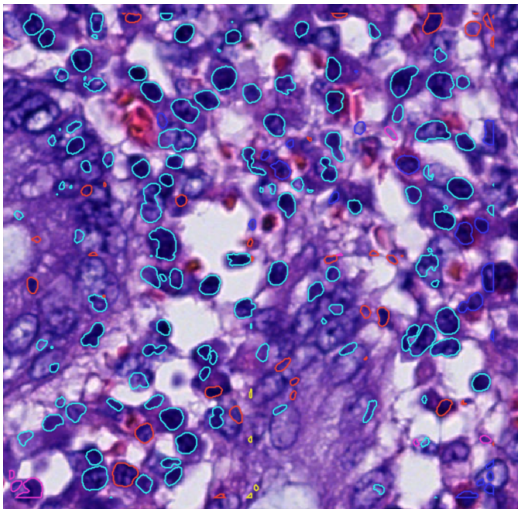
Classification task			
Dataset	Accuracy	Precision	Recall
CoNIC	93,00%	93,00%	93,50%
Custom Data	72,00%	66,00%	70,05%

Table 2: Results of classification module

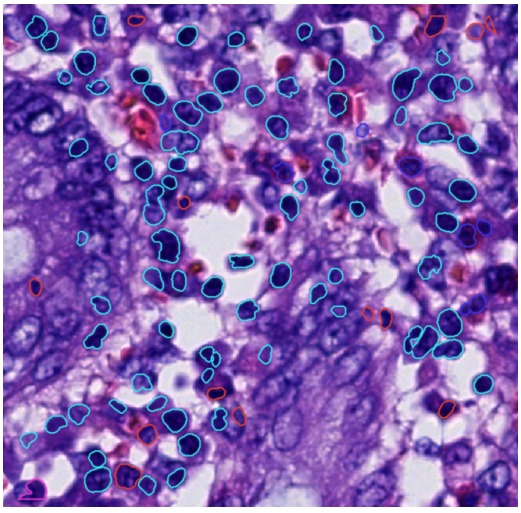
The second result of the visualization of data can be seen in fig. 8. Prediction is performed on histology images never seen by the model. These images have been provided directly by a pathologist. In a comparison of prediction *without filter* in fig. 8a and prediction *with filter* in fig. 8b, it can be observed, that adding of filtration module filtered a significant number of irrelevant cells. When focusing especially on small and oblong cells on a left and

bottom part of the image, it can be seen that in fig. 8b is much less of these cells then in fig. 8a. This filtration is provided with only filters and thresholds, which can be later modified. Modifications can be done anytime without changing neural network model architectures or retraining of models. The advantage of visualization of data is that it is more interpretative. Few results of prediction masks were evaluated directly by pathologists who noted good results, especially with *neutrophils*. These neutrophils are important in the diagnosis of Crohn's disease and Ulcerative Colitis, which makes them one of the most important types of cells for our solution.

The proposed solution of filtration architecture shows that the system can be modified by changing the values of *thresholds* in specific filters. This creates the required modifiability of the system by the user. It shows that it is possible to handle the creation of a prediction mask easily and fast, which was exactly the objective of the research.



(a) Prediction mask without filter module



(b) Prediction mask with filter module

Figure 8: Predicted masks

5 Conclusions

Due to the differences and complexity inherent in histology tasks, it is crucial to develop a system that is user-friendly and capable of accommodating modifications by pathologists. It is important in situations when a pathologist need to find cells based on their features. Sometimes pathologists need to find all specific features in areas of histology image. In such scenarios, challenges arise when the prediction system generates a mask with a large number of cells. Lots of the cells are irrelevant for pathologists who need to search in them for specific cells. The proposed solution helps exactly with this. The contribution shows in results that the difference with and without using of filtration module is significant. The filtration module can focus on *specific features based on filters*. What's more, it is designed to be easily expandable by more filters, based on pathologist needs. Tests have demonstrated that merely *adjusting thresholds* can result in significant changes to prediction masks. With a correct setting of threshold and correct knowledge, it is able to extract individual classes of cells, which are sometimes hard to find. The second advantage that comes with this solution is the fastening of the validation process. Directly after the segmentation module, the filtration module removes lots of contours, which significantly decreases the number of single cells prepared for classification. What is more, it can be beneficial for some pathologists, who have problems with using neural networks as black boxes. When they obtain the feeling that they can modify the model by themself, it can build greater trust in the system.

6 Future work

Implementation of a filtration module into the system has shown advantages in the classification task of single cells. However, pathologists need to be able to use this system. This involves implementing the system into the *custom annotation tool*, which is currently in development at the faculty. Following the implementation of the system in the annotation tool, the next intriguing approach is to adapt the system for *active learning*. This active learning will be based on output from the pathologist directly using a custom annotation tool. This could help the model with a better understanding of data and enhance the performance of the system. Also, the implementation of *new types of filters* and new methods of feature extractions into the system might be beneficial to try. By offering pathologists a wider array of filters, the system becomes more versatile and adaptable, ultimately enhancing its utility and efficacy in medical image analysis.

References

[1] José P. Amorim, Pedro H. Abreu, João Santos, Marc Cortes, and Victor Vila. Evaluating the faithfulness

of saliency maps in explaining deep learning models using realistic perturbations. *Information Processing and Management*, 60(2):103225, 2023.

[2] Graham Simon and et al. Conic: Colon nuclei identification and counting challenge 2022. *arXiv preprint arXiv:2111.14485*, 2021.

[3] Md. Tarek Aziz, S. M. Hasan Mahmud, and et al. A novel hybrid approach for classifying osteosarcoma using deep feature extraction and multilayer perceptron. *Diagnostics*, 13(12), 2023.

[4] Lokesh Borawar and Ravinder Kaur. Resnet: Solving vanishing gradient in deep networks. In Rajendra Prasad Mahapatra, Sateesh K. Peddoju, Sudip Roy, and Pritee Parwekar, editors, *Proceedings of International Conference on Recent Trends in Computing*, pages 235–247, Singapore, 2023. Springer Nature Singapore.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Gianni Brauwers and Flavius Frasincar. A general survey on attention mechanisms in deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3279–3298, 2023.

[7] Hercules G. Dimopoulos. *The Elliptic (Cauer) Approximation*, pages 143–183. Springer Netherlands, Dordrecht, 2012.

[8] Marietta Iacucci and et al. Tommaso Lorenzo Parigi. Artificial intelligence enabled histological prediction of remission or activity and clinical outcomes in ulcerative colitis. *Gastroenterology*, 164(7):1180–1188.e2, 2023.

[9] Barbora Kadlečková. Idiopatické črevné zápaly, Dec 2023.

[10] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):6999–7019, 2022.

[11] Dhiraj Maji, Prarthana Sigedar, and Munendra Singh. Attention res-unet with guided decoder for semantic segmentation of brain tumors. *Biomedical Signal Processing and Control*, 71:103077, 2022.

[12] Thomas Mansencal, Michael Mauderer, and et al. Parsons, Michael. Colour 0.4.4, dec 2023.

[13] R. Monarch, R. Munro, and C.D. Manning. *Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-centered AI*. Monarch, 2021.

[14] Arshi Parvaiz, Muhammad Anwaar Khalid, Rukhsana Zafar, Huma Ameer, Muhammad Ali, and Muhammad Moazam Fraz. Vision transformers in medical computer vision—a contemplative retrospection. *Engineering Applications of Artificial Intelligence*, 122:106126, 2023.

[15] Adam Paszke, Sam Gross, and et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[16] Nahian Siddique, Sidike Paheding, Colin P Elkin, and Vijay Devabhaktuni. U-net and its variants for medical image segmentation: A review of theory and applications. *Ieee Access*, 9:82031–82057, 2021.

[17] Chetan L. Srinidhi, Ozan Ciga, and Anne L. Martel. Deep neural network models for computational histopathology: A survey. *Medical Image Analysis*, 67:101813, 2021.

[18] Serdar Yildiz, Abbas Memiş, and Songül Varl. Nuclei segmentation in colon histology images by using the deep cnns: A u-net based multi-class segmentation analysis. In *2022 Medical Technologies Congress (TIPTEKNO)*, pages 1–4, 2022.

[19] Zitao Zeng, Weihao Xie, Yunzhe Zhang, and Yao Lu. Ric-unet: An improved neural network based on unet for nuclei segmentation in histology images. *IEEE Access*, 7:21420–21428, 2019.

Multimodal Brain MRI Registration Using Generative Adversarial Networks

Norbert Vígh*
Supervised by: Vanda Benešová†

Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava / Slovak Republic

Abstract

Deep learning methods have recently found applications in several fields, including the processing of medical imaging data. We explore the application of convolutional neural networks (CNN) for automatically processing magnetic resonance imaging (MRI) scans in ceT1- and T2-weighted modalities to assist doctors with executing accurate and time-efficient tumor diagnostics.

The main challenge of the work is the multimodal registration of coronal and axial scans, which are perpendicular to each other and, therefore, cannot be registered directly. We use a generative adversarial network (GAN) architecture to convert between modalities, making it easier to register them. The resulting registered scans can be used for a wide variety of further tasks, utilizing the complementary information contained in different MRI modalities, i.e. image segmentation.

Keywords: Multimodal Image Registration, Deep Neural Networks, Generative Adversarial Networks, Medical Imaging

1 Introduction

In recent years, there has been a substantial rise in novel techniques used for diagnostics in medicine. The introduction of artificial intelligence (AI) can speed up most of the tasks that doctors perform daily.

Despite legal and ethical challenges, it can still greatly aid the doctor. The most influential advantage of using AI is the reduced time it takes to diagnose a patient since, in general, the AI can process more data quicker than even an experienced doctor could.

One of the main problems in the learning process of AI is the need for a lot of data. Many tasks require using annotated data that must first be created by a domain expert, which is very tedious. By creating an intelligent assistant based on a neural network, we could simplify the process of data annotation and thus help create larger datasets available for future usage.

In many cases, doctors need to work with multiple modalities, providing complementary information, sometimes even acquired by using several different diagnostic devices. Due to this, we see a new research possibility in intermodal conversion. Although intermodal conversion is not able to synthesize information that is not present in the original imaging data, it can be sometimes useful, for example for our presented co-registration method. By creating a tool that could convert images between modalities, we could reduce the time needed to scan the patient multiple times, thus enormously reducing the waiting times to get diagnosed.

To achieve this, we first need to be able to **register modalities** correctly in the same space. Afterward, we can add the provided labels from one modality to the second modality. This enables the creation of neural networks that could be trained on a compacted dataset. This could possibly create a tool that could classify into classes that are not that visible in the actual modality provided to the doctor, but the neural network could pick up on this.

Our contribution lies in creating a **multimodal 3D image registration tool** using a GAN network. The core problem we try to overcome is the perpendicularity of the available dataset, which makes this problem even harder. Unfortunately, most real-world MRI data uses an anisotropic voxel grid. The spacing of the scans is different on the third axis, which motivates the creation of a custom-made image registration algorithm. To make the problem more graspable, we reduce the complexity of registering two modalities by introducing a GAN network that converts images between MRI modalities, making the registration process more straightforward.

2 Background

Pituitary adenomas (PA) are a type of benign tumor affecting the pituitary gland, with a prevalence of 96 ± 20 cases per 100,000 population [8]. Despite being benign, PA can exert pressure on surrounding tissues, necessitating cau-

*xvighn@stuba.sk
†vanda_benesova@stuba.sk

tious surgical intervention or they can cause abnormal hormonal production [18]. A precise diagnosis and characterization of PA is crucial for treatment planning and patient care. The segmentation mask can be used for further statistical analysis of the tumor, such as classification based on its diameter [16, 4] and shape relative to surrounding tissues [12], or radiomics analysis, which extracts quantitative features from medical images to predict patient outcomes [9, 15].

Medical imaging plays an essential role in tumor diagnosis. Techniques such as X-rays, CT scans, MRI, and ultrasound provide detailed scans for analysis. MRI scans, in particular, offer high-resolution images acquired in different planes, allowing for better visualization of anatomical structures and abnormalities. Different MRI modalities, like T1-weighted and T2-weighted, offer varying tissue contrast and information about tissue composition and structure, aiding in the detection and characterization of tumors [16, 4, 3].

Computer vision techniques are increasingly employed in medical image processing for tasks such as image registration and segmentation. Image registration aligns images from different modalities or time points, enabling accurate comparison and analysis. Segmentation identifies and labels regions of interest within images, facilitating quantitative analysis and diagnosis. Evaluation metrics such as mutual information and the Dice similarity coefficient (DSC) assess the accuracy and performance of these techniques, guiding their optimization and application in clinical settings [14].

The rise of deep neural networks, particularly convolutional neural networks (CNNs), has revolutionized medical image analysis by enabling automated feature extraction and classification. Architectures like U-net [17] excel in segmentation tasks, accurately delineating tumor boundaries and aiding in treatment planning. Generative models like variational autoencoders (VAEs) [11] and generative adversarial networks (GANs) [10] offer novel approaches for data generation and augmentation, enhancing the availability and diversity of training data for deep learning models.

2.1 Related Work

The fundamental problem with multimodal registration is that most models try to find a good mapping between the different intensities in the fixed and moving images. Depending on the actual modalities, it might be problematic to find such mappings in some cases. However, with the recent advancements in CNNs, there have been attempts to **facilitate the registration process by applying segmentation to both modalities** and subsequent image registration in the space of the segmented images.

One such work was published by Blendowski et al. [2] The authors suggest using a **convolutional autoencoder** architecture to extract shape features of both modalities. Using the encoder part on the input volume, they extract a 1584-dimensional shape space describing the objects in the volume. They propose that by applying linear interpolations between the moving and fixed image encodings, they can achieve iterative guidance of the image registration process. This should help the registration process by eliminating large non-linear deformations that could occur when the algorithm tries to register the moving image directly onto the fixed image. Their results confirm this, since when applying direct registration, they only achieved a DSC of 0.526, while when using the approach with iterative guidance, they achieved a DSC of 0.653 [2].

Another work by Cao et al. [5] is focused on the registration of CT and MRI using models that can **synthesize images of either modality** from the other image. The subsequent image registration is then much more manageable. However, especially synthesizing MRI from CT is a complicated, non-linear task. The authors do not use a neural network but rather a Multi-Target Regression Forest since, this way, they arguably needed less training data. The forest synthesizes both modalities, so the registration algorithm has two images (one original and one synthesized) in both modalities. The calculations can then be done in both modalities, where one way of transformation is inverse to the other way. The paper by Cao et al. [5] was used for MRI and CT pelvic image registration. However, there have been other papers about synthesizing other modalities for brain-related data, even by using neural networks. In a paper by Li et al. [13], the authors try synthesizing MRI from CT of brains using deep learning methods. They compare approaches based on a CycleGAN, Pix2Pix (conditional GAN) model, and the U-net architecture. Unexpectedly, the best results were achieved using the U-net architecture with L1 and L2 regularizations [13].

In a paper by Zheng et al. [20], the authors propose a method for multimodal image registration using a GAN network. Compared to the CycleGAN model used for image synthesis in the previous section, the authors propose a **Symmetric registration GAN model**, which also creates a cycle-consistent mapping between the two modalities. However, a transformation is also applied to the images, which transforms them into the same space as an image from the other modality. This way, even though the images are not aligned, they can be directly compared, and the loss can be calculated more precisely [20]. The transformation is done using Affine Transformation Regressors, which try to predict the affine transformation parameters between the two images they receive as input. There are also non-linear transformation regressors, which try to predict the non-linear transformation parameters using a VoxelMorph model [1]. These regressors are pre-trained on a dataset of artificial transformations with known parameters. Ultimately, the proposed model generates two symmetric transformations, which can be applied to register the images in either way [20].

3 Our solution

The main contribution of our work lies in the way we use data from multiple modalities by creating a registration algorithm that could enable the usage of labels of classes from multiple modalities, which are not greatly visible on only a scan from one modality. A neural network can still detect such cases, meaning full diagnostics could be possible by providing only one modality.
We have the following goals:

- to help the process of creation of new more robust datasets,
- to register MRI scans in multiple modalities, which could generate a dataset with labels from both modalities,
- to prove the possibility of precise segmentation of classes even if provided with an input consisting of only one modality,
- facilitate further research in segmentation using only one modality by providing a tool to create a dataset with labels from both modalities.

In summary, we aim to aid the diagnostics of pituitary adenoma by providing a tool that should enable the creation of more extensive datasets and incite more research in this field.

To make the next sections more followable, we introduce a notation of different kinds of scans. Letters C and A show whether the scans are coronal or axial respectively. These letters can be followed by an apostrophe $'$, indicating that the image was crated by inference through the GAN network.

3.1 Dataset

We obtained the dataset based on cooperation with a doctor from the Military University Hospital in Prague, It consists of Axial T2-weighted (A_{T2}) and Coronal contrast-enhanced T1-weighted (ceT1, C_{ceT1}) MRI scans in the Nifti format. The dataset is anonymized and contains no personal information about the patients.

In total, we have MRIs of 928 patients. However, the label masks in both modalities are only available for 330 patients, which makes the base of our dataset. The other 340 patients scanned in both modalities have annotations on neither (or just one) of the modalities. Annotating many MRI scans is very time-consuming, so even in the annotated samples, only a handful of 2D scans are annotated, and others are left untouched.

Axial scans are scans in the horizontal plane. They consist of 22 to 52 cross-sections with resolutions between a quarter and a half millimeter. On the contrary, coronal images are cross-sectional and consist of 12 to 24 cross-sections with a resolution of half to one millimeter.

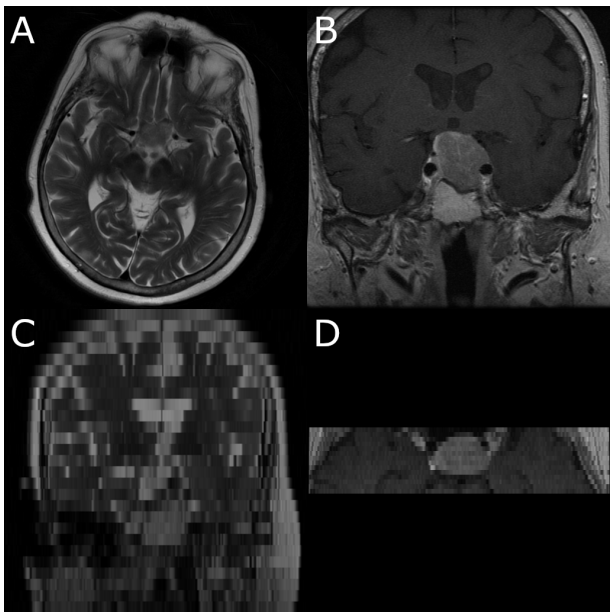


Figure 1: Comparison of axial (A) and coronal (B) scans. Axial scans shown in coronal plane (C) and coronal scans shown in axial plane (D)

There are several crucial problems with the dataset that we need to solve:

- The third axis of the resolution is always significantly worse and can be around 2 to 5 millimeters. Doctors only create a few slices, showing key brain areas that must be examined. This makes the MRI sampling faster, more convenient for the patient, and more economically feasible. Minimizing the examination time can also decrease the spatial shift between subsequent scans caused by patient movement.
- Another problem is that the axial and coronal slides may not be precisely perpendicular. Therefore, no clear transformation is available that could align these two modalities, and thus, it has to be calculated using image registration methods.

Label masks are available for both modalities; however, the classes are not equivalent. It is impossible to distinguish some tissue types with certainty in the respective modalities. The modality will be selected based on the tissue type the doctors want to examine. For example, arteries marked on ceT1-weighted scans may be poorly distinguishable from the surrounding tissue on T2-weighted scans.

In addition, we use another dataset of registered coronal MRI scans. This dataset consists of scans of 1157 patients, all of which were scanned in the coronal plane, half of which are contrast-enhanced T1-weighted and the other half T2-weighted slices. In total, this represents 10802 slices of each modality. Each sample is paired with a corresponding sample from the other modality, meaning that each patient was scanned in both modalities.

3.2 Challenges

There are several problems that we need to face:

- The moving image are just approximately perpendicular to the fixed image (can differ up to 5 degrees).
- The spacing of A_{T2} is too large, so we cannot interpolate them to an isotropic space, instead we can interpolate C_{ceT1} scans with a smaller spacing.
- C_{ceT1} scans stick out of the space of the A_{T2} scans, necessitating the need to find the correct subset of C_{ceT1}.

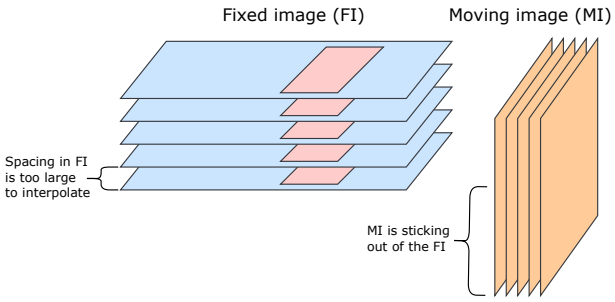


Figure 2: The core problems of the used dataset. (blue) axial slices, (orange) coronal slices, (pink) expected transformation of coronal slices into the space of axial slices.

3.3 Proposed pipeline

Figure 3 shows an overview of the proposed pipeline for processing the data.

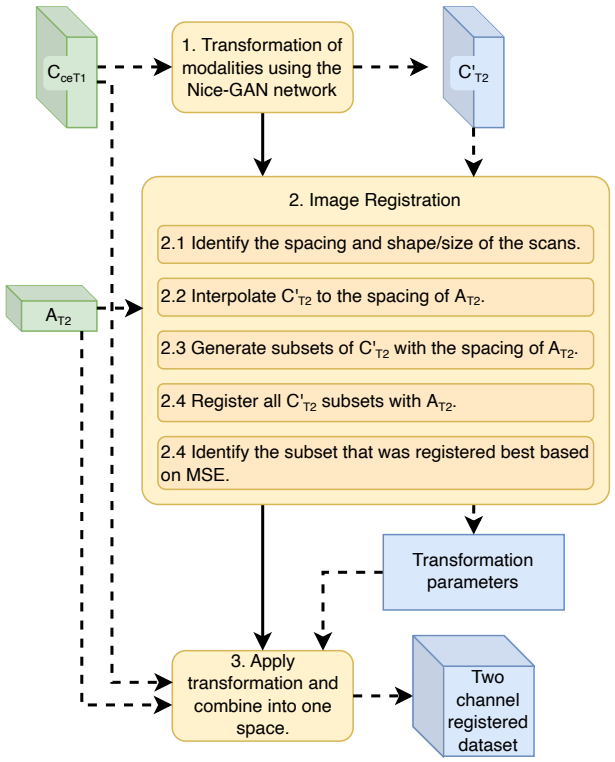


Figure 3: The data-flow chart of the processing pipeline. Green blocks show data that is used as input, blue blocks are data generated by the pipeline.

3.4 GAN network for intermodal conversion

The first step of our pipeline is to convert one of the modalities into the other modality.

We use the Nice-GAN architecture [7, 6], an improved version of the CycleGAN architecture. Compared to the CycleGAN, this architecture reuses the results from the Encoder part of the generator, which are then further processed in the Discriminator network. This makes the training process more stable and allows the generator to learn more complex mappings [7].

Moreover, this architecture tries to get to the same hidden vector (latent space) from both modalities, which makes the mapping more consistent. This is achieved by using a shared latent space, which is then used to reconstruct the original image since they can switch the decoders to generate the other modality. So, there are two loss functions. The first one is the reconstruction loss, which is calculated by comparing the original and reconstructed images of the same modality. The other loss is the cycle loss, calculated by comparing the original image and the image obtained by converting the original image to the other modality and then back to the original modality [7].

We train this network on the datatest mentioned in the last paragraph of Section 3.1.

The training was done on a machine with an NVIDIA RTX4090 desktop GPU with 24GB of VRAM. It ran for 40,000 iterations and took about 10 hours to complete. The results were saved after every 10,000 iterations, and visualization was generated for the intermediate results after every 1000 iterations. The learning rate was set to $1 * 10^{-4}$ with a batch size 1 and Adam optimizer.

We used the results of our GAN to convert C_{ceT1} into C'_{T2} in a slide-by-slide manner.

3.5 Our registration algorithm

The steps of the registration algorithm are indicated in Fig. 3. In this section we describe these steps in detail.

3.5.1 Transformation of coronal to axial slices

The images were first loaded to read the Nifti files, including their metadata. The metadata includes information about the spacing of the scans, which is crucial for the registration process. Based on the spacing, we were primarily interested in the third axis, which had the largest spacing.

The C'_{T2} scans were then interpolated to the same spacing as the A_{T2} scans. The interpolation was done by calculating the expected slice dimensions based on the spacing of A_{T2} scans and then interpolating C'_{T2} scans to the same spacing.

We have created a helper function to rotate C'_{T2} scans by a given angle.

3.5.2 Slice selection

From the interpolated C'_{T2} slices, we can extract subsets of the slices with the same spacing as A_{T2} scans. We calculate the number of slices that must be skipped between each slice to achieve the desired spacing. Then, starting from the first slice, we extract subsets of slices separated by the calculated number of slices. This process is repeated until we reach the end of the C'_{T2} scans. We are left with about 500 subsets of C'_{T2} slices, which are not disjoint.

To optimize the registration process, we can remove about 50% of the subsets since the C'_{T2} scans overflow from A_{T2} scans as depicted in Fig. 2.

3.5.3 Registration

The registration algorithm itself is based on finding the best transformation of the C'_{T2} scans into the space of the A_{T2} scans. We use the Mean Squared Error metric. This algorithm is run on each subset of C'_{T2} slices separately, and the MSEs and transformations used to achieve them are saved for each subset.

The registration is initialized by creating a transformation where the C'_{T2} slides are placed in the middle of the image. Since we are working with pituitary adenoma, which is located in the sellar region of the brain, we can assume that the tumor is approximately located in the middle of the image. This is a good starting point for the registration process. We calculate the MSE of the C'_{T2} scans nudged by a few pixels in each direction as well as rotated by one degree in each way. We choose the direction with the lowest MSE, creating a simple gradient descent algorithm. The transformation is then updated in the chosen direction by a number of pixels dependent on a learning rate defined as a hyperparameter of the algorithm. However, the learning rate decays during the registration process to prevent the algorithm from overshooting the optimal transformation.

We apply a function that calculates a new learning rate based on the index of the current iteration and the total number of iterations. The function is defined as seen in Eq. 1, where i is the index of the iteration, n is the total number of iterations, lr_0 is a hyperparameter of the training process.

$$lr_i = \frac{lr_0}{\frac{i}{n} + lr_0} \tag{1}$$

The best transformation is identified by comparing the achieved MSEs of all the subsets. The subset with the lowest MSE is chosen as the best one, and the transformation used to achieve it is saved.

Upon successfully identifying the best subset, we can apply the transformation to the *original* C_{ceT1} scans, and the result can be saved as a 3D tensor with multiple channels, each representing one of the modalities.

4 Results and evaluation

In this section we present the results of our registration pipeline. To fairly evaluate the results, we used a baseline method to compare the results with.

4.1 Baseline multimodal registration

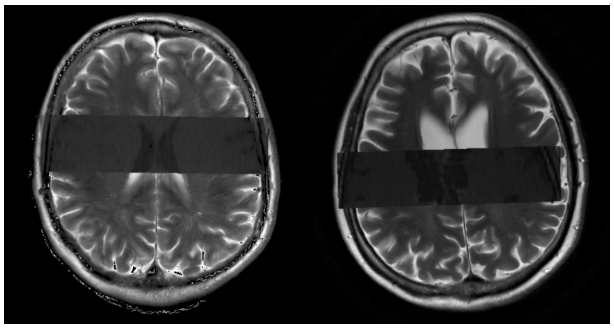


Figure 4: Results of image registration using the BSpline interpolation method. The first image shows a successful registration, while the second image was unsuccessful.

The registration was performed by an accommodated version of the sample code from the SimpleITK image-analysis notebooks collection [19] adjusted for 3D image registration. First, it was necessary to interpolate the C_{ceT1} slices to achieve isotropic voxels via BSpline or linear interpolation. Subsequently, we registered C_{ceT1} scans using the Mutual Information metric since we were dealing with multimodal images.

Regardless of the interpolation method, the registration was unsuccessful in many cases. The results of the registration using BSpline interpolation are shown in Fig. 4. Changing the interpolation method to linear interpolation did not improve the results significantly. To visualize the registration results, we overlapped the original A_{T2} scans with the C_{ceT1} scans (a dark horizontal rectangle).

4.2 GAN evaluation

The GAN network was trained to convert the input images between the two domains of MRI scans. The training setup is described in Sec. 3.4.

Since the GAN network is trained on a paired dataset, we can directly compare the original and generated images. Fig. 5 compares the input and output images. The results are of high quality, especially for the generated ceT1-weighted scans. For generated images of both modalities, the overall position of the whole head and all the structures inside the head are preserved. This is crucially important for the registration method to succeed. The contrast and lightness of the images are also very similar to the original images. There seems to be more difference in the lower part of the scans, but this is not a problem since the registration method only uses the upper part of the scans.

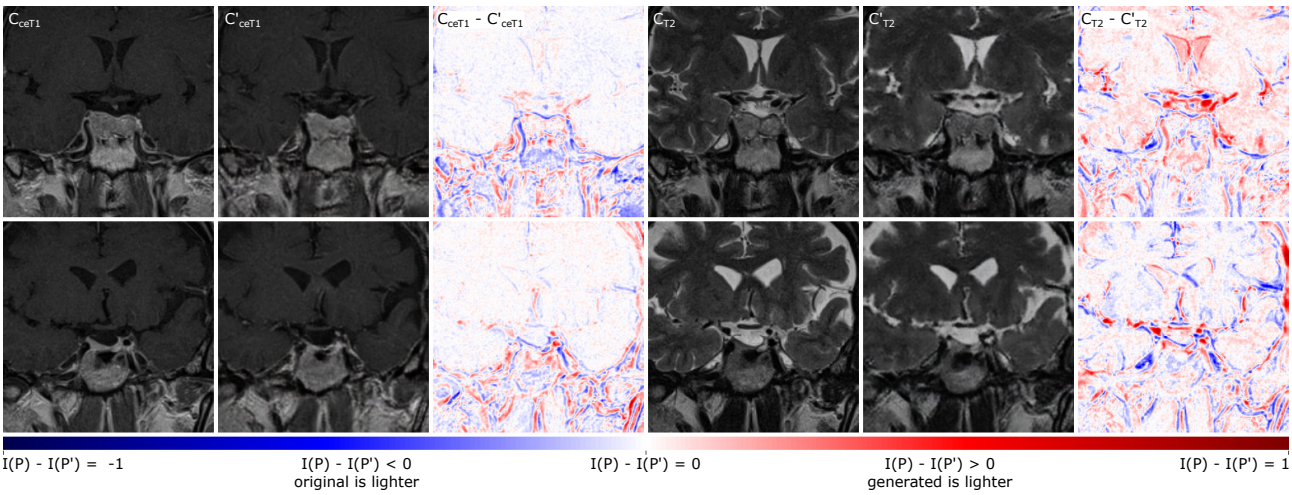


Figure 5: Comparison of original and GAN-generated coronal slices. Column 1 - C_{ceT1}. Column 2 - C'_{ceT1} generated from C_{T2} in column 4. Column 3 - difference between C_{ceT1} and C'_{ceT1} scans using the depicted colorscale. Columns 4 to 6 show the same for the T2 scans. $I(P)$ and $I(P')$ are the intensities of pixels in original and generated images respectively.

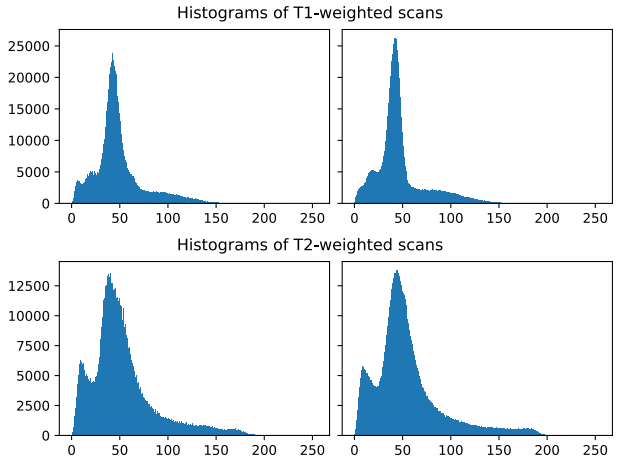


Figure 6: Comparison of histograms of the original and generated scans. Left column - original scans C_{ceT1} and C_{T2}, Right column - generated scans C'_{ceT1} and C'_{T2}.

	MSE of C'_{ceT1}	**MSE of C'_{T2}**
Mean	547.00	856.25
St. Dev.	181.11	292.48
Min	244.16	434.10
Lower quartile	437.74	724.79
Median	520.95	823.96
Upper quartile	612.72	931.70
Max	2969.93	4578.64

Table 1: Overview of the distribution of achieved MSEs by the GAN network over the paired dataset.

Based on the histograms displayed in Fig. 6, we can see that the intensity distribution of the generated images is very similar to the intensity distribution of the original images. This is a good sign since it means that the GAN network can preserve the intensity distribution of the input images as the generator network is trained to produce images that are indistinguishable from the original images.

Additionally, we compared the original and generated slices using the MSE metric. The distribution of these values is shown in Table 1. Similarly to our previous observations, we can see that the differences are more significant when generating T2 scans.

4.3 Registration method evaluation

The registration method described in Sec. 3.5 uses coronal slices converted from C_{ceT1} slices to C'_{T2} slices. We use the MSE of the C'_{T2} and the original A_{T2} slices to identify the correct subset of C'_{T2} slices.

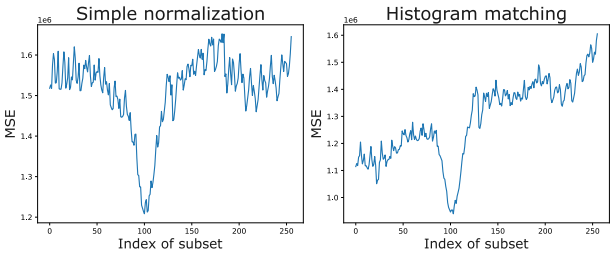


Figure 7: Cumulative MSEs over the subsets: Simple normalization (left) vs. Histogram matching (right). We can use this to identify the correct subset of matching images.

In Fig. 7, we can see the MSEs of all the subsets. We omitted the MSEs of the subsets that were removed from the registration process due to sticking out of the fixed image, as shown in Fig. 2. These subsets would show an even higher MSE than the ones shown in the chart.

The chart shows the cumulative MSEs of the subsets, which are calculated as the mean of the MSEs of the indi-

vidual slices in the subset. As seen in Fig. 7, we can effortlessly identify the correct subset of C'_{T2} slices by looking at the global minima of the cumulative MSEs.

The correct subset has an index of about 100. The registration results are shown in Fig. 8 over several slices of the selected subset.

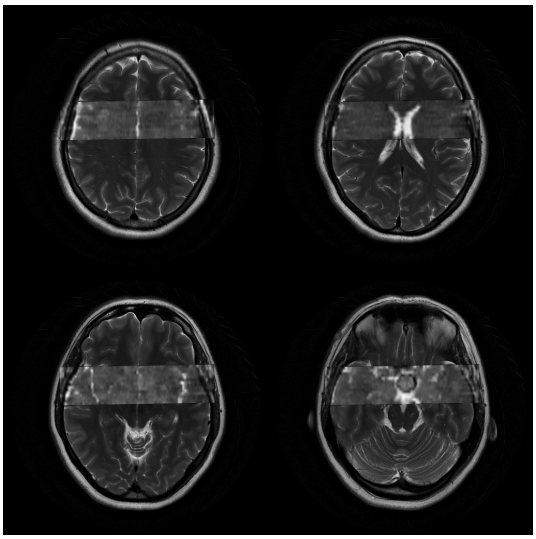


Figure 8: Registration results. Four A_{T2} slices of the same patient overlaid with the generated C'_{T2} slices.

The registration method was able to align the two images very well. The structures of the brain are aligned almost perfectly. The only visible differences are the graininess and lightness of the images. The interpolation of C'_{T2} slices to the same spacing causes graininess, which could be solved by applying some denoising techniques.

We opted for using just a simple normalization to the images, which converts the intensities of the produced images to the same range of intensities as the original images. Additionaly, we tried using a more sophisticated normalization technique, such as histogram matching, which matched the intensity distributions of the generated and original images over the area where we have both modalities. However, as depicted in Fig. 7, this approach didn't yield results that would make the process of the global minima identification clearer, and the registration algorithm took significantly longer to complete. Due to these reasons, we see no real benefit in a better normalization process.

Finally, we evaluate the achieved MSEs over a set of 77 patients. The registration process yielded in most of the cases relatively low MSEs as seen in Table 2 and Fig. 9. There have been some results that didn't converge to the correct results. Upon inspecting results of all 77 patients, we have found only 4 results that were noticeably misaligned, all of which were in the top 6 results with highest MSEs. We therefore evaluated, that the results with an MSE higher than 6×10^6 should be discarded as nonsuccessful. This leaves us with 73 successful registrations meaning a $94,81\%$ success rate.

	MSE of the registration
Mean	2.811766×10^6
St. Dev.	1.980013×10^6
Min	5.884174×10^4
Lower quartile	1.464993×10^6
Median	2.275179×10^6
Upper quartile	3.716739×10^6
Max	1.037747×10^7

Table 2: Overview of the distribution of achieved MSEs over the registered dataset.

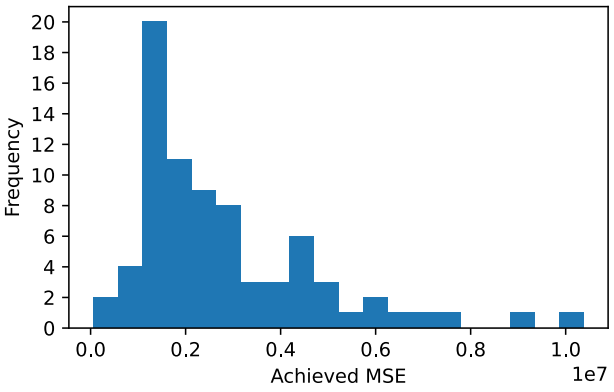


Figure 9: Histogram of achieved MSEs over the registered dataset.

5 Conclusions

We managed to create a comprehensive review of the main challenges of this field and gain an overview of the related state-of-the-art works. We ran several experiments that directed our research toward solving the most pressing issues. Moreover, we have employed a GAN network to convert images between MRI modalities. This helped to create a more robust image registration algorithm that yields respectable results of an acceptable accuracy.

Creating an excellent multimodal image registration is an important step that enables the fusion of the masks from the axial and coronal slides and, thus, the training of a segmentation network to segment tissue classes only marked for the other type of modality. As part of future work, we aim to create a segmentation CNN trained on the created registered dataset as a proof-of-concept of segmentation capabilities beyond what is possible when using scans of multiple modalities separately.

Acknowledgment

We want to thank MUDr. Martin Černý (Military University Hospital Prague) for providing the extensive PA dataset with annotations, the registered multimodal dataset of coronal MRI scans, for his valuable expertise in radiology, and his feedback on this paper.

References

[1] Guha Balakrishnan, Amy Zhao, Mert R. Sabuncu, Adrian V. Dalca, and John Guttag. An unsupervised learning model for deformable medical image registration. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9252–9260, 2018.

[2] Max Blendowski, Nassim Bouteldja, and Mattias P Heinrich. Multimodal 3D medical image registration guided by shape encoder–decoder networks. *International journal of computer assisted radiology and surgery*, 15:269–276, 2020.

[3] Jean-François Bonneville, Fabrice Bonneville, and Francoise Cattin. Magnetic resonance imaging of pituitary adenomas. *European radiology*, 15:543–8, 04 2005.

[4] Michael Buchfelder and Sven Schlaffer. Imaging of pituitary pathology. *Handbook of clinical neurology*, 124:151–166, 2014.

[5] Xiaohuan Cao, Jianhua Yang, Yaozong Gao, Qian Wang, and Dinggang Shen. Region-adaptive deformable registration of CT/MRI pelvic images via learning-based image synthesis. *IEEE Transactions on Image Processing*, 27(7):3500–3512, 2018.

[6] Runfa Chen. NICE-GAN-pytorch: Official PyTorch implementation of NICE-GAN: Reusing discriminators for encoding: Towards unsupervised image-to-image translation. `https://github.com/alpc91/NICE-GAN-pytorch`, 2020, Accessed on 05.03.2024.

[7] Runfa Chen, Wenbing Huang, Binghui Huang, Fuchun Sun, and Bin Fang. Reusing discriminators for encoding: Towards unsupervised image-to-image translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8165–8174, 2020.

[8] Adrian F. Daly and Albert Beckers. The epidemiology of pituitary adenomas. *Endocrinology and Metabolism Clinics of North America*, 49(3):347–355, 2020.

[9] Robert J Gillies, Paul E Kinahan, and Hedvig Hricak. Radiomics: Images are more than pictures, they are data. *Radiology*, 278(2):563–577, 2016.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[12] Engelbert Knosp, Erich Steiner, Klaus Kitz, and Christian Matula. Pituitary adenomas with invasion of the cavernous sinus space: a magnetic resonance imaging classification compared with surgical findings. *Neurosurgery*, 33(4):610–618, 1993.

[13] Wen Li, Yafen Li, Wenjian Qin, Xiaokun Liang, Jianyang Xu, Jing Xiong, and Yaoqin Xie. Magnetic resonance image (MRI) synthesis from brain computed tomography (CT) images based on deep learning methods for magnetic resonance (MR)-guided radiotherapy. *Quantitative imaging in medicine and surgery*, 10(6):1223, 2020.

[14] F. Maes, D. Vandermeulen, and P. Suetens. Medical image registration using mutual information. *Proceedings of the IEEE*, 91(10):1699–1722, 2003.

[15] Marius E. Mayerhoefer, Andrzej Materka, Georg Langs, Ida Häggström, Piotr Szczypiński, Peter Gibbs, and Gary Cook. Introduction to radiomics. *Journal of Nuclear Medicine*, 61(4):488–495, 2020.

[16] Gerald Raverot, Emmanuel Jouanneau, and Jacqueline Trouillas. MANAGEMENT OF ENDOCRINE DISEASE: Clinicopathological classification and molecular markers of pituitary tumours for personalized therapeutic strategies. *European Journal of Endocrinology*, 170(4):R121–R132, 04 2014.

[17] 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.

[18] Walavan Sivakumar, Roukoz Chamoun, Vinh Nguyen, William T Couldwell, et al. Incidental pituitary adenomas. *Neurosurgical focus*, 31(6):E18, 2011.

[19] Ziv Yaniv, Bradley C Lowekamp, Hans J Johnson, and Richard Beare. SimpleITK image-analysis notebooks: a collaborative environment for education and reproducible research. *Journal of digital imaging*, 31(3):290–303, 2018.

[20] Yuanjie Zheng, Xiaodan Sui, Yanyun Jiang, Tongtong Che, Shaoting Zhang, Jie Yang, and Hongsheng Li. SymReg-GAN: Symmetric image registration with generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5631–5646, 2022.

AnnotAid: AI-Driven Data Annotation Tool for Histology Images

Peter Škríba*
Adam Bublavý†
Angelika Kissová‡
Supervised by: Vanda Benešová§ & Kristína Mikuš Kuracinová¶

Faculty of Informatics and Information Technologies
Slovak University of Technology
&
Faculty of Medicine, Comenius University
Bratislava / Slovakia

Abstract

Automatic processing of digital histology images can greatly benefit from the utilization of deep learning methods. The development of such methods requires large amounts of annotated histological images. However, currently available annotation tools often have very poor usability, resulting in ineffective annotation processes. We aim to address the urgent need for a simplified approach to annotating histopathology images, a task that is crucial for advances in automated diagnosis and analysis. By combining our expertise, we strive to develop a user-friendly annotation tool integrated with state-of-the-art deep learning techniques. This tool is designed to alleviate the burden on pathologists during the annotation process by leveraging artificial intelligence models adapted to the various challenges in the field, such as the Nottingham Grading System of breast cancer. Through a comprehensive analysis of breast cancer and existing annotation tools, we propose a solution in the form of a multiplatform annotation tool powered by AI, developed in close cooperation with medical domain experts. By combining our knowledge and resources, we aim to bridge the gap between manual annotation processes and the potential of AI-based solutions, which will ultimately improve patient outcomes and advance medical research in breast cancer diagnosis.

The annotation tool is available at: **annotaid.com**.

Keywords: Computer Vision, Annotation Tool, Deep Neural Networks, User Experience (UX), Nottingham Grading System, Histopathology

*xskriba@stuba.sk
†xbublavy@stuba.sk
‡kissova154@uniba.sk
§vanda_benesova@stuba.sk
¶kuracinova1@uniba.sk

1 Introduction

Artificial intelligence has affected many areas of human life; one of them is the field of medicine, where the authors of various studies are trying to help doctors in their work by creating intelligent tools to help doctors diagnose multiple diseases. Annotated data plays a crucial role in training deep neural networks, yet acquiring it can be particularly challenging, especially in domains like healthcare where data may be scarce or costly to procure [3]. This entire process is not only arduous and time-consuming but also prone to errors, significantly affecting patient outcomes. Moreover, engaging domain experts in the annotation process can incur substantial expenses. Thus, annotation tools with excellent usability represent invaluable assets that streamline the process and optimize the efficiency of domain experts.

QuPath [7], ASAP [9], Orbit [6], or Cytomine [5] are annotation tools commonly employed for annotating histological images. Feedback from our domain experts indicates that the majority of these tools, particularly QuPath, possess a low learning curve and need training for proficient use.

In this work, we introduce **AnnotAid, a user-friendly annotation tool that utilizes deep learning methods** to facilitate the annotation creation process and **to support the diagnosis of Nottingham Grading System (NGS) criteria in breast cancer, which are jointly developed along with our annotation tool.** It was developed in close collaboration with medical domain experts. To create AnnotAid, we introduced a novel communication concept involving domain experts, UX experts, and AI experts. This concept draws inspiration from the methods and principles of User-Centered Design (UCD), aiming to achieve an optimal User Experience (UX) while adhering to Human-Computer Interaction (HCI) principles consistently. The annotation tool serves **to streamline communication between domain experts and development teams**.

The paper is organized as follows: Section 2 presents an overview of existing annotation tools and approaches for

Nottingham Grading System (NGS) evaluation. In Section 3, we delve into the architecture, user interface, and functionality of the developed annotation tool. Preliminary results are provided in Section 4. Finally, Section 5 outlines the conclusions drawn from our work and discusses avenues for future research.

2 Related Work

In section 2.1 the approaches to solve each criterion of NGS are reviewed, and section 2.2 provides an overview of existing annotation tools.

2.1 Nottingham Grading System

Nottingham Grading System (NGS) [8] is a modified version of the Bloom & Richardson method used for grading breast cancer. This modification aims to introduce more objectivity into the criteria. The NGS involves a semiquantitative evaluation of three morphological criteria: **nuclear pleomorphism (NP)**, **mitotic count (MC)**, and **tubular formation (TF)**. Each criterion is assigned a score from 1 to 3, resulting in a final score ranging from 3 to 9.

2.1.1 Nuclear Pleomorphism

Xu et al. [21] proposed a deep-learning framework for nuclear atypia scoring, consisting of two stages: epithelial and stromal segmentation model and nuclear atypia-scoring models (x10, x20, x40 magnification). In the first stage, the relevant areas are segmented from images (epithelial and stromal), from which smaller patches are extracted and classified into 1-3 classes. For each magnification, the nuclear atypia score is determined with majority voting and then the final score is determined with plurality voting among all magnifications. On the other hand, Mathew et al. [15] proposed a framework for the extraction and classification of individual cells into nuclear atypia scores classified with the DenseNet121 model. The main idea of the proposed framework is to redesign the three-class problem (score 1-3) of slide image classification as a six-class problem (score 1-3, lymphocytes, necrotic cells, stroma cells) on nuclei classification. The final atypia score is assigned after aggregation of the nuclei classification results via plurality voting. The authors argued that the problem reformulation as a six-class problem and no four-class problem helped to increase performance. Sreeraj M. et al. [13] and Mercan et al. [16] used YOLO and RetinaNet detection models respectively, to detect and classify individual cells or patches into nuclear atypia scores.

2.1.2 Mitotic Count

Wang et al. [20] proposed a deep learning solution named FMDet, designed for the detection of mitotic cells. The authors tackled the problem as a segmentation task, where SE-ResNeXt50 encoder and an SK-based decoder were used. To address the domain shift problem, the authors proposed Fourier-based data augmentation where the low-frequency spectrum of the source domain is replaced by the low-frequency spectrum of the target domain. Jahanifar et al. [10] and Venugopal et al. [19] adopted a two-stage approach where the initial model detected cell candidates and a subsequent model classified them as mitotic or non-mitotic cells.

2.1.3 Tubular Formation

During our in-depth analysis of tubular formation, we identified only one approach by using segmentation models. There is a big scarcity of relevant papers addressing the tubular formation problem. Tekin et al. [18] proposed a deep learning framework designed for tubule segmentation. The paper introduces a novel in-house dataset comprising 51 Whole-Slide images (WSI). The authors employed reflection padding to tackle the challenge of incomplete tubules within patches. EfficientNetB3 demonstrated superior segmentation results, achieving a dice score of 95.33%.

2.2 Annotation Tools for Histology Images

LindvaN et al. [12] presents a comparative study between manual annotation and TissueWand, revealing a significant increase in annotation speed with the tool. TissueWand, designed for histopathological sample annotation, garnered preference from pathologists for its improved user experience. **The research methodology comprised user observations, prototyping, and interviews** to achieve a balance of manual control and automatic support, enhancing feedback speed and annotation assistance.

Several tools have emerged in the field of bioimage analysis, of which **QuPath** [7] stands out. This open-source desktop software is widely used in digital pathology for its ability to retrieve and navigate large, high-resolution whole slide images (WSI), along with its versatile annotation tools complemented by a range of available plug-ins. Another tool is **ASAP** [9] (Automatic Slide Analysis Platform), which stands out for its speed of image analysis. It provides users with tools to calculate area, perimeter, and other morphological measurements of annotated structures; **Orbit** [6] is a multi-platform open-source tool that can perform various image analysis algorithms from Orbit or other platforms. It facilitates real-time collaborative annotation, allowing multiple users to work simultaneously on the same image; **Cytomine** [5], an open-source RESTful web platform, operates via Docker containers and emphasizes remote collaboration. It facilitates data model organization, semantic annotation of high-resolution images, and image quantification via machine learning algorithms.

MONAI [4] is an open-source framework tailored for healthcare deep learning, leveraging PyTorch. It comprises three main components: MONAI core, MONAI label, and MONAI deploy. This framework facilitates integration with other annotation tools, such as QuPath, through a plugin architecture. **Our goal is to provide users with instant visualization of the result using AI, and MONAI is a framework that supports such functionality.**

3 Our Proposed Annotation Tool

Section 3.1 outlines the communication concept design within our team, while Section 3.2 elaborates on the system architecture. The user interface is detailed in Section 3.3, and available annotation methods are described in Section 3.4.

3.1 Proposed Concept of Communication

In Figure 1, **we present our proposal for the design of communication concept within our team**, where the processes and procedures of the individual actors are also included. Our team comprises a **domain expert**, a **UX expert**, and an **AI expert**. The process involves two main stages: the first focuses on specifying and understanding the requirements, while the second evaluates the design and implementation against these requirements. In the subsequent paragraphs, we will outline the responsibilities of each team member.

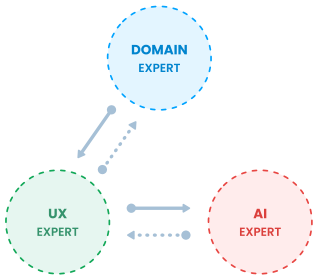


Figure 1: Concept of Communication

(1) Domain Expert: The domain experts **take the lead in specifying the requirements and functionalities** of the annotation tool under development. Throughout the development process, the UX expert collaborates with them to discuss each requirement and ensure their validation. Additionally, the domain expert provides their expertise and actively engages in data annotation, which is essential for training and enhancing the artificial intelligence methods employed by the annotation tool. In general, the domain expert serves as a potential user of the annotation tool.

(2) UX Expert: The UX Expert is **responsible for communicating with the domain expert to gain clarity on the specified requirements**. When necessary, they collaborate with the AI expert to define the requirements precisely and validate their fulfillment. Their primary objective is to incorporate the domain expert's requirements effectively into the annotation tool and guide the AI expert based on those needs.

(3) AI Expert: The AI expert is tasked with **analyzing and implementing the requirements** put forth by the UX expert. They focus on developing deep learning methods that support the annotation process.

3.2 System Architecture

The architecture diagram 2 delineates two main parts, where the implementation of the blue part is the responsibility of the UX expert and the implementation of the grey part is the responsibility of the AI expert. **The goal of the UX expert is to create the interface of the tool and integrate the AI API created by the AI expert**. This architectural view highlights the interaction and communication between the different components of the system. One of the main components of this architecture that we can highlight is the local server to support the reading and manipulation of WSI, which acts as an application module. The components of the system will be introduced in the paragraphs below.

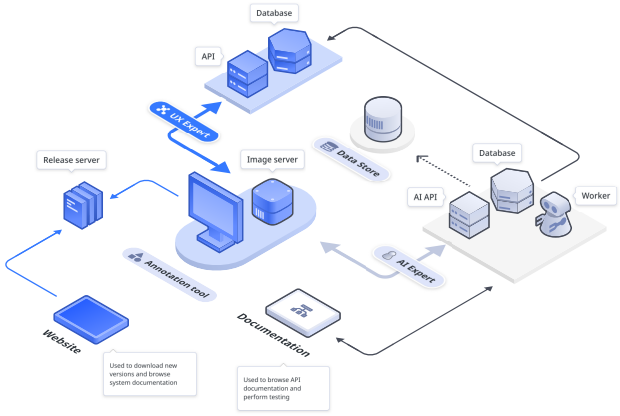


Figure 2: System Architecture

3.2.1 AI API

The AI API is developed using the FastAPI framework, facilitating communication between the annotation tool and the AI API through a REST API. The overall AI API architecture comprises distinct components, as illustrated in Figure 3: the **FastAPI backend**, **Redis message broker**, and **Celery worker**. Each component is encapsulated within a separate Docker container.

FastAPI[1] is a Python framework used to build modern and fast APIs. The FastAPI API utilizes an annotation

[1] fastapi.tiangolo.com

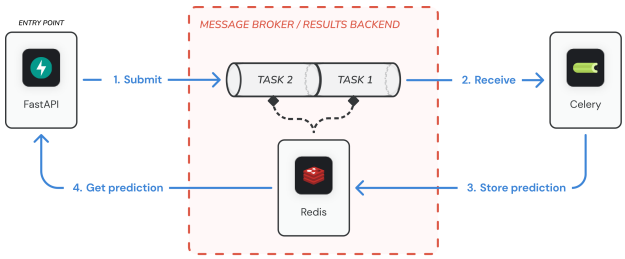


Figure 3: AI API Architecture

engine to make predictions using deep machine-learning models. **Upon receiving a request, the backend inserts the request into the message queue of the broker.** Subsequently, the client receives a unique task ID and uses the polling technique to query the task status and obtain the prediction result. The polling technique involves the client querying at regular intervals (e.g., every second) for the task result. If the task is completed, the client receives the prediction result; otherwise, they are notified to continue waiting. This architectural choice enables the asynchronous execution of a large number of prediction requests, treating them as background jobs to avoid blocking the server thread.

Redis message broker, acting as an intermediary and implemented as a Redis database, facilitates communication between the triggering application and the worker processes. **FastAPI inserts prediction tasks into this queue, and the Celery worker picks up and executes these tasks based on its workload.** The results are stored in the Redis database, allowing FastAPI to retrieve and provide them to the client.

Celery is an open-source distributed task queue system for Python that allows you to run tasks asynchronously. It offers high availability and easy horizontal scaling. The Celery worker continuously checks the queue for pending tasks. When a task is identified, it is executed by the Celery worker, which houses all the deep learning models necessary for task execution. The outcome of the task is written to the Redis database for further retrieval by FastAPI.

3.2.2 Annotation App

The system is primarily centered around an annotation tool, which is presented as a **cross-platform desktop application**. This tool connects to both a backend and third-party services to provide additional functionality. The backend consists of an API and a database, which manage user data, tool settings, and annotations. This setup promotes portability and facilitates modifications.

Installation files for various platforms are available on the annotation tool's website. Additionally, there is a release server that offers automatic updates for the annotation tool.

At the core of the system lies a local image server, which is initiated as a child process when the annotation tool launches. It is compiled into an executable file for each supported operating system and operates without requiring additional support software to be installed. During the tool's build process, this executable is integrated into the resulting installation files, tailored to the selected target platform.

(1) Annotation Tool: The foundation of the system is constructed using the Electron framework, operating on the NodeJS runtime. This setup enables the development of cross-platform desktop applications, utilizing the Vite[2] tool for efficient building processes. Moreover, the complete implementation is crafted in TypeScript, employing the React library for user interface design. This architecture is further enhanced by the integration of various libraries including: **OpenSeadragon**[3]: Used as a high-resolution zoomable image viewer; **Annotorious**[4]: Functioning as an extension for the OpenSeadragon library, this tool facilitates image annotation through drawing, commenting, or labeling. It supports a wide range of plugins and offers a high degree of customizability.

(2) Local Image Server: The solution is developed in Java, primarily due to the requirement of the **Bio-formats**[5] library, which is essential for reading and writing various life sciences image file formats. Additionally, an HTTP Server, crafted using the HttpServer library, facilitates communication with the application.

(3) Website: The website for the tool is developed using the Next.js framework, leveraging the React library and TypeScript for crafting a user interface. In addition to other libraries, the inclusion of the Stitches library simplifies the styling process.

(4) Release Server: The release server connects to GitHub to provide the latest versions and includes an interface for checking updates and downloading them. It's a modified version of the Hazel[6] update server for Electron apps.

The tool integrates with an AI API via HTTP requests, facilitating seamless communication. This architecture ensures modularity that makes it easy to modify or add additional methods, allowing independent development and offloading resource-intensive tasks to a robust server. **The API offers two types of methods: "instant" and "process".** Instant methods yield immediate results, ideal for operations requiring quick responses. On the other hand, process methods involve user-triggered actions, initiating

[2] electron-vite.org
[3] openseadragon.github.io
[4] annotorious.github.io
[5] openmicroscopy.org/bio-formats
[6] github.com/vercel/hazel

a sequence where necessary information is gathered, including image cropping. This data is then sent for analysis and queued for processing. Periodically, the tool checks for result availability, upon reception, the information undergoes parsing, storage, and display tailored to the specific process type. This communication style is selected to accommodate the longer processing time associated with the input of these methods.

3.3 User Interface

In the first step, wireframes were created to define the layout of various interface elements. These wireframes underwent consultation with domain experts, and after integrating their feedback, a high-fidelity prototype was created. Our design process embraced **rapid prototyping by directly implementing tool functionalities**, as our project timeline made it impractical to conduct separate prototype testing in Figma and subsequent implementation phases. The layout of this screen as well as other settings of the tool, which include changing the language, can be modified according to the user's preferences.

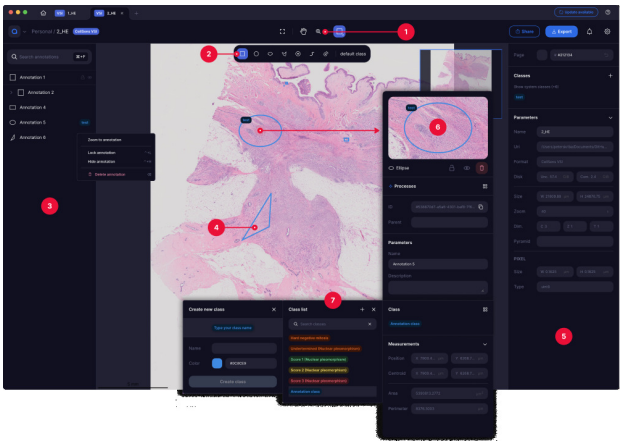


Figure 4: User Interface

Figure 4 shows **7 key parts and functionalities** of the annotation tool, the intent and focus of which we will discuss in the following sections:

(1) Toolbar: The Toolbar serves as the primary control hub for the tool, offering functions to manipulate images such as "Zoom to fit", a default "Hand" tool for navigation and annotation selection, and a "Zoom" tool for adjusting image magnification. The "Zoom" tool primarily utilizes the mouse wheel for control. Additionally, there's a tool to toggle annotation mode, facilitating seamless switching between browsing and annotation modes. Users can swiftly transition between annotation and navigation modes by holding down a key.

(2) Annotation Tools: When the annotation tool is selected from the toolbar, the interface switches to annotation mode, displaying all available tools along with the default class automatically assigned to newly created annotations. Depending on the selected tool, the cursor's appearance changes to reflect the tool icon. Keyboard shortcuts facilitate seamless tool switching.

(3) Annotation List: This panel primarily displays and allows searching of annotations, comprising a text input field for filtering annotations shown in a hierarchical tree structure. Each annotation includes basic information such as name, shape, and associated class. The tree structure also groups annotations based on visual hierarchy. Positioned on the left, it adheres to the principle of maintaining a familiar design for users.

(4) Annotation: Annotations appear as bounded shapes that become editable when clicked, enabling users to adjust the shape or position using handles. When hovered over, the border and shape colors reflect the selected class color for improved identification. The default border color is blue, chosen for its visibility in this image type.

(5) Image Properties: The right panel adapts dynamically based on selected functionality, annotation, or user state. Initially, it displays image information and workspace settings. Its placement on the right signifies it as an additional section linked to actions on the left side of the screen.

(6) Annotation Properties: Upon annotation creation or selection, the right bar transforms into an annotation detail panel, presenting information, parameters, previews, and additional functionalities. **Key parameters include the annotation name and description, facilitating communication between domain experts and the development team.** Automatic calculations such as position, size, or area are shown at the bottom of the panel. An annotation preview aids in identification and illustrates potential changes users can make within the panel.

(7) Annotation classes: The tool offers predefined classes for annotation categorization, with users able to create custom classes, each with a unique name and color. These user-defined classes are stored within the image settings for future reference and editing. Assigned classes are indicated on each annotation's top left corner, matched with corresponding colors for easy identification. When exporting annotations, class definitions are included, enabling the transfer of both annotations and their associated class information between projects. **This feature enhances utility, ensuring consistency and facilitating collaborative work across images.**

3.4 Annotation Methods

To streamline and enhance the annotation process of histological images, we have devised various approaches tailored to specific requirements and observations. These methods are categorized into two groups, delineated by user complexity and the degree of artificial intelligence assistance, to **manual** and **(semi)automated** which contains specialized subsection targeting the use case of **Nottingham Grading System**.

3.4.1 Manual Methods

Manual approaches can be defined as those that **do not use any form of AI assistance** in the process of creating the annotation. This means that the user is responsible for the overall result. In most cases, these forms of annotation must be performed before running the automated annotation method as a way of specifying the domain or input needed for the automated methods.

These manual methods include: **Rectangle annotation:** A basic shape to define boundaries or regions of interest (ROI), is most commonly used in defining regions for automated methods; **Circle annotation:** Circular annotation with the possibility to change the radius of the circle; **Ellipse annotation:** Similar to circular annotation with the possibility of changing two radii; **Polygon annotation:** Annotation with the possibility of adding more points, which results in a closed shape. It is used as a result of several automated annotation methods; **Free-hand annotation:** Freehand annotation that creates open shapes without points; **Point annotation:** An annotation point that indicates coordinates without the possibility of defining shape or size. Also used in defining the click position for automated methods.

3.4.2 (Semi)Automated Methods

Automated or semi-automated annotation methods (Fig. 5) can be defined as annotations where **only minimal user input is required** to specify the domain or interaction that is used as input for the AI models. Once the image and the specified input parameters have been analyzed, the result is returned in the form of a specified class (after classification), a modified annotation (as a polygon, for segmentation), or multiple annotations created (in the form of bounding boxes or polygons) with possible classification into multiple classes. These methods include:

(1) Nuclei Segmentation: NuClick [1] model is used for single-click cell segmentation. We acquired the model weights from the nuclick_torch[7] repository. The prediction outcome is a segmentation map which is subsequently adjusted using various postprocessing techniques aimed at enhancing the segmentation mask's quality. Postprocessing methods include removing objects smaller than a specified threshold, filling empty holes, and reconstruction. The refined segmentation mask is then converted into a polygon of points and transmitted to the annotation tool.

(2) Bbox Nuclei Segmentation: Nuclei segmentation from Bbox is a similar method to Nuclei Segmentation. It is a modified method where the user can get a more accurate annotation from the nuclei boundary by segmenting the nuclei from the selected Bbox.

(3) Segment Anything: SAM [11] is employed in the annotation tool for interactive segmentation of structures beyond the scope of the previously specified models. This model facilitates segmentation using bounding boxes and foreground/background clicks. Foreground clicks mark the desired segmentation area, while background clicks exclude it. The vit_b variant of the model is used for prediction, obtained from the segment-anything[8] repository. The predicted segmentation mask is subsequently adjusted using various postprocessing techniques identical to the NuClick model.

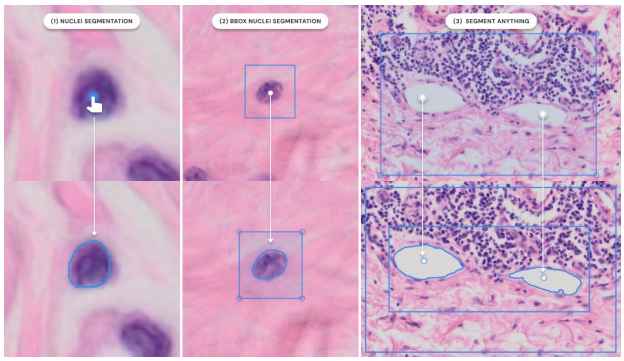


Figure 5: (Semi)Automated annotation methods

3.4.3 Use Case: Nottingham Grading System

Our objective was to develop methods that facilitate the scoring of the Nottingham grading system (Fig. 6). These methods have been incorporated into the automated annotation processes. These methods include:

(1) Mitosis Detection: Mitotic detection is employed to identify mitoses and hard-negative mitoses. The AI API utilizes a one-stage version comprising a YOLOv8 detector trained on the MIDOG++ dataset [2], see Table 1. The focus is solely on detecting mitoses and hard-negative mitoses, without evaluating the Nottingham Grading System's mitosis count criterion. The main reason behind this approach is rooted in the limitations of the developed annotation tool, which currently **cannot accurately reflect the**

[7] github.com/mostafajahanifar/nuclick_torch

[8] github.com/facebookresearch/segment-anything

real state of the score. This is due to the necessity of evaluating the score from an area equivalent to **10 High Power Fields (HPFs)** or approximately **2mm^2** [14]. We trained the model to detect hard-negative mitoses, aiming to provide domain experts with a definitive decision on whether a given instance is a mitosis or not. Its input is an ROI that delimits the area from which the analysis can give the user an annotated mitosis with an assigned class.

The most favorable outcomes were obtained with the medium variant of the pre-trained YOLOv8 model, developed by ultralytics[9], and employing FFTStain-Augmentation [20]. These results yielded an F1 score of 0.643, precision of 0.592, recall of 0.703, mAP50 of 0.664, and mAP of 0.476 on the test set.

Dataset	Mitotic figures	Hard-negative figures
train	17216	20944
val	862	961
test	2467	2916

Table 1: Number of mitotic and non-mitotic figures per set in FFT augmented MIDOG++ dataset

(2) Nuclear Pleomorphism: Another criterion for the Nottingham Grading System is Nuclear Pleomorphism. The EfficientNetB4 model aims to predict the nuclear atypia score based on images received by the AI API from the annotation tool. During image processing, the input is divided into patches, and a nuclear atypia score (1, 2, or 3) prediction is generated for each patch. The ultimate score is determined through a majority voting mechanism. This approach draws inspiration from the methodology outlined in the article [21]. MITOS-ATYPIA-2014 dataset [17] was used to train the model, see Table 2.

Dataset	Score 1	Score 2	Score 3
train	1941	13017	2150
val	88	2472	394
test	3078	4582	1747

Table 2: Number of patches per nuclear atypia score

The most favorable outcomes were obtained with the EfficientNetB4 model, pre-trained on the ImageNet dataset. These results yielded an F1 score of 0.349, accuracy of 0.511, precision of 0.480, recall of 0.382, and AUC of 0.547 on the test set. The results achieved are unsatisfactory, which forces us to look for improvements through annotations of our own data.

(3) Tubular Formation: We did not address this criterion because we could not find any public dataset.

[9]https://github.com/ultralytics/ultralytics

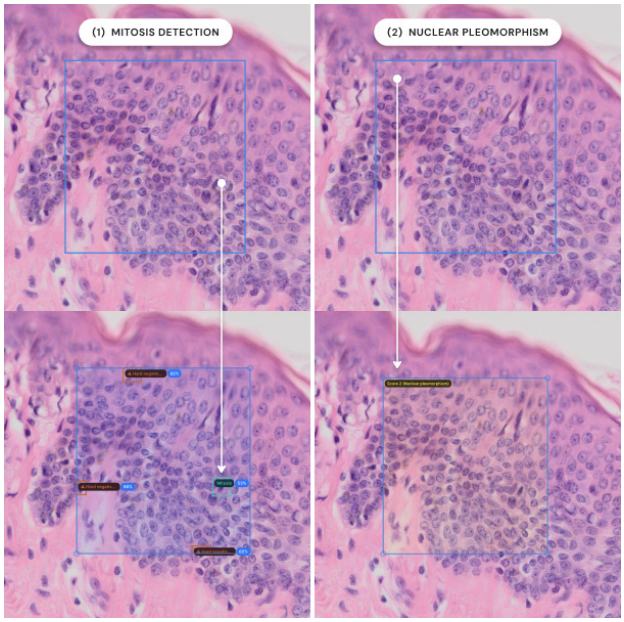


Figure 6: NGS Annotation methods

4 Results

Our preliminary user-testing with domain expert resulted in UI changes related to behaviour of the annotation tool based on user expectations and their feedback. Additionally, in near future we plan to conduct A/B testing of the AnnotAid and QuPath. The main goal of this user testing is to simulate real-word scenario of evaluation mitotic count criterion of Nottingham Grading System. We plan to compare efficiency of manual (in both tools) and semi-automated methods (in AnnotAid).

Based on the results obtained from developing models for evaluation individual criteria of the Nottingham Grading System, it is evident that there is a requirement for additional data and improvement of these models. **This can be accomplished through the utilization of the annotation tool we have developed.**

5 Conclusion and Future work

In our work, **our primary focus centered on developing an annotation tool** in close cooperation with domain experts. We conducted multiple user testing sessions where valuable feedback was incorporated into the functionality of the annotation tool. **Our research primarily revolves around the creation of deep learning methods to support the diagnosis of Nottingham Grading System criteria.**

Moving forward, we aim to **integrate active learning** into our annotation tool, prioritizing methods to improve model performance and user experience, alongside **enhancing model explainability**. Additionally, we plan to introduce **new annotation methods** and collaboration fea-

tures, informed by regular user tests to ensure ongoing refinement of the tool's functionality and usability, aligning with user expectations and preferences.

References

[1] Navid Alemi Koohbanani, Mostafa Jahanifar, Neda Zamani Tajadin, et al. Nuclick: A deep learning framework for interactive segmentation of microscopic images. *Medical Image Analysis*, 65:101771, 2020.

[2] Marc Aubreville, Frauke Wilm, Nikolas Stathonikos, et al. A comprehensive multi-domain dataset for mitotic figure detection. *Scientific Data*, 10(1):484, 7 2023.

[3] Sugata Banerji and Sushmita Mitra. Deep learning in histopathology: A review. *WIREs Data Mining and Knowledge Discovery*, 12(1):e1439, 2022.

[4] M. Jorge Cardoso, Wenqi Li, Richard Brown, et al. Monai: An open-source framework for deep learning in healthcare, 2022.

[5] Cytomine community. Introduction. `https://doc.cytomine.org/admin-guide/`, 2022. Accessed: 2022-12-09.

[6] Orbit docs authors. Tissue quantification. `https://www.orbit.bio/tissue-quantification/`, 2023. Accessed: 2023-05-22.

[7] QuPath docs authors. What is qupath? `https://qupath.readthedocs.io/en/stable/docs/intro/about.html`, 2022. Accessed: 2022-12-09.

[8] Christopher W. Elston and I. O. Ellis. pathological prognostic factors in breast cancer. i. the value of histological grade in breast cancer: experience from a large study with long-term follow-up. *Histopathology*, 19, 1991.

[9] Computational Pathology Group. Asap. `https://computationalpathologygroup.github.io/ASAP/`, 2018. Accessed: 2022-12-09.

[10] Mostafa Jahanifar, Adam Shephard, Neda Zamani Tajeddin, et al. Stain-robust mitotic figure detection for the mitosis domain generalization challenge, 2021.

[11] Alexander Kirillov, Eric Mintun, Nikhila Ravi, et al. Segment anything, 2023.

[12] Martin LindvaN, Alexander Sanner, Fredrik Petre, et al. Tissuewand, a rapid histopathology annotation tool. *Journal of Pathology Informatics*, 11(1):27, 2020.

[13] Sreeraj M. and Jestin Joy. A machine learning based framework for assisting pathologists in grading and counting of breast cancer cells. *ICT Express*, 7(4):440–444, 2021.

[14] Siddhartha Mantrala, Paula S Ginter, Aditya Mitkari, et al. Concordance in breast cancer grading by artificial intelligence on whole slide images compares with a multi-institutional cohort of breast pathologists. *Archives of pathology & laboratory medicine*, 146(11):1369–1377, 2022.

[15] Tojo Mathew, C.I. Johnpaul, B. Ajith, et al. A deep learning based classifier framework for automated nuclear atypia scoring of breast carcinoma. *Engineering Applications of Artificial Intelligence*, 120:105949, 2023.

[16] Caner Mercan, Maschenka Balkenhol, Roberto Salgado, et al. Deep learning for fully-automated nuclear pleomorphism scoring in breast cancer. *npj Breast Cancer*, 8(1):120, 11 2022.

[17] Daniel Racoceanu, Jessica Calvo, Elham Attieh, et al. Detection of mitosis and evaluation of nuclear atypia score in breast cancer histological images. 2014.

[18] Eren Tekin, Çisem Yazıcı, Huseyin Kusetogullari, et al. Tubule-u-net: a novel dataset and deep learning-based tubule segmentation framework in whole slide images of breast cancer. *Scientific Reports*, 13(1):128, 1 2023.

[19] Anjali Venugopal and Lekha S. Nair. Two-phase mitotic detection using deep learning techniques. In Milan Tuba, Shyam Akashe, and Amit Joshi, editors, *ICT Infrastructure and Computing*, pages 479–489, Singapore, 2023. Springer Nature Singapore.

[20] Xiyue Wang, Jun Zhang, Sen Yang, et al. A generalizable and robust deep learning algorithm for mitosis detection in multicenter breast histopathological images. *Medical Image Analysis*, 84:102703, 2023.

[21] Jun Xu, Chao Zhou, Bing Lang, and Qingshan Liu. *Deep Learning for Histopathological Image Analysis: Towards Computerized Diagnosis on Cancers*, pages 73–95. Springer International Publishing, Cham, 2017.

Virtual Reality

Cognitive Maps Acquisition by Those with Vision Impairments in Virtual Reality

Matyáš Koval'
Supervised by: Ing. Miroslav Macík, Ph.D.

Faculty of Electrical Engineering
Czech Technical University
Czech republic / Prague

Abstract

Individuals with vision impairments (VI) require specific methods to acquire spatial knowledge of the environment they need to orientate themselves. Such knowledge is called a cognitive map of the spatial environment and has multiple components (landmarks, distances, directions, routes, etc.). The performance of interaction methods varies in the acquisition of different components of spatial knowledge. Our research focuses on the employment of Virtual Reality adapted for VI as a novel method for acquiring cognitive maps. We leverage a combination of interaction modalities (vibrations, haptic feedback through a modified white cane, and in the future even spatial audio) to provide spatial knowledge of indoor environments.

Keywords: Virtual Reality, Haptics, Tactile, Spatial orientation, Visually impaired.

1 Introduction

Visually impaired (VI) individuals deal with more difficulties when exploring a new environment than users without visual impairment. Depending on the environment, it may take them more time and effort to orientate themselves, or in some cases, it may even be dangerous. The creation of even basic cognitive maps (CMs) beforehand may lead to a significant improvement during their first real experience with said environment.

We reflect this issue in our research question, which for this work is: If we implement a simple haptic feedback source via a white cane, is it enough information for a VI individual to create at least a rough CM in a safe and controlled environment?

To provide VI individuals with the option above, we have created virtual environments – scenes made in the Unity game development engine with the inclusion of Virtual Reality (VR) libraries needed. We have created so far two environments that, in one case, represent a singular room with basic boundaries and an obstacle, as can be seen in Figure 1. The other scene (shown in Figure 2) represents a more complex scene based on a real environment. It comprises a study room and a section of an adjacent corridor. There are more obstacles than in the basic scene. It contains both hallways, doors, and furniture and is accessible to us on demand. Modeling part of a real environment will allow for more complex evaluation based on the test procedures where it is involved.

These environments are then projected into a VR head-

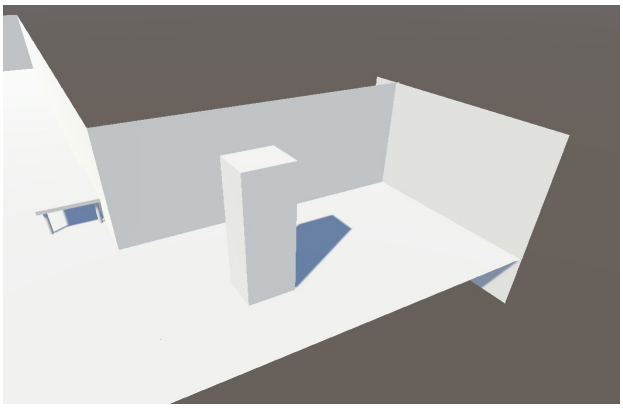


Figure 1: The virtual preliminary testing area - due to real space limitations, the participant was exploring only this part of the room

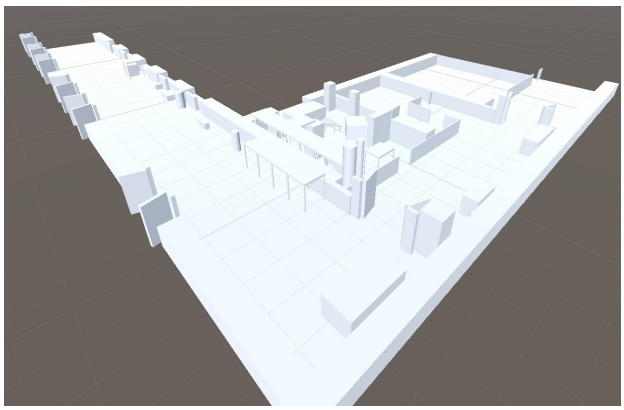


Figure 2: A screenshot of the Unity scene with the study room and halls

set, which the VI participant is wearing. The participant is also provided with a modified white cane, with which they can then explore the created environment.

The paper is structured as follows. In Section 2, we summarize the related work, including examples of methods that utilize VR for purposes of creation of CMs by VI individuals. In Section 3, we discuss the details of two developed prototypes that implement the aforementioned scenes. Section 4 presents only preliminary evaluation that, however, provides strong indications that even simple interaction methods can be utilized for the creation of a CM for VI participants in VR conditions. We describe the testing process for the preliminary evaluation as well as for the more advanced tests we have planned for the near future. Finally, in Section 5, we focus on the results of this work and evaluate the preliminary results.

2 Related work

This section focuses on the cognitive maps and means for their acquisition as an important contributing factor for efficient spatial orientation of VI individuals. Later, we list examples of methods that leverage VR for the VI.

Cognitive map refers to the internally represented model of a spatial environment [9], which contains knowledge of landmarks, route connections, distance and direction relations, and non-spatial attributes. *Cognitive maps* comprise more types of spatial knowledge: locations, layout, routes, distance, and directions between locations [7]. Two basic frames of reference related to spatial knowledge exist – allocentric (object-to-object) and egocentric (subject-to-object) [2]. Well-developed cognitive maps contribute to good spatial orientation and efficient navigation through both indoor and outdoor environments [3].

Interaction methods that employ different sensory modalities can contribute to the acquisition of cognitive maps. For sighted individuals, the natural method is a direct experience in the visited environment, but in many cases, different kinds of topographical maps are used (classical, digital, 2D, 3D) [4]. In some cases, Virtual Reality and Augmented Reality are useful to increase efficiency safety (training of movement in dangerous areas) or provide specific information that would be less accessible using other methods (i.e., the spatial position of electrical wires or plumbing) [5]. In the case of VI individuals, the situation is similar; however, they (VI individuals) have specific needs, abilities, and preferences. For them, it is more complicated to get information in the allocentric frame of reference. For this purpose, (interactive) tactile maps are usually employed [1].

The formation of *cognitive maps* is a challenging process for VI individuals as it requires substituting vision with other sensory modalities or their combination. Ottink et al. [7] provide a literature overview of methods for cognitive map acquisition based on non-visual modalities, with a particular focus on the auditory, haptic, and multimodal approach for the VI. The authors conclude that VI individuals can form *cognitive maps* using more sensory modalities or their combination. However, some modalities are better suited for building different types of spatial knowledge in *cognitive maps*. Navigational strategies that affect the formation of *cognitive maps* are the route and survey strategies. Survey strategies require map-like (allocentric) representations of the spatial environment in a *cognitive map* and are usually connected with a better orientation performance.

Kunz et al. [6] and Siu et al. [8] provide examples of approaches that utilize walkable VR for purposes of creation of *cognitive maps* for the VI.

Kunz et al. [6] focus on implementing and testing a purely auditory method for the navigation and orientation of non-VI blindfolded users, who then navigated a virtual maze based on the audio feedback that has been supplied to them via headphones. The main source of feedback — audio — is spatial, so the participant can change his movement according to where the obstacle is detected. From the results of this study, it is apparent that audio feedback by itself, while definitely providing enough information about the environment to improve the participant's awareness of their surroundings, is not enough to sufficiently improve the participant's orientation capabilities for it to be the only source of information about the environment. This is an important takeaway for our work, as while not being completely sufficient, the auditory feedback nevertheless improved the creation process of CMs.

The work of Siu et al. [8], however, is closest in both its aim and realization to this work. The authors developed and created a wearable harness connected with pulleys and motors to a physical white cane, which was then controlled accordingly by collisions with VR objects by the pulleys. This served mainly as an inspiration as to what the end goal of the work may be while focusing on a more straightforward and less complex solution in terms of the hardware (HW) used.

The aforementioned approaches are focused on a similar goal as this work, albeit they utilize slightly different means of implementation than what is done in this preliminary work or planned for future work (Section6). We utilize these works as points of reference, sources, and inspirations during the design and implementation of our work, and they also help us to orientate ourselves more in the area of interest and understand the issues that may arise during our own development.

3 Prototypes

As mentioned in Section 1, the whole setup for our work consists of a single Virtual Reality headset (Oculus Quest 2), one prototype white cane consisting of a VR controller coupled with an actual white cane, and two virtual environments, which can be switched to at will through the Unity editor on a computer connected to the headset.

The real-world prototype white cane is coupled with the controller via a 3D-printed holder, as shown in Figure 3, which affixes the controller near the handle of the white cane so as not to overly affect the balance, which would impact the overall handling. This placement with a direct fixation on the cane serves to transmit the controller vibrations directly into the cane, from where the participant can comfortably feel them. Also, the holder can be rotated to customize the placement of the controller according to the participant's preference.

Figure 3: Implementation of the real world cane with a controller coupled via a 3D-printed holder, mounted right at the end of the cane next to the handle

The participant wears the Virtual Reality headset, and even though the visual information it provides is redundant, it is necessary for the tracking of the participant's body and head in the virtual environment. This procedure is similar to the approach used in articles by Kunz et al. [6] or Siu et al. [8] – both of which served as the initial inspiration for this approach.

Where our approach differs is the implementation of the way we provide feedback to the participant about his surroundings. The aforementioned articles either had a custom-built harness with a white cane or only audio feedback. The virtual cane used in this preliminary state is still only an actual adjustable white cane coupled with a Virtual Reality controller, and the haptic feedback is provided via the vibrations of the controller and its intensity. This is, at the current state, a limited fidelity haptic feedback that can be provided by the available hardware. It is important for this and the following works so that we can determine if even this is enough for the creation of cognitive maps and then follow up on it by adding more and better feedback to the participant.

The vibrations themselves are set up so they trigger during the contact of the virtual white cane with any obstacle - be it a wall or a piece of furniture - excluding the floor. The floor is excluded, as the cane is supposed to be in contact with it during the whole test. Therefore, it would provide no additional information during these tests. The interaction method also leverages a variable intensity of the vibrations. The vibrations are set up in a way that they get more intense and faster as the participant pushes the cane deeper into objects. This is supposed to provide the participant with stronger feedback in case they miss the initial vibrations or get so far into an object that they are unsure which way to go into open space.

The last part of the physical setup, the computer, is optional since the virtual environments may be compiled and entirely run inside the specific headset we are using - that being Oculus Quest 2, as mentioned above. For our preliminary test, however, it is still necessary to fine-tune and calibrate the virtual cane and the position of the participant in the virtual environment, so at this stage, we can not omit a computer as a part of the setup.

The virtual part of our setup includes the two virtual environments running in the Unity game development engine and its associated editor. The first scene, as seen in Figure 1, serves mostly as a proof of concept with the main purpose being to check the validity of our methods and whether they are at all suitable for the most basic of obstacles, such as walls and bigger obstacles with uncomplicated bounding boxes.

The second environment, as seen in Figure 2, is a virtualization of a real-world study room (as shown in Figure 4 and Figure 5) and in short, it is a square room containing multiple obstacles such as chairs, tables, counters and shelves, with the associated hallways being obstructed by plants, slight nooks in walls, again desks and chairs along with some other obstacles as well (fire extinguishers etc.). This is a more intermediate environment, containing many obstacles, and should be a little harder to navigate. And since it is a virtual copy of a real-world environment, the participant who will test this room has the option of exploring the real-world counterpart as well, so we could evaluate whether the cognitive map he or she has created during the virtual exploration has helped in any way.

4 Evaluation

This section focuses on experiments to evaluate the utility and usability of our method. The primary aim is to answer our research question, whether or not our current implementation of haptic feedback is enough for a VI participant to create at least a rough CM in a safe and controlled environment.

4.1 Preliminary Test

The implemented prototype used for the preliminary evaluation does not employ multi-modal interaction. It uses only haptic feedback and is focused primarily on an egocentric orientation in a virtual environment.

4.1.1 Procedure

The testing that has been done so far has a clearly defined procedure that will be adhered to during the testing if pos-

Figure 4: A comparison of the real and virtualized study room

Figure 5: A comparison of the real hallway and the virtualized version of the view from approximately the same spot

sible. The procedure is as follows:

Preparation: The participant will be familiarized with how the VR setup works, how they will use it, and what he or she should expect going into the testing. This will prepare the participant for the actual testing phase and should limit any unnecessary confusion that may arise from the possible inexperience with VR.

Calibration: The participant will stand in one place, will put on the VR headset, and will be handed the real-world white cane with an attached VR controller. The participant will then point the cane straight down and touch the floor with the tip of it. Then, the supervisor of the test will adjust the size and orientation of the cane in the VR application so it corresponds with the real-world placement.

Test walk-trough: This phase is self-explanatory. The participant will have the option to explore the VR environment using the provided HW. In the beginning phases, this will be without specific goals to check the whole proof of concept. In later stages, this will include objectives, such as finding specific objects or navigating to a specific place.

Feedback gathering: This will be the last phase, during which the participant will describe his experience with the application and provide feedback.

4.1.2 Measures:

During the experiment, we focused on subjective qualitative feedback (obstacle and boundary detection, the usability of the interaction method) and the ability of the participant to describe the explored virtual environment.

4.1.3 Participants

One participant with vision impairment was involved in the experiment. He has no previous experience with workable Virtual Reality based on wearable devices. He is male, has been late blind for more than 20 years, and is 40–50 years old.

4.1.4 Test setup and execution

For the preliminary evaluation, we used a simple virtual environment – a room with a nook as depicted in Figure 1. The experiment was conducted in a room with an available empty space 2.5×2.5 meters. The participant used a prototype white cane with the attached controller, as depicted in Figure 3. Two members of the project team were present to ensure the participant's safety (avoid possible collisions with objects in the real environment).

The preliminary testing followed the procedure described above, with the repetition of the test walk-through.

One instance has adhered strictly to the description above, so the participant explored the virtual environment without additional interference from the supervisor's side using the cane. The second instance of the testing was done on demand by the participant, as he wanted to explore the environment more. He has now included a discussion with the supervisors about the mechanics and features of the prototype. We will discuss the mentioned feedback along with our observations in Section 5.

4.1.5 Results

After we received the participant's feedback, the main takeaway points were these:

- The main goal – orientation in a room is possible, as even in the current prototype, implementing only vibrations of various intensities, the participant was able to use them to quickly and efficiently find his bearing. He was able to find walls and navigate along them without much of a problem, even finding obstacles. The participant, however, perceived different dimensions of the obstacles he found, so much so that he determined a narrow space between the pillar seen in Figure 1 to be too narrow to move through. This may be a result of inaccuracies in calibration, but for future work, it may need to be accounted for.
- The lack of feedback for the participant being in an object or obstacle sometimes caused problems, as the participant's virtual body has no collision detection in place, and the participant can step outside of the current boundaries, which then severely complicates navigation and will end up needing intervention from a supervisor.
- What seemed to be a problem, in general, was the perception of the vibrations caused by the room having a carpet. Even this very slight roughness of the ground sometimes caused the participant not to be able to feel the vibrations, and after a while, he resorted to using the cane raised slightly in the air to counteract this.
- An unexpected discovery was the fact that the participants can and will use audio queues in the real world to center themselves in the virtual environment, as static audio sources can be used as an anchor of sorts.

4.2 Evaluation of complex environment

In this section, we describe the planned evaluation of the prototype that will comprise the complex environment as depicted in Figure 6.

4.2.1 Procedure

The test procedure regarding the VR setup will be similar to the preliminary evaluation with further differences. However, the advanced prototype, as described in this section, resembles a real environment (as depicted in Figure 6) and allows for the creation of a simple 3D printed tactile map as depicted in Figure 7. This evaluation will be done mainly to explore the boundaries of how far we can go with just basic haptic feedback in a more complicated and cluttered virtual environment. Furthermore, during this testing, the users will have a clear goal – that is to navigate into the study room, with a start in the hallways outside of it. Once in the room, they should explore it and be able to describe the layout of the room at least approximately – they will probably not be able to differentiate between objects themselves, but what is the main goal is to be able to determine obstacles in general and their rough placements. During the feedback-gathering phases of this evaluation, the users will have either the tactile map (Figure 7) or the real-world environment at their disposal – we plan to utilize both.

Figure 6: A photo of the real world environment around the study room area

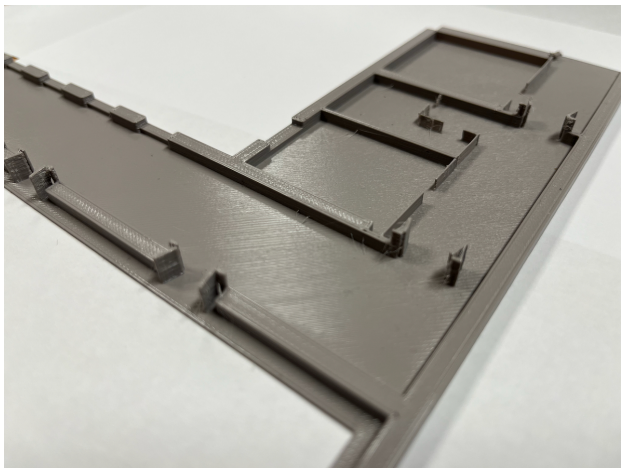


Figure 7: A photo of a tactile map printed according to the virtual environment seen in Figure 2

The preparation and calibration phases will be similar to the preliminary evaluation.

Test walk-trough: As mentioned at the beginning of this subsection, the procedure during this evaluation will differ both in the environment the users will be exploring and the goal. The users will begin in the empty hallways connected to the study room, with the goal of navigating to it. For this purpose, they will be given a rough verbal description of where the room is supposed to be (e.g., at the end of this hallway, there is a door, on the left side, go through it and a few meters after the door on your left, you should expect the entry to the room). After they have successfully navigated to the study room, they will now begin the free exploration of the room with the goal of remembering the layout and creating a CM of it and preferably of the environment around it.

Feedback gathering: This will be the last phase, during which the participant will both describe his experience with the application and provide feedback verbally, but also will be asked to describe or show the landmarks, obstacles, and objects encountered during their exploration, along with the path they took.

4.2.2 Measures

As in the preliminary evaluation, we will focus on subjective qualitative feedback. Moreover, we will evaluate the quality of the acquired cognitive maps by requesting the participants to:

- Show/describe the position of objects and landmarks encountered in the VR using the tactile map.
- Show/describe the position of objects and landmarks encountered in the VR using the real environment.

4.2.3 Participants

We plan to recruit six participants with vision impairment. The inclusion criterion is that they are capable of independent orientation in simple indoor environments other than their own flat (i.e. workplace, nearby convenience store, etc.). We plan to sample the audience by selecting at least two congenitally blind and two late blind participants.

5 Discussion

The preliminary evaluation indicates that even a simple interaction method based on tactile feedback provided by vibrations allows for the creation of CMs. This is alongside the results of Kunz et al. [6] a somewhat expected, but nonetheless significant result, as it proves that even basic tactile feedback is enough of a foundation that can be built upon with further enhancements with auditory feedback and further refining of tactile feedback.

However, there were also drawbacks discovered that were not observed by Kunz et al. [6] or Siu et al. [8] as they were exclusive to our testing environment and implementation. There were difficulties with calibration, where a white cane – if set up in such a way that its length does not correspond to the height of the VR participant perfectly, will go through the floor. Therefore, it will interact with obstacle hitboxes/boundary boxes under the floor if they are present or will not collide with the tip of the white cane but with its body. This causes slight but perceivable changes in obstacle placement and, therefore, provides spatial information different from what the virtual environment depicts.

Another problem was, as mentioned in the previous Section 4, the floor surface of the real-world testing environment. In our case, it was covered by a carpet, which caused vibrations in the white cane, interfering with the tactile feedback from the vibrations of the controller. This is of great importance for future testing, as flooring with as smooth a surface as possible will be needed. But once again, the participant was able to explore the environment even with this disadvantage, which only further confirms our conclusions on the viability of this feedback method.

The last takeaway for discussion is actually not much of a problem and has been mentioned by Siu et al. [8] as well. This takeaway is the mechanic of an auditory *anchor* of sorts. Siu et al. [8] utilize virtual audio sources as checkpoints through which the users travel and which help them to put their surroundings into perspective and center themselves around them and in relation to them. We unintentionally provided the participant with a real-world auditory anchor in the form of the computer, through which the virtual environment was running, which had noisy ventilators and, as such, provided the participant with a point of reference he then automatically used for orientation. This is a feature that we plan on using in the future, most probably in the form of a virtual auditory source along with headphones for the participant so as to filter out outside influences.

6 Conclusions and Future work

In this paper, we described the results of a project that aims to employ VR for the purpose of the creation of CMs. These CM are used for the improvement of the spatial orientation of those with visual impairments in indoor settings. Our preliminary results show that VR is a promising method to achieve this goal.

It is the subject of future work to evaluate the advanced prototype as described in Section 4.2. Along with this, we will construct more complicated virtual environments that will incorporate different goals and exploration methods. We also plan on enhancing our current form of haptic feedback by itself with proprietary hardware and further white cane prototypes, along with adding auditory feedback in a few different forms (auditory *anchor* and also obstacle or collision detection). This will also warrant further evaluation in relation to our results gathered so far. In the end,

we will further explore how to combine different methods for the acquisition of CM to achieve optimal results for specific environments and individuals with specific needs, abilities, and preferences.

References

[1] Anke M Brock, Philippe Truillet, Bernard Oriola, Delphine Picard, and Christophe Jouffrais. Interactivity improves usability of geographic maps for visually impaired people. *Human–Computer Interaction*, 30(2):156–194, 2015.

[2] Neil Burgess. Spatial cognition and the brain. *Annals of the New York Academy of Sciences*, 1124(1):77–97, 2008.

[3] Reginald G Golledge, R Daniel Jacobson, Robert Kitchin, and Mark Blades. Cognitive maps, spatial abilities, and human wayfinding. *Geographical Review of Japan, Series B.*, 73(2):93–104, 2000.

[4] Hongyun Guo, Nai Yang, Zhong Wang, and Hao Fang. Effects of spatial reference frames, map dimensionality, and navigation modes on spatial orientation efficiency. *ISPRS International Journal of Geo-Information*, 12(12):476, 2023.

[5] Lasse H Hansen, Philipp Fleck, Marco Stranner, Dieter Schmalstieg, and Clemens Arth. Augmented reality for subsurface utility engineering, revisited. *IEEE Transactions on Visualization and Computer Graphics*, 27(11):4119–4128, 2021.

[6] Andreas Kunz, Klaus Miesenberger, Limin Zeng, and Gerhard Weber. Virtual navigation environment for blind and low vision people. In *International Conference on Computers Helping People with Special Needs*, pages 114–122. Springer, 2018.

[7] Loes Ottink, Hendrik Buimer, Bram van Raalte, Christian F Doeller, Thea M van der Geest, and Richard JA van Wezel. Cognitive map formation supported by auditory, haptic, and multimodal information in persons with blindness. *Neuroscience & Biobehavioral Reviews*, 140(104797), 2022.

[8] Alexa F Siu, Mike Sinclair, Robert Kovacs, Eyal Ofek, Christian Holz, and Edward Cutrell. Virtual reality without vision: A haptic and auditory white cane to navigate complex virtual worlds. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–13, 2020.

[9] Edward C Tolman. Cognitive maps in rats and men. *Psychological review*, 55(4):189, 1948.

VR Therapia: Utilizing Immersive Virtual Reality for Applied Psychology Interventions

Yulian Rusyn *
Supervised by: Zuzana Berger Haladová†

Abstract

Virtual reality (VR) holds the potential to positively impact mental well-being by transporting individuals to serene environments, such as a calming forest or a contrasting cityscape plagued by urban challenges. This study investigates the therapeutic effectiveness of VR experiences utilizing Head-Mounted Display (HMD), body position tracking and heart rate tracking.

By implementing full-body tracking users transition seamlessly to contrasting environments — a tranquil "Forest" with animals, the sound of wind, or a challenging "City" with rats, trash, and the cacophony of urban noise.

Our preliminary findings indicate a significant difference in relaxation levels between the "Forest" and "City" scenarios, highlighting the potential of VR to elicit distinct emotional responses. The incorporation of heart rate monitoring, emerged as a valuable component for estimation of the stress level of the participants. This research not only underscores the potential of VR applications in promoting relaxation but also contributes to a nuanced understanding of the emotional impact of contrasting virtual environments.

Keywords: Forest therapy, Full body tracking, immersive environments, Virtual reality

1 Introduction

In recent years, virtual environments have garnered increased attention as a promising tool for mental health intervention. Inspired by the calming effects of Japanese Forest Therapy, our paper aims to explore the efficacy of virtual environments in enhancing psychological well-being. Specifically, we focus on comparing the impact of virtual forests and virtual cities on key variables such as self-compassion, self-care, self-criticism, and stress levels.

Moreover, our study delves into the comparison between 360-degree videos depicting forests and cities and fully immersive 3D scenes. This broader exploration allows us to assess the nuanced differences in therapeutic outcomes between these modalities, contributing valuable insights to the evolving landscape of virtual interventions in mental health.

*rusyn1@uniba.sk
†haladova@fmph.uniba.sk

Drawing from the therapeutic potential of nature-based interventions, particularly Shinrin-yoku[3] principles, our research plan involves developing an application to assess the broad applicability and therapeutic value of virtual environments in non-clinical and clinical populations. This work is done in collaboration with applied psychology researchers from Faculty of Social and Economic Sciences Comenius University Bratislava. Its aim is to understand whether a virtual forest, compared to a virtual city, can effectively promote key psychological variables.

To conduct our research, we will employ Virtual reality setup consisting of HMD and position trackers to connect participants to the virtual world and record their movements. Heart rate measurement devices was used to collect accurate data throughout the intervention, stored for detailed analysis.

The paper is organized as follows. In the next section the review of previous work is presented. In the third part, we will address various aspects of developing our application, including technologies, devices, and programs.

2 Previous Works

In the area of virtual reality forest therapy, several works were published.

In 2022, the study by Leung et al. [5] examining the impact of VR exposure on nature connection and affective states in individuals with low affinity for nature. Two studies were conducted to test this hypothesis. In the first study, participants experienced three VR sessions, each with a unique 360-degree video, while the second study included a control group exposed to a virtual urban environment. Results showed increased nature connection and positive affect in the virtual nature group. The second study aimed to replicate these findings with fewer and shorter sessions. Both studies utilized HTC Vive Pro headsets and statistical analyses for evaluation. Participants completed the intervention over two weeks, with sessions spaced approximately 3,9 days apart, in a controlled environment. The VR system presented nine ultra-high-resolution 360° VR videos in a fixed order, demonstrating positive impacts on nature connection and emotional well-being.

Second article by Wang et al.[10] explored the impact of various forest settings on stress levels through virtual reality (VR) videos. Seven forest recreation sites in Beijing

were tested, with stress levels monitored using physiological and psychological indicators. Participants aged 18-35 with good health were enrolled. Before the experiment, an introduction covered its purpose, process, risks, and confidentiality. Subjects' baseline heart rates were measured, followed by the Trier Social Stress Test (TSST), inducing stress through a public speech and mental counting task. Pre-tests measured blood pressure, heart rate, salivary amylase, and mood. Subjects then viewed a 5-minute VR video of a forest environment to potentially reduce stress. Post-tests mirrored pre-tests. The UCVR EYE-01 camera captured images in seven forests, and videos were recorded in 33 areas. The HEM-7111 electronic sphygmomanometer measured blood pressure and heart rate. Results revealed varying stress-alleviating effects among forest scenes, with aquatic environments notably reducing stress. This study contributes to understanding forest therapy use.

The next study by Chia-Pin Yu et al. [11] utilized the HTC Vive VR system. 360-degree videos recorded by researchers were played on the HMD in this study.

For the urban environment, researchers selected Ximending in Taipei, Taiwan, known as a shopping paradise. The recorded video in Ximending captured urban elements such as crowds, noise, traffic, and low greenery coverage. To showcase the forest environment, researchers filmed in the recreational area of Aowanda National Forest in Nantao City, Taiwan. This area features water protection zones, coniferous and deciduous trees, and diverse wildlife. The video presented natural elements of Aowanda, including double waterfalls, a maple path, a pine zone, cypress trees, a spruce forest observatory, and the Cingshuei River.

A Kodak Pixpro SP360 4K camera was used for video recording in this experiment. Notably, this camera can capture 360-degree views in high resolution. With two lenses, each capturing 235 degrees, researchers created 360-degree videos by merging footage using Kodak-provided software. Two videos were generated, simulating urban and forest environments, each containing seven clips with associated sounds.

Study by Takayama et al. [9] explored the physiological and psychological benefits of a digital Shinrin-yoku environment indoors in an urban facility. It observed changes in 25 subjects physical and mental states before, during, and after exposure to digital elements replicating a forest setting. Results indicated increased parasympathetic nerve activity and decreased heart rate during exposure, alongside reductions in negative mood states and increased feelings of restorativeness.

However, virtual reality (VR) experiences, while immersive, may be inaccessible to certain demographics, such as older individuals, those with dementia, and children with attention-deficit hyperactivity disorder. Additionally, VR experiences tend to be solitary, limiting the opportunity for shared experiences, which is essential for human connection.

For the experiment, two rooms were prepared: a waiting room and an experimental room. The experimental room simulated a forest environment with visual, auditory, and olfactory cues projected using five projectors, aiming to provide an immersive forest bathing experience within an indoor urban setting.

Another work, by Lopes et al. [7] discuss an experiment which was conducted in the SENSIKS multisensory booth (SENSIKS, The Netherlands), where participants experienced a multisensory nature walk synchronized with a 360-degree video through an Oculus Quest VR headset. The setup included fans, heating elements, high-resolution speakers, an under-seat subwoofer, and a scent device. The experience lasted approximately one minute and involved participants immersed in a forest environment. Physiological data were collected using a BioHarness3 chest belt and an E4 wristband. The AV sessions included a main soundtrack with a whispering voice encouraging self-reflection. Participants experienced visual, auditory, olfactory, and tactile stimuli corresponding to elements in the video, such as leaves changing, sunlight streaming, and wind sound, with corresponding sensations like air currents and seat vibrations.

All of the aforementioned texts discussed the utilization of 360-degree video for various forms of therapy. Our project endeavors to adopt a similar concept, albeit with enhancements through the incorporation of a full-body tracking system.

3 Specification of the proposed VR system

In the following section, we will focus on choosing virtual reality devices, including a body tracking system.

Virtual reality (VR) is a simulated experience that employs 3D near-eye displays and pose tracking to give the user an immersive feel of a virtual world. Applications of virtual reality include entertainment (particularly video games), education (such as medical, safety or military training) and business (such as virtual meetings). VR is one of the key technologies in the reality-virtuality continuum. As such, it is different from other digital visualization solutions, such as augmented virtuality and augmented reality.[8]

3.1 Headsets for Virtual Reality:

VR headsets are equipped with high-resolution displays for each eye, projecting images or videos to create the illusion of a 3D environment. These displays offer a wide field of view, enhancing immersion. Modern VR headsets typically prioritize high frame rates, at least 80 frames per second (FPS), for a smooth and comfortable experience. Some newer headsets push the boundaries with a frame rate of 120 Hz or higher, improving realism and reducing

motion sickness and eye strain. VR headsets usually feature high-resolution displays, often OLED or AMOLED screens. Each eye sees a slightly different perspective, mimicking how human vision works. Lenses placed between the user's eyes and the displays bend and focus light, ensuring proper image display and creating a wide field of view. Modern VR headsets come with various sensors, including accelerometers and gyroscopes, tracking the user's head movements. This tracking is crucial for real-time updates of visuals based on user's head movements, maintaining the illusion of a consistent virtual world. Some headsets also use external sensors or cameras to enhance tracking accuracy. Many VR headsets feature integrated headphones or spatial audio technology, providing a 3D sound experience. Precise audio feedback enhances the sense of presence and immersion. VR headsets often come with handheld controllers or gloves, allowing users to interact with objects and navigate in the virtual space. These input devices are tracked in the virtual reality environment, enabling accurate interaction. They are also equipped with sensors to track their position and movements. Depending on the type of VR headset, it may connect to a computer (tethered), operate independently (standalone), or use a smartphone (mobile) as its computing unit.

VR headsets come in various forms, each with its advantages and limitations, generally divided into the following categories:

Tethered VR Headsets: Connected to a powerful computer or gaming console using cables, providing high-quality graphics and immersive VR experiences but limiting user mobility.

Standalone VR Headsets: Have built-in computing power, eliminating the need for external devices. They offer portability and convenience but may have processing limitations compared to tethered headsets.

Mobile VR Headsets: Utilize smartphones as display and processing units. They are affordable and easily accessible but typically provide less immersive experiences compared to tethered and standalone headsets.

Main Differences Between Types Performance: Tethered VR headsets generally offer the best performance and graphic quality, followed by standalone headsets, while mobile headsets provide the least powerful experience.

Mobility: Standalone and mobile VR headsets offer greater mobility and can be used in a broader range of environments.

Cost: Mobile VR headsets are often the most cost-effective, followed by standalone and tethered headsets, which tend to be more expensive due to advanced hardware.

Selected Model: HTC Vive Pro 2 The HTC Vive Pro 2 was chosen for its exceptional performance and features, representing an advanced tethered VR headset suitable for various applications.

3.2 Body tracking in virtual reality

Tracking body movements in virtual reality allows transferring movements from the real world to the virtual environment, creating a more authentic experience. In our project, we integrate advanced body motion tracking, including optical motion sensing technology with HTC Tracker 3.0.

In virtual therapy, often only headsets and controllers are used, limiting tracking to the upper body. Our advanced method incorporates full-body tracking, enhancing the authenticity of the virtual experience.

There are several body tracking technologies in VR, including IMUs[1], depth sensors[4], optical motion sensing, EMG, magnetic tracking[6], ultrasound tracking, and camera-based systems.

For a comprehensive experience, we chose optical motion sensing technology with HTC Tracker 3.0, which enables full-body tracking with high accuracy and low latency.

HTC Tracker 3.0 offers a compact and lightweight design with wireless connectivity and infrared LEDs for precise tracking. The base station transmits infrared rays and uses triangulation for accurate position and orientation calculations.

Line-of-sight between tracking devices and base stations, optimal station placement, and accurate calibration are key to successful optical motion tracking in VR. HTC Tracker 3.0 with optical motion sensing delivers high accuracy, low latency and an immersive virtual reality experience.

3.3 POLAR H10: Heart Rate Monitoring

Heart Rate Variability (HRV) is crucial in psychology, offering insights into the autonomic nervous system. Psychologists use HRV to understand emotional states and stress levels, enhancing diagnostic accuracy. Integrating HRV into virtual reality therapy allows real-time monitoring of emotional reactions. Among heart rate monitoring devices, Polar H10 stands out for its accuracy and versatility.

4 Avatar tracking

Virtual reality enables users to embody virtual avatars from a first-person perspective, adapting their body image to various shapes, sizes, ages, ethnicities, or genders. Research in cognitive neuroscience reveals that this illusion of embodiment stems from multisensory correlations between real and virtual bodies, akin to the rubber hand illusion (RHI)[2]. When visual cues from the virtual body match physical sensations, such as touch and proprioception, the brain attributes them to a common source, leading to embodiment of the avatar. Spatial correspondence between virtual and physical bodies, even with static avatars,

can induce strong illusions, but dynamic replication of movements in real-time enhances the effect. Embodiment in virtual avatars improves performance in VR scenarios, reducing cognitive load and offering potential applications in therapy, rehabilitation, education, and recreation.

We employ full-body tracking using 8 trackers: HTC Vive Pro 2 for the head and 7 HTC trackers for the legs, waist, and arms (2 on each arm and 2 on each hand).

We have obtained details on the Manus Dashboard, focusing on Manus Core 1.9.0—a tool for configuring tracking devices[1]. After connecting all devices and integrating them into Steam VR, we launched the Manus application.

On the "Polygon" page, we create a user profile and ensure device connectivity in the "Tracker" section. The interface simplifies this process, labeling inputs for body part identification.

4.1 Avatar creation

We then adjust avatar parameters to match user dimensions. Calibration involves two methods: step-by-step guidance or mimicking a white robot avatar's movements in VR. Users navigate using a hand-controlled button.

Different methods and software are used to create 3D avatars - 3D Laser Scanning, which uses laser triangulation, time-of-flight, etc.

In our efforts to use photogrammetry to create customized avatars for individual users within virtual reality (VR), we encountered problems that hindered the effectiveness of our system. Despite our best efforts, the process of generating avatars for each user proved to be time and resource consuming.

In order to optimize our approach, we made a strategic decision to streamline the creation of avatars. Instead of individual avatars for each user, we took a more practical approach and used only two avatars - one designed for male users and one designed for female users. To increase the realism and detail of these avatars, we sourced high-quality realistic models.

Utilizing pre-existing, carefully crafted avatars contributes to the overall efficiency and user satisfaction in our VR environment.

5 Recording video with 360-degree cameras

The essence of 360-degree videos lies in their ability to overcome the limitations of flat screens and immerse users into a truly surrounding experience. From the comfort of a VR headset, users can turn their heads in any direction and explore the intricacies of the captured moment as if they were physically present. This transformative capability has profound implications in various domains, ranging from entertainment and education to travel and training.

[1] MANUS Knowledge Center

In recent years, monoscopic 360-degree cameras have found new applications in the emerging field of virtual reality (VR). While stereoscopic 3D is often preferred for creating immersive VR experiences, monoscopic 360-degree cameras are valuable for capturing environments where depth perception is less critical. They are commonly used for VR video content creation, virtual tours, and live streaming of events in VR.

Advancements in camera technology have led to the development of compact and high-resolution monoscopic cameras capable of capturing stunning imagery in various conditions. These cameras are equipped with features such as image stabilization, high dynamic range (HDR), and advanced autofocus systems, making them suitable for a wide range of professional and consumer applications.

The Insta360 X3 is a monoscopic compact and versatile 360-degree camera designed for immersive panoramic recording. Its supplemented by two cameras positioned on opposite sides, it captures a complete view of the surroundings simultaneously, delivering high-quality video footage.

The integration of 360-degree video content into the Unity platform necessitates careful consideration of optimal methodologies. Upon thorough investigation, two viable approaches have surfaced. The initial method entails the creation of a spherical entity within the virtual environment, wherein the video content is mapped onto the surface material. Within this construct, the user assumes a position enveloped by the sphere, thereby facilitating immersive engagement with the content. However, a better alternative appears to be to set the video scene as a skybox, because then we have less visual artifacts.

6 VR application

6.1 Main Menu

The main menu scene serves as the initial interface visible only to the psychotherapist. Within this scene, the psychotherapist is offered the option to select an avatar, choosing between a male and female representation. A button to switch between avatars increases the flexibility of avatar selection. In addition, this scene facilitates the connection of a Bluetooth heart rate measurement device, namely the Polar H10, to the laptop running the app.

6.2 Preparation Room

The training room scene is designed with a dual view of both the psychotherapist and the user. The scene includes two distinct parts: one visible to the psychotherapist, which contains the menu, and the other experienced by the user.

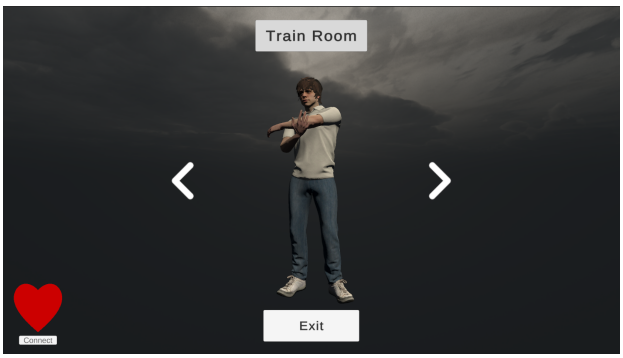


Figure 1: Main Menu scene

Figure 2: Preparation room scene: psychotherapist view

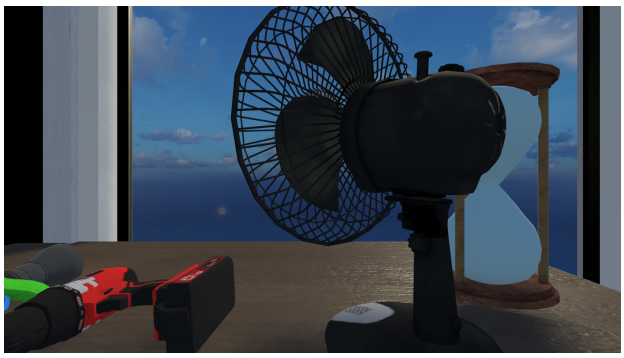


Figure 3: Preparation room scene: user view

Figure 4: Forrest scene

6.2.1 The psychotherapist's perspective

In the psychotherapist's interface, check figure 2, the menu offers basic functions:

Back button: Allows the psychotherapist to seamlessly return to the main menu scene.

Hide Menu Button: Facilitates the ability to hide the psychotherapist's menu, providing the user with an unobstructed view of their experience.

Scene overview image with arrow buttons: A visual display of the scene along with arrow buttons allows the psychotherapist to navigate between different perspectives. Clicking on the image loads the corresponding scene, offering a simplified selection process.

6.2.2 User experience

Upon entering the training room, the user finds themselves in a bedroom with various details, figure 3. Distinctive elements include a chair and a table that are decorated with various objects. To increase user engagement and adaptability, the user can touch and interact with the objects on the table, even throwing them.

The immersive environment encourages users to explore and gradually acclimate to the virtual reality environment. As the user engages with the room, the psychotherapist has ample time to reflect and select the next scene, which is consistent with the therapeutic process.

The thoughtful incorporation of interactive elements not only promotes user adaptation, but also serves as a valuable tool for the psychotherapist to gauge the user's comfort level before beginning a therapeutic intervention.

6.3 Forest Scene 3D

The forest scene is a relaxing place where the user can feel free in the forest, surrounded by nature, animals and pleasant weather with the sound of the wind.

Various pre-made elements from the Unity Asset Store were used to prepare the scene in the forest. A forest area with lots of different details and plants has been created, and the skybox has been changed. Other added details were rocks, stumps and altered forest sounds to add to the authenticity of the environment. Figure 4

One of the distinctive elements is the time dynamics. A script was created that steadily moves the light in the scene, creating a simulation of the progression of the day.

Another distinctive element is the addition of animals to the scene, which makes the environment even more realistic. Birds, squirrels, and butterflies were integrated into the scenery, which were obtained from the Unity Asset Store. Each of these assets had basic animations that were then used to create movement scenarios and animations for these animals.

To make it easier to write the paths that the animals move along, an object containing the script was created

in the scene. This script is associated with a pre-made animal, for example a squirrel with a ”move” animation. The script contains variables such as x, y, z position, rotation, clear time, and command line. The command line can be used to write commands to move the animal, for example: [L, 10, 4], [R, 90, 10]. These commands make two changes to the squirrel's movement, with each square bracket representing one command. The command [L, 10, 4] means to turn 10 degrees to the left in 4 seconds.

Another addition is the ability to create new animals at the same location with the same commands. This means that in a 30-minute intervention, a squirrel can take the same path as many times as we want.

6.4 City Scene 3d

The city scene, check figure 5, is designed with a more depressing atmosphere and depicts the drawbacks of an urban environment. This city scene is expected to be less comfortable compared to the forest scene.

Various pre-made elements from the Unity Asset Store were used to prepare the city scene, specifically the ”Urban Construction Pack” which contains a large number of urban-themed elements. The visual content is dominated by buildings and elements typical of the urban outdoor environment, such as streets and traffic lights. Shades of grey predominate in the image, both in the foreground and in the background. The user sits close to the wall, from where he can see a road that extends straight ahead of him and another road that runs in a perpendicular direction.

The first important step was to add parked cars and cars moving further away. The various cars were obtained from the Unity Asset Store. To efficiently display the traffic in the city scene, a script was written that was able to create cars in positions at random time intervals from 7 to 15 seconds. These cars moved along the road, giving the impression of a realistic urban traffic flow.

The goal of adding the moving cars was to increase the dynamism and authenticity of the city scene. In this way, the user experiences the city not only as a static environment, but also as a place with a vibrant urban movement. The cars, which appear and move according to random time intervals, add an element of unpredictability to the scene and at the same time enliven the overall impression of the urban environment.

Another detail was the addition of trash cans, which were also obtained from the Unity Asset Store. Rats were also added; some are ”static” and simulate that they are looking for something next to the trash cans. There is also another type of rat that spawns at some intervals and runs a certain route. As with the squirrels, a script was used with the same approach, in which the animal's position, rotation and movement commands could be entered. An array of strings was also added to indicate the times when the rat was to be created in the scene.

It is important to mention that all these created scenes were in accordance with psychologists from the Faculty of

Figure 5: City scene

Social and Economic Sciences (FSEV), who will conduct interventions with VR therapy designed in this paper.

6.5 Forest Scene 360-degree video

We employ an Insta360 camera system. Resulted in a thirty-minute long panoramic video, crafted to encapsulate the serene beauty of the ancient forest. Subsequently, employing Adobe Premiere Pro 2020, we enhanced the vibrancy and contrast of the video, enriching the viewer's immersive experience. Finally, the video was integrated into a skybox of scene.

6.6 City Scene 360-degree video

Utilizing an Insta360 camera, we got a 30-minute long video encompassing the city's center. Subsequently, the preparatory steps undertaken to integrate this footage into Unity mirror those employed for the creation of our forest 360-degree video.

7 3D scene vs. 360 video

In this section, we will undertake a comparative analysis between 360-degree video and 3D scenes.

First, we will discuss the disadvantages associated with 360 video technology. First of all, it is worth emphasizing the significant file size due to the high resolution inherent in such videos. Rendering these videos can be a computationally intensive task, often requiring around 5-10 hours, even when using high-performance eight-core processors. However, this temporary investment becomes trivial if we have the necessary time to prepare.

The primary concern lies in the recorded resolution, typically set at 5.7K, and the perceived quality experienced when using Head Mounted Displays (HMDs). Despite the seemingly high resolution, the visual fidelity observed through HMDs frequently fails to meet expectations. Additionally, the inherent monoscopic nature of these videos precludes the simulation of depth perception, thereby diminishing the immersive potential. While the integration

of stereoscopic cameras offers a potential remedy to this issue, the high associated with such equipment, ranging from 5000 to 6000 EUR, present a significant barrier to widespread adoption. Moreover, even with the deployment of these advanced cameras boasting resolutions up to 8K, the resultant quality remains suboptimal, failing to fully capture the desired immersive experience.

Furthermore, initial tests revealed a problem regarding the integration of three-dimensional models into the spherical structure of 360-degree videos. If a 360-degree video is within a sphere or created as a skybox, discrepancies arise when incorporating avatar bodies, where the spatial alignment of rendered elements appears incongruous within the scene. It seems as if the legs are not touching the ground. Additionally, perceptual anomalies arise when viewers direct their gaze downwards, wherein visual distortions lead to objects appearing disproportionately larger than anatomical elements, further compromising the immersive fidelity. These discrepancies are indicative of challenges inherent between two-dimensional video elements and three-dimensional objects within the context of a spherical environment.

In preliminary study we have tested both 360 and VR forest scenes with 5 participants. All of the 5 participants perceived the VR scene as more immersive, calming and possessing higher resolution. We are planning to carried out a user study with more participants to further validate our findings and assess any potential differences in the perception of different types of natural scenes in virtual reality environments.

8 Rotating chair for scenes

For better quality testing, psychologists added a chair on which the user can turn around. So then in each scene, the user will be situated in a chair capable of full 360-degree rotation. Consequently, it is imperative to incorporate this functionality within the application. Initially, we identified a suitable 3D chair model from the Unity Asset Store for seamless integration into the interactive environment. With the chair model now integrated into our scene, the primary challenge lies in enabling its rotation to coincide with the user's movements, thus simulating realistic behavior.

To address this challenge, we devised a solution leveraging two box colliders. The first collider is strategically positioned at the avatar's posterior, aligning with the optimal location for chair alignment. Subsequently, the second collider is placed on the chair itself. The mechanics operate as follows: upon collision detection between these two colliders, indicative of the user's seated position, the chair emulates rotational orientation along the y-axis of the collider situated at the avatar's posterior. This method facilitates seamless synchronization between the user's movements and the chair's rotation, thereby augmenting the realism of the simulation.

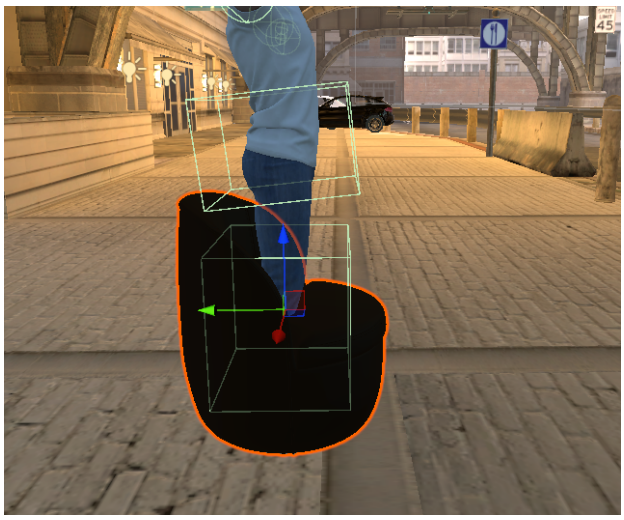


Figure 6: Chair & Avatar with colliders

The actual chair is virtually aligned during the initial calibration of the system and is positioned and rotated identically at the start of each intervention.

9 Future work

The developed therapy application is been currently used in interventions with participants across diverse demographics by doctoral students of the Faculty of Social and Economic Sciences COMENIUS UNIVERSITY BRATISLAVA (FSEV UK) for their applied psychology research. The undergoing study will include more than 50 participants and the results will be available in late 2025.

10 Conclusions

In this study, we embarked on an exploration of virtual reality (VR) environments and their potential implications for mental well-being. By contrasting 3D scenes with 360-degree videos, we aimed to shed light on the nuanced differences between these modalities in eliciting emotional responses and promoting relaxation.

Our preliminary investigation revealed several key findings. Firstly, while 360-degree videos offer immersive panoramic experiences, their limitations in capturing depth perception and spatial relationships challenges in delivering truly immersive environments. The integration of stereoscopic cameras may present a solution, albeit at a considerable cost. Conversely, 3D scenes afford greater flexibility and dynamic control over environmental elements, albeit with computational demands and potential fidelity limitations. Additionally, it's imperative to validate these findings through more extensive user studies. While our research underscores the importance of interactive elements within 3D scenes for fostering user engagement

and immersion, further investigation is necessary to confirm these observations. Additional research is warranted to substantiate these claims and refine our understanding of their impact.

Unlike previous approaches/articles, we have added full-body tracking for better immersion.

The completed application has been forwarded to the researchers at FSEV UK.

References

[1] Javier González-Alonso, David Oviedo-Pastor, Héctor J Aguado, Francisco J Díaz-Pernas, David González-Ortega, and Mario Martínez-Zarzuela. Custom imu-based wearable system for robust 2.4 ghz wireless human body parts orientation tracking and 3d movement visualization on an avatar. *Sensors*, 21(19):6642, 2021.

[2] Mar Gonzalez-Franco, Brian Cohn, Eyal Ofek, Dalila Burin, and Antonella Maselli. The self-avatar follower effect in virtual reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 18–25, 2020.

[3] Margaret M. Hansen, Reo Jones, and Kirsten Tocchini. Shinrin-yoku (forest bathing) and nature therapy: A state-of-the-art review. *International Journal of Environmental Research and Public Health*, 14(8), 2017.

[4] Belinda Lange, A Rizzo, Chien-Yen Chang, Evan A Suma, and Mark Bolas. Markerless full body tracking: Depth-sensing technology within virtual environments. In *Interservice/industry training, simulation, and education conference (I/ITSEC)*, 2011.

[5] Grace Y.S. Leung, Hadar Hazan, and Christian S. Chan. Exposure to nature in immersive virtual reality increases connectedness to nature among people with low nature affinity. *Journal of Environmental Psychology*, 83:101863, 2022.

[6] Gabriele Ligorio, Elena Bergamini, Ilaria Pasciuto, Giuseppe Vannozzi, Aurelio Cappozzo, and Angelo Maria Sabatini. Assessing the performance of sensor fusion methods: Application to magnetic-inertial-based human body tracking. *Sensors*, 16(2), 2016.

[7] Marilia K. S. Lopes, Belmir J. de Jesus, Marc-Antoine Moinnereau, Reza A. Gougeh, Olivier M. Rosanne, Walter Schubert, Alcyr A. de Oliveira, and Tiago H. Falk. Nat(ur)e: Quantifying the relaxation potential of ultra-reality multisensory nature walk experiences. In *2022 IEEE International Conference on Metrology for Extended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)*, pages 459–464, 2022.

[8] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, volume 2351, pages 282–292. Spie, 1995.

[9] Norimasa Takayama, Takeshi Morikawa, Kazuko Koga, Yoichi Miyazaki, Kenichi Harada, Keiko Fukumoto, and Yuji Tsujiki. Exploring the physiological and psychological effects of digital shinrin-yoku and its characteristics as a restorative environment. *International Journal of Environmental Research and Public Health*, 19(3), 2022.

[10] Xiaobo Wang, Yaxing Shi, Bo Zhang, and Yencheng Chiang. The influence of forest resting environments on stress using virtual reality. *International Journal of Environmental Research and Public Health*, 16(18), 2019.

[11] Chia-Pin Yu, Hsiao-Yun Lee, and Xiang-Yi Luo. The effect of virtual reality forest and urban environments on physiological and psychological responses. *Urban Forestry & Urban Greening*, 35:106–114, 2018.

Presentation and Quiz Visualization of Virtual Procedure Manuals

Amna Jusić*
Emir Cogo†
Ehlimana Cogo‡
Aya Ali Al Zayat§
Supervised by: Selma Rizvić¶

Faculty of Electrical Engineering
University of Sarajevo
Sarajevo / Bosnia and Herzegovina

Abstract

Virtual manuals for the usage of different procedures require a certain set of steps, for the users to familiarize themselves with the procedures required to use a set of desired objects. Apart from the textual information, it is also important to present different types of spatial information at the same time, while also including the direction of this type of information. Usually, a set of operations needs to be visualized by presenting the key locations of the operations, the order of operations, and their motion. Some operations require pulling, twisting, pushing, pressing, or some combination of these motions in different places at the same time. It is often important to test how much the user learned about all of the presented operations, to verify that the user can operate the procedure safely and successfully without supervision. In this paper, we analyze different visualization methods for multiple types of information by using animated 3D arrows. These arrows can present information by using gradient colors, shapes, sizes, and rotations of shapes, as well as shape animations in the desired direction. The proposed approach was successfully applied to create a virtual usage manual for a set of operations of a procedure. A virtual quiz that verifies whether the user learned all of the required steps was created. The quiz also requires the user to visually show the required operation motion. A small user survey was conducted, indicating that younger, highly educated age groups are more open to the usage of virtual manuals and that users find virtual quizzes helpful but are not confident that they possess the digital skills necessary for undertaking them.

Keywords: Virtual Reality in Education, Intelligent Tutoring Systems, 3D UI

*ajusic5@etf.unsa.ba
†ec15261@etf.unsa.ba
‡ekrupalija1@etf.unsa.ba
§aalialzaya1@etf.unsa.ba
¶srizvic@etf.unsa.ba

1 Introduction

The fast advancement of technology has enhanced our everyday lives in multiple aspects, from the near-instant availability of huge amounts of data to the complex user interfaces of web-applications offering business and customer service. One such aspect is the virtualization of objects and the development of simulation environments [2] that allow users to execute safety-critical programming code without endangering themselves, the objects of operation, or their surroundings. Apart from being able to design, implement, and test various products virtually, it is also possible to learn how to use them without owning or having access to the physical copy of the given product, which eliminates the possibility of breaking the product or getting hurt due to misuse. This can be especially useful for rare or very expensive products that require long training time (e.g. flight simulators) or are very fragile and require special care (e.g. chemistry or medical equipment).

User manuals [13] contain detailed information about products and their functionalities, as well as sets of instruction steps for their operation. The high amount of details, however, makes user manuals very long and therefore hard to perceive by customers, which is why the usage of visual information conveyed by images is very important for customer satisfaction [18]. The quality of the user manual is correlated with the quality of the product by customers [6], which is why companies need to properly convey important information about the usage of the product for the product to be accepted well. The study conducted by Tsai et al. [16] showed that a very high number of older adults often use product manuals and are willing to learn new technologies and use additional training materials. This indicates that the usage of the newest trends to convey information about the correct usage of products in a fast and straightforward way is suitable for all age groups and should be adopted to improve customer satisfaction.

In recent years, virtual reality (VR) and augmented reality (AR) technologies were introduced into training and assistance systems [9] as help and support to technicians. Different types of solutions have different strengths and

weaknesses, as discovered in a study by Laviola et al. [8], enabling the production companies to choose the type of technological solution that suits the type of their product the most. However, virtual product manuals usually convey information textually, without using the full potential of the technologies used to create them. This work presents a new approach that takes into consideration the type of information users are expected to learn through different product manuals. A method that uses various types of arrows for visualizing multiple types of information at the same time successfully is presented, without confusing the user with lots of text that is difficult to understand and hard to memorize.

The proposed approach was applied to a use case that simulates various types of instructions for the usage of an example product a user is expected to learn, to demonstrate the ease of presenting information by visualizing them instead of by using textual fields. A quiz mode in the virtual environment is also proposed to test the knowledge of the user about the learned procedure. However, in order to avoid the user mechanically learning the answers to the questions, randomization and different difficulty levels are used, utilizing the strengths of parametrizing the proposed virtual arrows and hand objects. This approach entirely removes the textual information from virtual product manuals and quizzes about their usage to reduce their complexity and improve the ease of their understanding by customers.

This paper is structured in the following way: Section 2 describes the background and related work for virtual product manuals, Section 3 introduces the proposed ways of visualizing different types of information in virtual environments, Section 4 contains the results of applying the proposed approach on an example use case containing a set of ten steps for operating a procedure, whereas Section 5 summarizes the achieved results and gives directions for future work.

2 Background

Several works propose the usage of VR applications as an alternative to traditional training manuals, mainly due to safety concerns and a possible lack of required equipment. Tichon and Scott in [15] compared the use of VR in safety training to a *PowerPoint* presentation. They showed that the usage of VR might provide more effective training because the group of users that was trained by using VR materials gained higher performance scores after their knowledge and skills in identifying manual handling hazards were tested. AlAwadhi et al. [1] presented a VR application for educational purposes such as practical learning and performing live experiments in engineering and science. This application was meant to help students practice dangerous experiments safely, avoiding risks and problems due to lack of access to equipment. De Lorenzis et al. [10] presented a Virtual Reality Training System (VRTS) designed to train first responders in the high-capacity pumping procedure. Participants reacted positively to the application and the overall quality of the training experience improved, as shown by the scores of the quiz session that showed a knowledge gain associated with the use of the VRTS. Kind et al. [7] presented an architecture that enables engineers to perform virtual assembly simulation with force feedback in a VR environment. This architecture is the basis of a testbed for conducting virtual assembly simulations.

The previously mentioned approaches rely on the usage of VR technologies that are expensive, require specialized equipment, and mostly cannot be used from home by most customers. Several approaches have therefore turned to AR technologies that are much easier to use and available to a large number of customers, such as Ferrati et al. [5] (for assembling hydraulic hoses for cherry picking) or Xue et al. [17] (for assembling and maintaining avionics equipment). A study by Dorloh et al. [4] showed that when comparing the usage of printed manuals, video guides, and AR technology, the speed of using an AR-enhanced manual was slow, but the quality of the performed task was the best. However, regardless of whether AR or VR technologies are used for enhancing the product manuals, the information is mostly textually presented, or at best by using highlight colors and shapes, as well as virtual hands to depict the user that operates the product. In a study conducted by Pekerti [12], the usage of pictures and arrows in an operation instruction set improved the success of performing the given task, indicating that arrows convey unique types of information and need to be used as instructional objects in user manuals.

3 Proposed approach

The proposed approach for creating virtual manuals contains two modes that will be described in detail in the following paragraphs.

3.1 The procedure presentation mode

In the presentation mode, the user cycles through all steps of the procedure to learn the information about each step. The scene for the presentation contains a single procedure. The procedure is composed of procedure steps. One step can be composed of several operations. One operation can have multiple hand objects and arrow objects, depending on the number of users who participate in the procedure and the number of required operations. The scene contains two virtual hand objects for each user participating in the procedure and they are arbitrarily separately configurable. The hand objects can be configured to use the left hand, right hand, or both hands. Both hands can be rotated and each finger part can be rotated to mimic the real-world pose of the hand, as shown in Figure 1. In this example, two operators participate in the procedure (*Operator 1* -

green virtual hand objects, *Operator 2* - blue virtual hand objects). Both hands of *Operator 1* are rotated upwards, enabling the operator to push an object forward. The left hand of *Operator 2* has the index finger pointing forward, enabling the operator to push a button, whereas the right hand is rotated towards the left with slightly bent fingers, enabling the operator to lift an object by the handle from the left side. This example shows that different operators can perform entirely different operations within the same procedure step. Moreover, a single operator can use different hand gestures to do two different things in parallel, mimicking the necessary operations that need to be performed in real life.

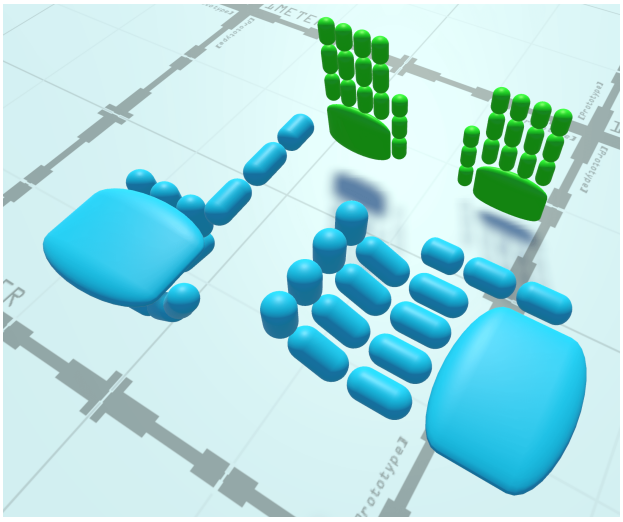


Figure 1: Proposed types of two-handed gestures for multiple operators in the virtual environment

The arrow objects of the presentation scene can contain multiple pieces of information. The different types of arrows are shown in Figure 2. The shape of each arrow shows the path of the required operation (e.g. the green arrow follows a straight path, then turns towards the left, and then follows a straight path again), the tip of the arrow shows the direction of the operation (e.g. the red arrow rotates upwards and then downwards in a clockwise manner), whereas the color of the arrow shows the operation group. The operation group is used to differentiate steps that require multiple simultaneous operations performed by multiple operators at the same time. If multiple operators participate in a single operation step, a single operator is designated a color and all objects that require the assistance of that operator are marked in the designated color. The body of the arrow can also have different configurations along its path. The normal body (the green arrow in Figure 2) is used to show the path of a moving object linked to the starting point of the arrow. A dashed body with animated moving dashes (the blue arrow in Figure 2) is used to show the speed of the operation in addition to its path. A twisted arrow body (the orange arrow in Figure 2) is used to show an operation where pulling and twisting at the same time is required. A circular arrow body (the red arrow in Figure 2) is used to show a rotational operation. If an operation has a circular motion larger than 360 degrees, multiple arrow objects are used for each full circle and a leftover circle if the last circle is not complete. Some operations can also contain information about the operation result, depicted by using color changes or position changes of other objects after the operation has been performed.

Figure 2: Proposed types of arrows in the virtual environment (red - circular, green - normal, blue - dashed, orange - twist and pull)

3.2 The procedure quiz mode

After the user learns all the steps of the procedure, they can switch to the procedure quiz mode. The quiz mode contains the same number of steps as the presentation mode. Each step contains all objects related to the step from the presentation part (i.e. the correct set of operations) and additional objects that contain incorrect operations. Each question of the quiz is related to a single procedure step where the user is required to choose the right answer by using multiple parameters (the path, direction of the operation, and the operation group, as well as the speed, circular motions, or the operation result if necessary). To answer the question correctly, the user must choose the correct locations where the operations of the given procedure step will be performed, the correct operation types, the correct operators and simultaneous operations, as well as the correct directions and gestures. Throughout the quiz, the user can cycle through each location in the virtual scene. An operation is shown at each location and the user can change its parameters in the desired way or delete the operation so that the location has no operations. It is possible to cycle through all hand and arrow objects of the operation of a single location. When an arrow object or hand object is selected, the color can also be changed in order to choose the correct operators and simultaneous operations. The gestures of the virtual hands can be changed to pick their correct positioning. Multiple dimensions contribute to the difficulty of each question. After each step, the user is informed of whether their answer is correct or incorrect. Only visual ways of describing the outcome are used, as shown in Figure 3, where the user is informed of answering the question wrong by using the *X* mark colored in red

and shown above their answer on the product. At the end of the quiz, the user is presented with the overall number of incorrect answers and whether they passed the quiz or not.

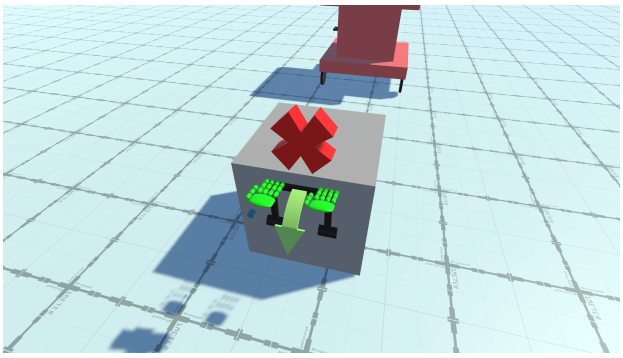


Figure 3: Visualization of the wrong answer in the quiz mode

4 Results

In order to demonstrate the proposed approach, a virtual scene was created by using the *Unity real-time development platform* [14]. The *Procedural indicators* Unity Asset Store plugin [3] was used for creating the 3D arrow models that allow for dynamic parameter modification required for the previously described operation changes. The quiz, all manipulated objects, and the sequence of operation steps were assembled in Unity Editor without importing any data from external sources in the runtime build. The application contains the question templates, operation steps, and the programming logic of the quiz. The operation of the application and the results of a conducted user study are explained in the following paragraphs in detail.

4.1 The procedure presentation mode

The user is first shown the presentation of the procedure that is visualized on the virtual scene in Figure 4. It is visible that the procedure contains four different objects, indicating that the procedure for their usage is very complex. This was intentionally done so that all the different capabilities of the proposed approach can be demonstrated. However, it is important to note that most consumer products would require a much lower level of difficulty, and the procedure from the created use case is more fitting for an industrial or experimental setup. The user can cycle through all of the procedure steps and view the correct operations and all the relevant information about every operation. This process can be repeated until the user successfully learns all operation steps and is satisfied with their level of knowledge.

All steps of the operation are shown in Figure 5 and are described as follows:

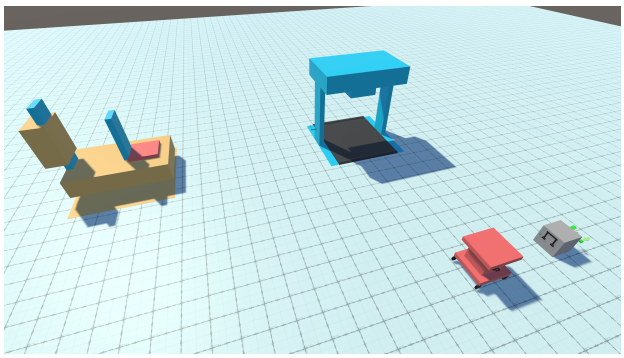
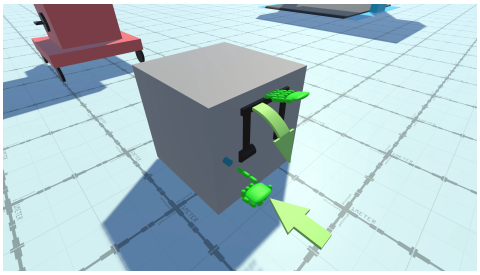


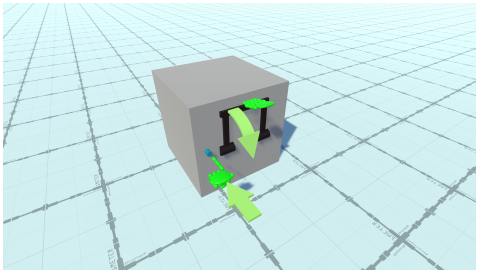
Figure 4: The virtual environment containing the 3D model of the example product

- **Step 1**: Push the button next to the handle and pull the handle down all the way.
- **Step 2**: Do the same operations as during Step 1 on the other side of the box.
- **Step 3**: Two persons need to grab the box by the handles and lift it onto the cart.
- **Step 4**: Pull and twist the pin on the cart so that the wheels unlock.
- **Step 5**: Bring the cart to the scan area.
- **Step 6**: Slowly turn the crank for two full circles in a clockwise manner, until the cart reaches the scan finish area.
- **Step 7**: Bring the cart to the loading area.
- **Step 8**: Lift the box onto the device.
- **Step 9**: Turn the range selector knob full circle in a clockwise manner.
- **Step 10**: Push the button so that the red light turns to green.

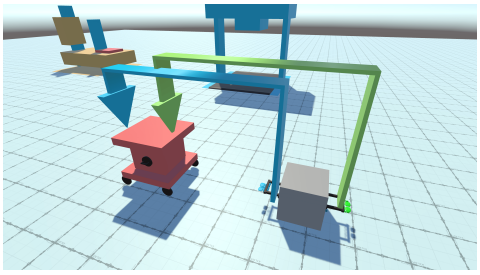
Step 1 (Figure 5a) demonstrates an action composed of two simultaneous operations because the handle can be moved only while the button is pushed down. Steps 3 (Figure 5c) and 8 (Figure 5h) demonstrate actions that are performed by two operators. Step 4 (Figure 5d) demonstrates the twist and pull operation. Steps 5 (Figure 5e) and 7 (Figure 5g) demonstrate different movement operations. Step 6 (Figure 5f) demonstrates the operation speed and circular operations. Steps 9 (Figure 5i) and 10 (Figure 5j) demonstrate operations on a smaller scale. Step 10 also introduces an indicator for the color change that visualizes the result of the operation after it is successfully finished.



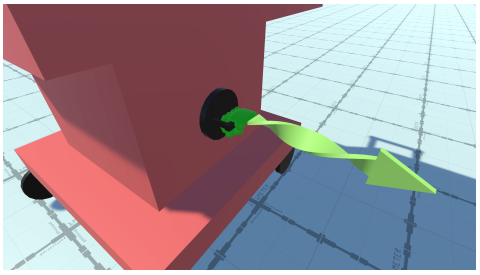
(a) Step 1



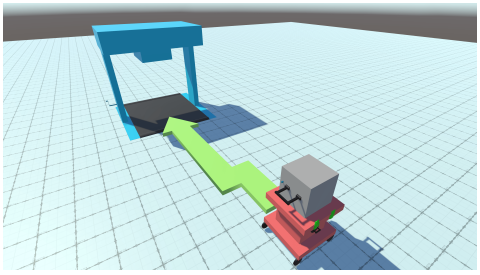
(b) Step 2



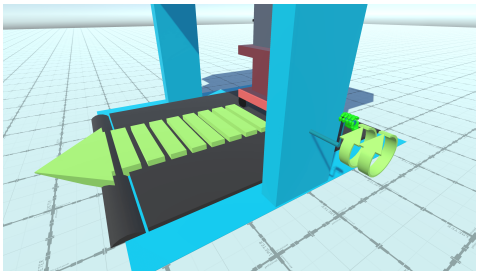
(c) Step 3



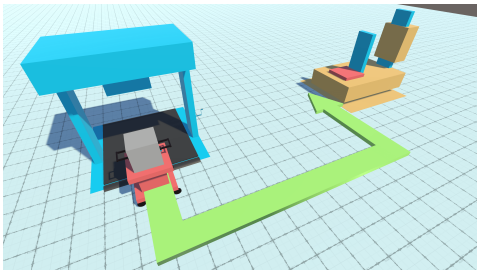
(d) Step 4



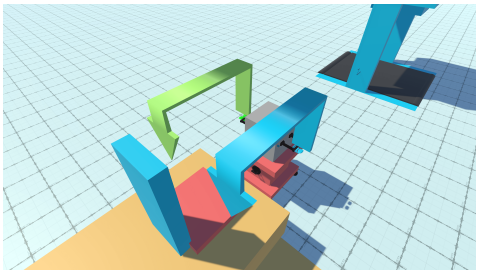
(e) Step 5



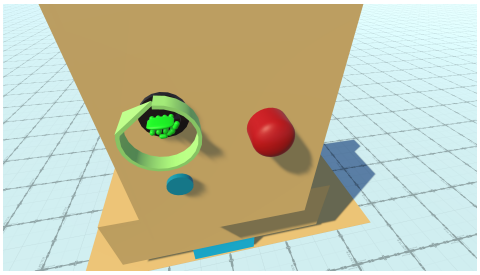
(f) Step 6



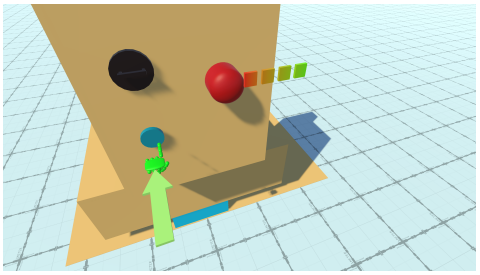
(g) Step 7



(h) Step 8



(i) Step 9



(j) Step 10

Figure 5: Visualization of all steps for operating an example procedure

4.2 The procedure quiz mode

The quiz mode of the virtual scene contains the correct answer, as well as misleading information to test the certainty of the user that the procedure they chose is correct. Each step contains several incorrect locations and operations, and the user must also set the proper groups and directions. The initial setup of the directions and groups is randomized. Each answer also contains different hand positions in the operation, to ensure knowledge of the proper operation type and the correct hand gesture required to correctly perform the operation. The difficulty of the test can be configured by lowering or increasing the number of incorrect locations, incorrect operations, and their similarity to the correct answer. Several possible scenarios that might confuse the user are shown in Figure 6. To ensure the proper understanding of the order of the steps, information from future steps is included as the answers to a given step. For example, all potentially correct and incorrect answers for step 4 are included as answers for step 1 as well (Scenario 1 on Figure 6a). This is used to test the user if they are certain that they should unlock the cart first, or put the box on the cart instead. Loading the box on the cart that has unlocked wheels is a potentially dangerous operation and it is important to teach the user the proper order of the operations. In step 1, the user is also provided with the locations of the two handles and one button. The user must choose only one handle linked to the same button for that handle and a button operation linked to the opening of the handle, as well as the proper directions of the operations and the proper operation group. Step 2 contains incorrect answers from step 3 that include only a single operator with the open handle (Scenario 2 on Figure 6b). This might mislead the user that the box can be lifted without a second operator, which is not possible. Movement operations in steps 5 and 7 contain answers with wrong paths. Step 6 contains answers with the wrong operation speeds of different movements of the cart during the crank-turning operation. One of the variants also shows the user turning the crank once instead of two times and in the wrong direction (counter-clockwise instead of clockwise), as shown in Scenario 3 in Figure 6c.

4.3 User evaluation

A user survey was conducted by using Lyssna [11] to evaluate the proposed approach. The survey was completed by a total of 15 users from different countries around the world with different characteristics (53.33% male and 46.67% female; 20% high school graduates, 46.67% college graduates, and 33.33% postgraduates; 26.67% intermediate and 73.33% advanced computer users; 20% aged 20-24, 20% aged 25-29, 26.67% aged 30-34, 20% aged 45-49, 6.67% aged 50-54 and 6.67% aged 60-64). The users were presented with the image of a single step of the virtual procedure manual and the image of the textual description of the step, after which they chose one or multiple statements that they agreed with. The results of the survey are summarized in Table 1. A total of 46.67% users chose the virtual manual image as their preferred design, with the maximum age group of 45-49 years. The results indicate that a significant number of users do not read product manuals and that younger users are more open to the usage of virtual product manuals than older users, who rarely chose the virtual manual as their preferred one and agreed with the proposed statements.

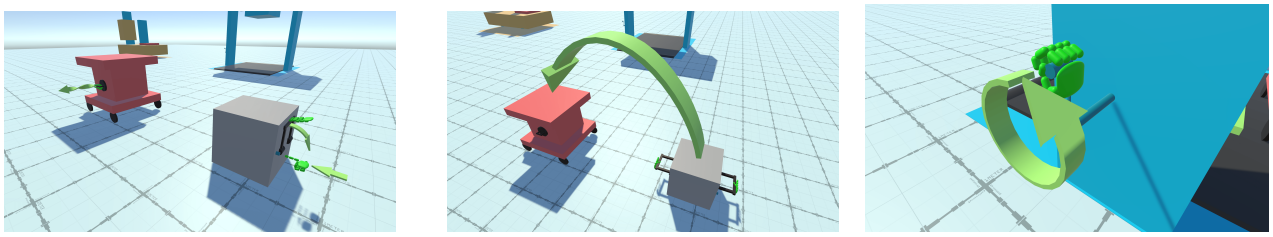
Statement	Agrees	Age group	Education level
When I buy a product, I usually read and use the product manual supplied with it.	9	22.22% (20-24) 11.11% (25-29) 33.33% (30-34) 11.11% (45-49) 11.11% (50-54) 11.11% (60-64)	11.11% (High school) 44.44% (College) 44.44% (Postgraduate)
I would like to use the virtual manual that was shown on the image to learn how to operate the device.	6	50% (20-24) 33.33% (25-29) 16.67% (30-34)	16.67% (High school) 66.66% (College) 16.67% (Postgraduate)
The visual description of the procedure by using virtual arrows is simple to understand.	6	50% (20-24) 33.33% (25-29) 16.67% (45-49)	16.67% (High school) 50% (College) 33.33% (Postgraduate)
The usage of virtual arrows and hands makes it easier to perceive the required action for handling the device.	6	50% (20-24) 16.67% (25-29) 33.33% (45-49)	33.33% (High school) 50% (College) 16.67% (Postgraduate)
It is easier to understand the visual than the textual description of the operation.	3	66.67% (20-24) 33.33% (25-29)	33.33% (High school) 33.33% (College) 33.33% (Postgraduate)

Table 1: The results of the preference test of the user evaluation survey

An example quiz question was shown to the users to evaluate the procedure quiz mode, depicting a step of the procedure and what the user needs to choose to perform the procedure correctly. The results are summarized in Table 2. The achieved results indicate that the maximum age group that was familiar with the usage of AR or VR technologies is 30-34. A high number of users would be willing to undertake the quiz, but they did not feel that they possessed a high enough level of digital skills to do it. This is contrary to the perceived level of computer skills that the users selected at the beginning of the survey, which was advanced for a total of 86.67% users.

5 Conclusion

In this paper, we proposed an approach in a virtual environment that can be used in product manuals to teach users about the handling of different products or procedures. Afterward, the user's knowledge about the presented procedure steps can also be tested by using a virtual quiz. The usage of virtual arrows and hand objects makes it possible to show a large scope of potential operations including information that is hard to understand or memorize textually, which makes this general approach applicable to many different types of procedures. The usage of our approach also makes it possible to show all steps of a procedure without using textual information at all. The created quiz has configurable difficulty levels by using different parameter values of the virtual arrows and hand objects. In this way, a



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figure 6: Visualization of different scenarios containing incorrect information during the quiz mode

Statement	Agrees	Age group	Education level
I have previous experience with the usage of virtual reality (VR) or augmented reality (AR) technologies.	5	20% (20-24), 40% (25-29) 40% (30-34)	60% (College) 40% (Postgraduate)
I would be able to undertake this virtual quiz because I have enough digital skills.	6	50% (20-24), 16.67% (25-29) 16.67% (30-34), 16.66% (45-49)	33.33% (High school) 50% (College) 16.67% (Postgraduate)
I would be willing to undertake this virtual quiz to check my level of knowledge.	8	12.5% (20-24), 25% (25-29) 25% (30-34), 12.5% (45-49) 12.5% (50-54), 12.5% (60-64)	62.5% (College) 37.5% (Postgraduate)
The virtual quiz questions are straightforward and understandable.	4	75% (20-24) 25% (30-34)	25% (High school) 75% (College)
I believe that this quiz would help me memorize the correct operation for handling the device.	6	33.33% (20-24), 33.33% (25-29) 33.34% (30-34)	16.67% (High school) 33.33% (College) 50% (Postgraduate)
It is more difficult to answer the questions correctly when choosing the correct arrow than by choosing one of the provided textual answers.	4	25% (25-29), 25% (30-34) 25% (45-49), 25% (60-64)	50% (College) 50% (Postgraduate)
I do not agree with any of the above.	1	100% (45-49)	100% (High school)

Table 2: The results of the design test of the user evaluation survey

large set of potential answers can be created automatically, without needing to design large textual answers. The time needed to describe the path is faster and more straightforward when using arrow objects than by merely describing the whole path with words. The multidimensional nature of the questions furthermore reduces the need to mention every variant that would otherwise need to be described by text, or by using still images. The starting configuration of each question is randomized and answers for multiple steps are intertwined, which removes the possibility of the user learning the questions mechanically, without understanding the procedure or investing the effort to learn the correct order of procedure steps. A small user survey was conducted and it showed that the virtual manuals should mainly target younger age groups with a higher level of education. The number of users that found the usage of virtual arrows helpful is promising, however it was not at a satisfying level for age groups over 34 and users who did not go to college. Most users considered virtual quizzes as helpful, however they were not confident in their digital skills to undertake them. This indicates that additional training needs to be performed to encourage the users to start using modern technologies to familiarize themselves with the desired products.

Our approach presents only the beginning idea for making virtual product manuals more interactive and educational. VR and AR technologies were not utilized in our approach to make the virtual environment as simple as possible. The strengths of these technologies can be incorporated to improve the quality of the virtual manual presentation and quiz scenes. Existing virtual manuals can be adjusted to include interactive arrows, with the purpose of improving their quality and the level of understanding of the required operation steps by customers. A bigger user survey could then be conducted, requiring the users to learn the procedure by using the first version of the virtual product manual (that contains textual information about product usage) and the improved version of the virtual product manual (that contains interactive virtual arrows and hand objects). The results of the survey could be used to further improve the proposed approach and possibly integrate the two approaches for achieving the best results and a high level of customer satisfaction.

References

[1] Shafiqah AlAwadhi, Ahmad Said Nafi AlHabib, Dareen Murad, F. AlDeei, M. AlHouti, Taha Beyrouthy, and Samer Al-Kork. Virtual reality application for interactive and informative learning. In *2017 2nd International Conference on Bio-engineering for Smart Technologies (BioSMART)*, pages 1–4, Paris, 2017. doi: 10.1109/BIOSMART.2017.8095336.

[2] Aitor Arrieta, Shuai Wang, Ainhoa Arruabarrena, Urtzi Markiegi, Goiuria Sagardui, and Leire Etxeberria. Multi-objective black-box test case selection for cost-effectively testing simulation models. In *GECCO 2018 - Proceedings of the 2018 Genetic and Evolutionary Computation Conference*, pages 1411–1418, Kyoto, 2018. doi: 10.1145/3205455.3205490.

[3] Cogobyte. Procedural indicators. https://assetstore.unity.com/packages/tools/modeling/procedural-indicators-105710, 2024. Accessed: (30/03/2024).

[4] Halimoh Dorloh, Kai-Way Li, and Samsiya Khaday. Presenting job instructions using an augmented reality device, a printed manual, and a video display for assembly and disassembly tasks: What are the differences? *Applied Sciences*, 13(4):1–17, 2024. doi: 10.3390/app13042186.

[5] Francesca Ferrati, John Ahmet Erkoyuncu, and Samuel Court. Developing an augmented reality based training demonstrator for manufacturing cherry pickers. *Procedia CIRP*, 81:803–808, 2019. doi: 10.1016/j.procir.2019.03.203.

[6] Osman Gök, Pervin Ersoy, and Gülmuş Börühan. The effect of user manual quality on customer satisfaction: The mediating effect of perceived product quality. *Journal of Product and Brand Management*, 28(4):475–488, 2019. doi: 10.1108/JPBM-10-2018-2054.

[7] Simon Kind, Andreas Geiger, Nora Kießling, Michael Schmitz, and Rainer Stark. Haptic interaction in virtual reality environments for manual assembly validation. *Procedia CIRP*, 91:802–807, 2020. doi: 10.1016/j.procir.2020.02.238.

[8] Enricoandrea Laviola, Michele Gattullo, and Alessandro Evangelista. Displaying augmented reality manuals in the design phase of the product lifecycle. In *JCM 2022: Advances on Mechanics, Design Engineering and Manufacturing IV*, page 1316–1326, Ischia, 2022. doi: 10.1007/978-3-031-15928-2_115.

[9] Frieder Loch, Gennadiy Koltun, Victoria Karaseva, Dorothea Pantförder, and Birgit Vogel-Heuser. Model-based training of manual procedures in automated production systems. *Mechatronics*, 55:212–223, 2018. doi: 10.1016/j.mechatronics.2018.05.010.

[10] Federico De Lorenzis, Filippo Gabriele Pratticò, and Fabrizio Lamberti. Hcp-vr: Training first responders through a virtual reality application for hydrogeological risk management. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 2: VISIGRAP*, pages 273–280, Torino, 2022. doi: 10.5220/0011007800003124.

[11] Lyssna. Usability test - virtual manual. https://app.lyssna.com/do/9b90102e5423/caf4, 2024. Accessed: (30/03/2024).

[12] Andre A. Pekerti. Augmentation of information in educational objects: Effectiveness of arrows and pictures as information for actions in instructional objects. *Australasian Journal of Educational Technology*, 29(6):840–869, 2013. doi: 10.14742/ajet.312.

[13] Duc Truong Pham, Rossi Setchi, and Stefan Dimov. Enhanced product support through intelligent product manuals. *International Journal of Systems Science*, 33(6):433–449, 2002. doi: 10.1080/00207720210133624.

[14] Unity Technologies. Unity real-time development platform: 3d, 2d, vr and ar engine. https://unity.com, 2024. Accessed: (08/02/2024).

[15] Jennifer Tichon and S. Scott. Virtual reality manual handling induction training: Impact on hazard identification. *Asia Pacific Journal of Contemporary Education and Communication Technology*, 5(1):49–58, 2019. doi: 10.25275/apjcectv5i1edu5.

[16] Wang-Chin Tsai, Wendy A. Rogers, and Chang-Franw Lee. Older adults' motivations, patterns, and improvised strategies of using product manuals. *International Journal of Design*, 6(2):55–65, 2012.

[17] Zhengjie Xue, Jun Yang, Ruchen Chen, Qiang He, Qixiu Li, and Xuesong Mei. Ar-assisted guidance for assembly and maintenance of avionics equipment. *Applied Sciences*, 14(3):1–23, 2024. doi: 10.3390/app14031137.

[18] Liang Zhang, Anwen Hu, Jing Zhang, Shuo Hu, and Qin Jin. Mpmqa: Multimodal question answering on product manuals. In *The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)*, pages 13958–13966, Washington DC, 2023. doi: 10.1609/aaai.v37i11.26634.

Comparing Interaction Methods in a VR Rock Climbing Simulation

Ajla Abdukić*
Supervisor Bojan Mijatović†

Sarajevo School of Science and Technology
Faculty of Computer Science - Game Design and Development
Sarajevo, Bosnia and Herzegovina

Abstract

In this research paper, we present an empirical analysis of interaction methods in virtual reality (VR) simulations of extreme sports, with a specific focus on a rock-climbing simulation developed in Unity. Leveraging the Meta Quest controllers, hand tracking technologies, and TactGloves by bHaptics, this study aims to identify the most effective VR interaction modality that enhances user engagement, realism, and safety in simulated extreme sports environments. Through a comparative analysis of these interaction methods, the research investigates the potential of VR technologies to deliver immersive and realistic extreme sports experiences without the associated risks. The study employs a mixed-methods approach, combining quantitative performance metrics with qualitative user feedback to evaluate the efficacy of each interaction method in terms of immersion, usability, and user satisfaction. Preliminary results indicate that, contrary to initial expectations, hand tracking technologies provided users with a heightened sense of immersion compared to the advanced haptic gloves. This unexpected outcome, emerging from challenges encountered during the integration of bHaptics software, suggests that hand tracking might offer more promising avenues for training, rehabilitation, and entertainment in the realm of extreme sports VR simulations, and that more research is needed in field of haptic gloves. This paper contributes to the growing body of literature on VR interaction methods by providing insights into the benefits and limitations of various technologies, thereby informing future developments in VR simulations for extreme sports and beyond.

Keywords: VR Interaction, Extreme sport simulation, Virtual Reality Hand tracking, VR gloves, Haptic feedback

*ajla.abdukic@stu.ssst.edu.ba
†bojan.mijatovic@ssst.edu.ba

1 Introduction

Virtual Reality (VR) has ushered in a new era of digital interaction, enabling users to experience immersive environments with unprecedented realism. Specifically, in the context of climbing—one of the extreme sports—VR offers a safe yet exhilarating platform to mimic the intricate movements and psychological aspects associated with the sport. This research focuses on the climbing experience within VR, employing a simulation developed in Unity [1] to investigate how different interaction methods affect user engagement and performance.

In our climbing simulation, players can interact using VR controllers, hand tracking, and TactGloves by bHaptics [2]. Each method offers a distinct mode of interaction: VR controllers provide a standard, familiar interface; hand tracking offers intuitive, natural movements; and TactGloves deliver tactile feedback, simulating the texture and resistance one would feel when gripping real climbing holds.

Our primary aim is to determine which interaction method most effectively enhances the climbing experience in VR. We posit that a more immersive and interactive method can significantly improve the user's skill acquisition, strategy planning, and overall enjoyment. This is particularly relevant in a sport like climbing, where tactile feedback and precise movements are crucial.

The value of VR in simulating risk-laden sports like climbing extends beyond entertainment. It provides a platform for athletes to train, experiment with strategies, and refine their skills without the physical dangers associated with the sport. This aspect has been highlighted in previous studies, such as those by Döllinger et al. [3] and Sawade [4], who emphasize VR's potential to transform training and performance in extreme sports.

The haptic feedback provided by TactGloves is of particular interest due to its ability to replicate the tactile sensations of climbing, which are essential for a realistic and beneficial training experience. Studies by Lee et al. [5] and Patel et al. [6] support the notion that haptic feedback can significantly enhance spatial awareness and user interaction in VR, an idea further reinforced by the Haptic Fidelity Framework proposed by Muender et al. [7].

This research is structured to meticulously evaluate the different interaction methods within VR climbing simulations. After an in-depth review of VR technologies and their application in simulating extreme sports, we will analyze the specific contributions of VR controllers, hand tracking, and haptic feedback gloves. A comparative analysis, supported by a user study with our rock climbing simulation, will follow. We will conclude with discussions on our findings and their implications for future VR applications in extreme sports training and simulation.

2 Literature review

VR technologies, through their evolution, have sought to bridge the gap between virtual experiences and real-world sensations, a pursuit that has seen the development of various interaction methods aimed at enhancing user engagement and realism. This section delves into the related work surrounding extreme sports VR simulations, focusing on the interaction methods employed, their inherent advantages and drawbacks, and briefly compares these with the our approach.

2.1 Extreme Sports VR Simulations Projects

The application of VR in simulating extreme sports has grown in popularity, driven by the desire to safely replicate the thrill and challenge of sports like rock climbing within a virtual setting. Projects like the rowing simulation developed by Shoib et al. [8] and the robotic disaster response simulation by Agüero-Durán et al. [9] exemplify the diverse applications of VR in creating complex, interactive environments. Steindl's exploration of hybrid tracking technology [10] targets the accuracy and realism needed for virtual rock climbing simulations. The VreeClimber project, which combines a movable climbing wall with VR, offers a notable example of how VR can enhance the realism and safety of climbing simulations, by integrating hand tracking to maintain a realistic representation of the climber's movements [11]. This approach aligns with the principle that VR can significantly enhance skill acquisition and performance in complex tasks, as evidenced by Seymour et al. [12], directly applicable to extreme sports VR simulations. Pagé et al. utilized VR to improve decision-making skills in basketball, showcasing VR's potential in enhancing cognitive aspects of sports, which are crucial in navigating the challenging terrains in rock climbing [13].

2.2 Hardware Technologies in Use

In VR simulations, the hardware technologies range from traditional controllers to hand tracking and haptic gloves, each offering different levels of interaction fidelity. Controllers provide precision but may not replicate the naturalistic feel of climbing. Hand tracking technologies offer an intuitive interface, enabling users to maneuver in the virtual space in a more lifelike manner. However, they can sometimes be inaccurate and do not provide tactile feedback. Haptic gloves, particularly the bHaptics TactGloves [14], represent a significant leap forward by delivering detailed tactile responses, mirroring the textures and resistances encountered in actual rock climbing.

2.3 Drawbacks and Advantages

Each interaction method comes with its set of advantages and drawbacks. Traditional controllers are lauded for their reliability and precision but fall short in immersive quality. Hand tracking offers a more natural interaction experience but can suffer from inaccuracies and lacks tactile feedback [6]. Haptic gloves bridge these gaps by delivering precise tactile feedback, although they are not without challenges, including high production costs and integration complexities [6, 14]. The tactile feedback technology, particularly as implemented through devices like the TactGloves, provides a compelling solution to these issues by offering a more immersive and intuitive interaction method that closely mimics real-world sensations.

2.4 Comparison with Our Method

Our study employs the bHaptics TactGloves within a VR rock climbing simulation, diverging from previous studies that predominantly used controllers or hand tracking. Our approach capitalizes on the haptic gloves' advanced feedback mechanisms to enhance the climbing experience, offering a tactile dimension that closely resembles the real-world activity. This research aims to discern how tactile feedback influences user interaction and realism in VR, contrasting the experiences provided by TactGloves with those from other hardware technologies.

3 Case study

This section delves into the specifics of how our VR climbing simulation was constructed, detailing the creation of an immersive environment, the choice of location for the simulation, and how various elements were integrated to provide a realistic climbing experience.

3.1 Application design and structure

In our study, we developed an immersive VR environment based on Babin Zub, a towering, slender, and spiky rock formation that emerges into view after exiting an old Austrian tunnel near Sarajevo. Utilizing Unity, we employed a 360-degree camera image to construct a realistic skybox. Additional elements like foliage and grass were sourced from the Unity Asset Store [15], while rocks were custom modeled in Blender to enhance realism (see Figure 1).

Figure 1: The Babin Zub VR environment in Unity.

Figure 2: Climbable ball meshes.

Figure 3: Hovering over climbable mesh using VR Controllers.

To accurately model the rock surface, Blender [16] was utilized to sculpt the virtual representation of Babin Zub. The integration of the XR Interaction Toolkit enabled locomotion and interaction through various user actions. Climbable objects (see Figure 2), marked as meshes, were strategically placed to designate interactive points on the rock, guiding users through their virtual climbing experience.

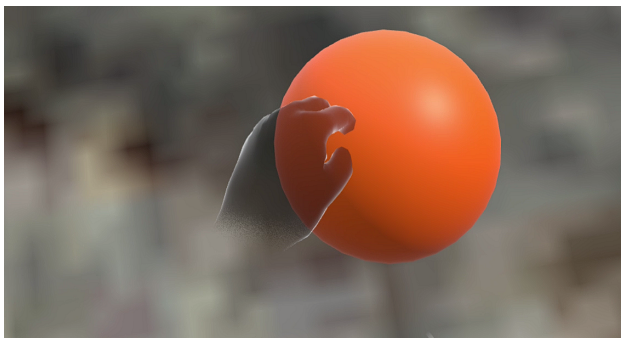


Figure 4: Grabbing the climbable mesh using Hand Tracking.

The interaction with these points is visually represented by color changes in the climbable ball meshes: white indicates an inactive state with no user interaction, blue appears when a user hovers over the climbable object, signaling readiness for interaction, and orange denotes that the user is actively engaging with the object, either by pressing and holding the trigger button or by using hand tracking to grasp the object (see Figures 3 and 4).

The bHaptics SDK is a pivotal component of the application, enabling haptic feedback inside the Unity editor. This SDK allows the application to communicate with bHaptics TactGloves, sending precise feedback based on users hand location within the virtual environment.

3.2 Implementation

The implementation phase involved integrating various components to enable a comprehensive VR climbing experience. The XR Toolkit facilitated the creation of an XR Rig, providing foundational support for locomotion methods such as continuous movement, teleportation, and, crucially, climbing. Hand tracking capabilities were introduced, allowing users to interact with the environment not only through VR controllers but also via natural hand movements, enhancing the immersive quality of the simulation (see Figure 5). Following the integration of these interaction functionalities, our attention turned to the environment's design. We implemented a procedural terrain system. This system allowed us to automate the placement of foliage, enabling us to distribute various trees and plants across the terrain seamlessly. By utilizing this procedural approach, we gained the flexibility to easily manipulate the ecosystem's composition, adjusting the density and variety of the vegetation to achieve the desired level of realism and environmental complexity (see Figure 6). After the addition of climbable objects, we advanced to the integration of haptic feedback in the VR environment. We imported the SDK from bHaptics and utilized their scripts to establish a connection between the climbable objects and the

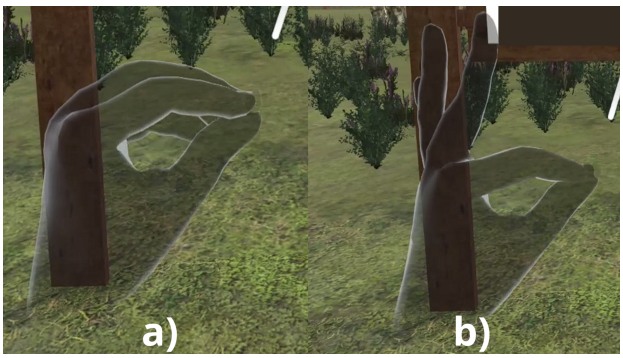


Figure 5: Grab activation using a)Hand Tracking and b)TactGloves.

VR controllers/hand tracking system. This was achieved by adding specific tags to the climbable meshes, which, when interacted with by the controllers or hands, would trigger haptic feedback, simulating the sensation of touching or gripping the objects.

Figure 6: Showcase of the realistic skybox and background.

Our next step focused on refining the user experience by implementing a heads-up display (HUD). This HUD plays a crucial role in guiding the player through the climbing experience, providing real-time feedback on their progress toward the goal or indicating if they have failed to complete a specific challenge. Recognizing the limitations of hand tracking technology, particularly the inability of cameras to detect hands covered by gloves, an experimental approach was taken to adapt the TactGloves. We developed custom scripts intending to enable the gloves to function akin to VR controllers, with simplified hand motions designated for 'grab' and 'move' functions. However, these adaptations faced challenges, as the responsiveness of the scripts did not meet the project's requirements. One of the most challenging aspects of the implementation was the integration of bHaptics TactGloves. The lack of readily available tutorials necessitated a deep dive into older documentation to locate the SDK for bHaptic products. The primary goal was to achieve haptic feedback upon interaction with climbable objects, simulating the tactile sensation of touching or gripping the rock surface. However, initial attempts to provide direct feedback to the specific hand engaging with an object encountered technical hurdles. Ultimately, a compromise was reached where both gloves would activate upon interaction, offering a uniform haptic response that, while not individually targeted, significantly enhanced the overall sense of touch within the simulation. Additionally, the experiment explored the handling of gravity within the VR environment. A glitch was identified wherein the physics engine did not consistently calculate falling gravity across interaction methods, leading to the disabling of fall mechanics during certain builds. Notably, the gravity fall feature functioned only when using the VR controller's joystick. This inadvertent design consequence had a silver lining: beginner VR users experienced a less discouraging introduction to VR climbing, as the absence of fall consequences reduced frustration and the likelihood of early cessation of the activity.

3.3 Interaction in application

The simulation's interaction design leverages various input methods, including VR controllers, hand tracking, and haptic gloves, to create a comprehensive and engaging user interface. The use of VR controllers is a standard mode of interaction, allowing users to navigate the virtual environment and interact with objects through familiar button presses and joystick movements. Specific actions, such as grabbing or initiating movement, are assigned to designated buttons (see Figure 7). The trigger button, for instance, is used to simulate the act of grasping climbable points on the rock, while the left joystick facilitates movement within the virtual space. The right joystick enables snap turn, and the B button (secondary button) is used for the restart function.

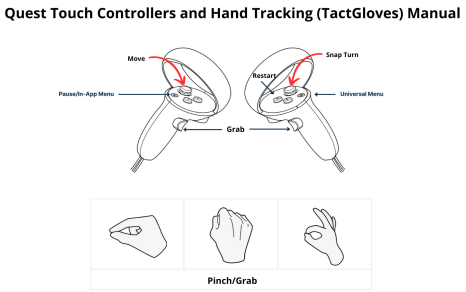


Figure 7: Manual for movement.

Hand tracking introduces a more naturalistic layer of interaction, enabling users to engage with the environment using their hand movements and gestures. This method allows for intuitive actions like pinching or reaching out to simulate grabbing and climbing without the need for physical controllers. Specifically, the pinch gesture is employed as the primary interaction mechanism for 'grabbing' in the virtual environment. This choice is under-

pinned by the gesture's distinct visibility to the headset's cameras. The pinch motion, characterized by bringing the thumb and a finger together, creates a clear and recognizable signal for the hand tracking system. The hand tracking system translates these real-world gestures into corresponding virtual actions, enhancing the immersion and realism of the climbing experience.

The integration of bHaptics TactGloves provides tactile feedback that corresponds to the user's virtual activities. The gloves are designed to deliver vibrating sensations that mimic the tactile experience of touching or gripping the rock surface. When a user interacts with a climbable mesh, the gloves activate, delivering feedback that enhances the perception of contact and grip. This haptic response is crucial for creating a convincing and immersive climbing experience, as it bridges the gap between visual input and physical sensation.

4 User experience evaluation

The simulation tasked participants with ascending a virtual model of Babin Zub, engaging with climbable objects via each interaction method. To ensure a systematic analysis, participants were instructed to utilize the interaction methods in a specific sequence: initially with VR controllers, subsequently through hand tracking, and finally employing hand tracking with the TactGloves. This sequence was intended to standardize the experiment's procedure and minimize any potential learning effects. Notably, the climbing task was not time-constrained, allowing participants to proceed at their own pace, thereby facilitating a more authentic assessment of each method's immersive quality and user-friendliness.

4.1 User study description

A total of 29 participants engaged in a sequential trial of the three interaction methods within a VR rock climbing simulation (see Figure 8). They started first with VR controllers, hand tracking and then hand tracking using TactGloves. The average testing lasted around 10 minutes per person. Following the interaction, participants completed a questionnaire assessing various aspects of their experience, including immersion, realism, ease of use, and physical comfort.

The user study was conducted in the VR laboratory at SSST[17], designed to ensure consistent conditions for all participants. The environment was equipped with a standard VR setup, including a high-performance computing system and a designated play area with adequate space for movement. Lighting and acoustics were optimized to minimize external distractions, ensuring that participants' experiences were solely influenced by the VR simulation. At the outset of the testing session, participants were thoroughly briefed on the rules and procedures. They were informed that they would be participating individually, which allowed for a focused and undisturbed experience. Additionally, they were assured of their autonomy during the experiment, with the explicit option to terminate their participation at any point should they experience discomfort, dizziness, or any other adverse effects, thereby prioritizing their safety and well-being.

Figure 8: The testing process for each user from using VR Controllers, Hand Tracking to TactGloves.

4.2 Results

Regarding immersion, participants' feedback highlighted a clear preference for hand tracking as the most immersive interaction method, with 48.7% of participants favoring it, compared to 35.1% for VR controllers and 16.2% for haptic gloves (see Figure 9). Despite the technological improvement of haptic gloves, their current implementation was less immersive for the majority of users, potentially due to the existing challenges in their integration and responsiveness.

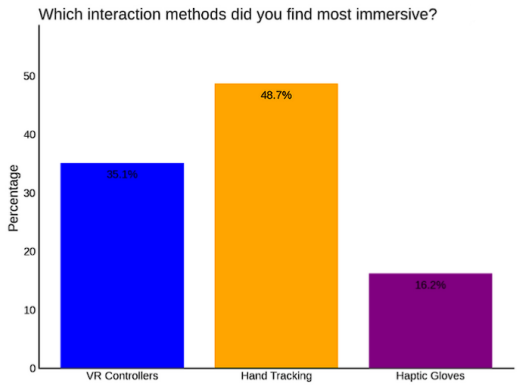


Figure 9: Distribution of participant preferences for immersive interaction methods.

The evaluation of interaction methods revealed distinct user preferences and experiences. While VR controllers were considered the most immersive by nearly half of the participants, the ease of use was notably higher for hand tracking, with 56% of participants finding it very easy to

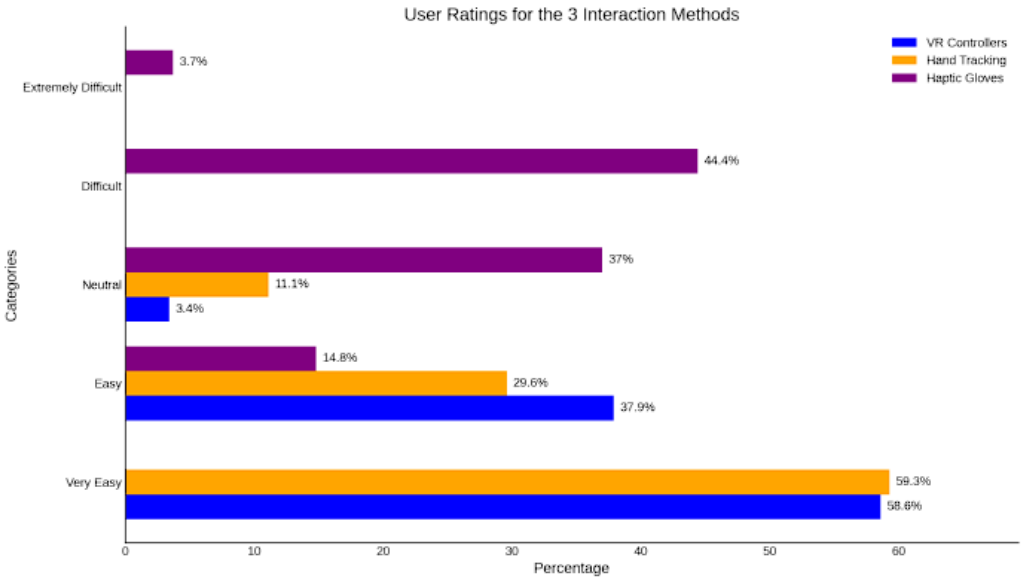


Figure 10: Rated ease of use for all three interaction methods using the Likert Scale.

use. In contrast, haptic gloves, despite their potential for enhanced tactile feedback, were rated as difficult by 48% of the participants (see Figure 10).

Haptic feedback, a core aspect of our study, showed promising yet mixed results. The majority of participants acknowledged the realism of haptic feedback when using controllers (66.7%) and gloves (64%). However, the feedback from gloves did not universally translate to a more immersive or preferred experience, highlighting the complexity of integrating tactile sensations in a manner that consistently enhances user experience (see Figure 11).

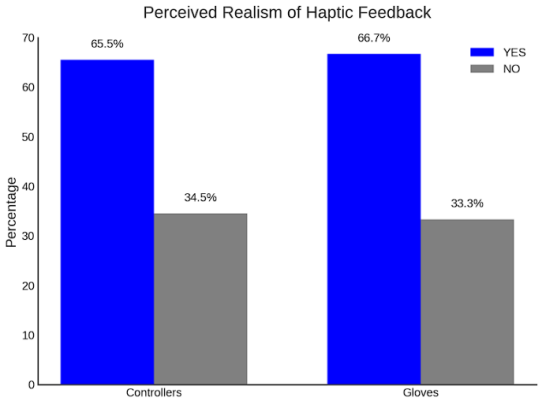


Figure 11: Graphic illustration of participant responses on the realism of haptic feedback using VR controllers and TactGloves.

Looking ahead, the majority of participants (90%) expressed interest in using haptic gloves for other simulations, suggesting a strong perceived potential for this technology, provided that issues related to responsiveness and tracking accuracy are addressed. This enthusiasm aligns with our focus on refining the gravity mechanics and developing more precise TactGlove implementations in future iterations of the simulation. By improving these aspects, we aim to enhance the realism and user engagement, potentially making haptic gloves a preferred method for interaction in VR simulations beyond rock climbing. Participants appreciated the overall experience, suggesting minor adjustments for enhanced realism and interaction quality. There were mentions of the potential for competitive elements and a desire for further development of the haptic gloves.

5 Conclusion and future work

The research conducted provided valuable insights into user experiences with different VR interaction methods in a rock climbing simulation. Through a comprehensive study the investigation shed light on the comparative effectiveness, realism, and user preferences associated with VR controllers, hand tracking, and bHaptics TactGloves. The findings indicated a general preference for hand tracking in terms of immersion and realism, though the haptic gloves offered unique tactile feedback that some users found more realistic and engaging.

Despite the innovative approach, the study unveiled challenges, particularly with the TactGloves and gravity mechanics within the simulation. The attempt to integrate the gloves as functional controllers highlighted the current technological limitations, affecting user experience and interaction precision. Furthermore, inconsistencies in gravity simulation revealed areas where the VR environment's realism could be enhanced. As for the future work we will concentrate on refining the gravity mechanics within the VR simulation to enhance realism and consistency. This

improvement is essential for a more immersive and authentic rock climbing experience. Additionally, the development will focus on improving the precision and responsiveness of the TactGloves. The aim is to achieve a more intuitive interaction, where users do not need to consciously adjust their hand positioning for the gloves to function effectively. Enhancing gesture recognition and sensor technology will be key to this advancement.

These enhancements, verified through further user testing, will not only elevate the user experience in the rock climbing simulation but could also inform interaction designs in other VR applications, expanding the impact of this research in the field of virtual reality.

References

[1] Unity real-time development platform — 3d, 2d, vr & ar engine. `https://unity.com/`. Accessed: 2024-01-21.

[2] Most advanced full body haptic suit - bhaptics tactsuit. `https://www.bhaptics.com/`. Accessed: 2024-01-10.

[3] N. Döllinger, E. Wolf, D. Mal, S. Wenninger, M. Botsch, M. Latoschik, and C. Wienrich. Resize me! exploring the user experience of embodied realistic modulatable avatars for body image intervention in virtual reality. *Frontiers in Virtual Reality*, 3, 2022.

[4] Caleb A. Sawade. *Learning Interventions in Olympic Skeleton Through the Use of Physical Simulation*. Doctoral thesis, University of Southampton, Engineering and the Environment, 2014.

[5] J. Lee, N. Rajeev, and A. Bhojan. Goldeye: Enhanced spatial awareness for the visually impaired using mixed reality and vibrotactile feedback. In *Proceedings of the 3rd ACM International Conference on Multimedia in Asia*, 2021.

[6] R. V. Patel, S. F. Atashzar, and M. Tavakoli. Haptic feedback and force-based teleoperation in surgical robotics. *Proceedings of the IEEE*, 110:1012–1027, 2022.

[7] T. Muender, M. Bonfert, A. V. Reinschluessel, R. Malaka, and T. Döring. Haptic fidelity framework: Defining the factors of realistic haptic feedback for virtual reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022.

[8] N. Shoib, M. S. Sunar, N. N. Nor, A. Azman, M. N. Jamaludin, and H. F. M. Latip. Rowing simulation using rower machine in virtual reality. In *2020 6th International Conference on Interactive Digital Media (ICIDM)*, pages 1–6, 2020.

[9] C. E. Agüero-Durán et al. Inside the virtual robotics challenge: Simulating real-time robotic disaster response. *IEEE Transactions on Automation Science and Engineering*, 12(2):494–506, 2015.

[10] Ludwig Steindl. *Hybrid Tracking Technology for Virtual Rock Climbing*. Diploma thesis, Vienna University of Technology, January 2018. Publication ID: 7149.

[11] Roman Voglhuber. *Hand Simulation for Virtual Climbing*. Diploma thesis, Vienna University of Technology, January 2019. Publication ID: 13747.

[12] Neal E Seymour, Anthony G Gallagher, Sanziana A Roman, Michael K O'Brien, Vipin K Bansal, Dana K Andersen, and Richard M Satava. Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Ann Surg*, 236(4):458–63; discussion 463–4, October 2002.

[13] C. Pagé, P. Bernier, and M. Trempe. Using video simulations and virtual reality to improve decision-making skills in basketball. *Journal of Sports Sciences*, 37:2403–2410, 2019.

[14] L. B. Rosenberg, J. E. Cha, and D. M. P. Kontarinis. Toward tactilely transparent gloves: Collocated slip sensing and vibrotactile actuation. In *World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE, 2011.

[15] Unity asset store. `https://assetstore.unity.com/`. Accessed: 2024-01-10.

[16] Blender. `https://www.blender.org/`. Accessed: 2024-02-13.

[17] Sarajevo school of science and technology. `https://www.ssst.edu.ba/`, 2023. Accessed: 2024-03-26.

Computer Vision

Optimal Crop-Out for Photographing People during Sporting Activities

Anastasia Lebedenko*
Supervised by: prof. Ing. Adam Herout Ph.D.†

Faculty of Information Technology
Brno University of Technology
Brno / Czech Republic

Abstract

This paper presents a solution for processing footage featuring human subjects to generate videos of optimal dimensions, focused on the individual, and eliminating redundant background. Utilizing computer vision models, the program identifies and tracks human positions in the input videos, then applies a specialized cropping algorithm to generate output frames. The solution offers customization options for aspect ratio, crop mode, and graphic overlay in the output video. Thus, it eliminates the necessity for capturing multiple videos to meet varied technical or aesthetic requirements, allowing the creation of diverse outputs from a single high-resolution video using predefined cropping parameters.

Keywords: computer vision, cropping algorithm, video processing

1 Introduction

Capturing videos of individuals in motion is challenging due to potential issues of exiting the frame or being disproportionately small compared to their environment [11]. The objective is to simplify the filming process by allowing users to capture one extensive video, and subsequently process it to meet diverse specifications, including adjusted frame size, aspect ratio, or focusing on specific segments of the human body. Such functionality enables generation of multiple customized video outputs from a single source file.

The solution requires developing an algorithm for precise Region of Interest (ROI) identification within each frame and a cropping strategy that ensures consistent positioning of the ROI across frames. The quality output video should appear stable from frame to frame, without visible jumps, that can be induced by frame cropping [12].

Existing video cropping solutions lack automation and comprehensive coverage of the human body (as discussed in Section 2).

*xlebed11@vutbr.cz
†herout@fit.vut.cz

The developed automated program, discussed in this paper, offers multiple cropping parameters and modes, ensuring the output video is stable and visually appealing. The solution eliminates the need for specialized recording equipment.

2 Existing Solutions

Incorporating the essential feature of cropping entire video clips, a number of video editing platforms, such as Final Cut Pro, extend the functionality to manually modify cropping parameters for individually segmented portions of video [6].

Adobe Premiere Pro employs an automatic AI-powered Auto Reframe feature [5], which crops footage to fit specified aspect ratios. This tool leverages motion tracking to accurately identify and maintain the visibility of the ROI throughout changes in frame resolution, ensuring critical elements remain within view in the output video. The process is predominantly automated, users are given the option to fine-tune the result by selecting among three predefined levels of camera motion intensity.

Apple's Center Stage feature [7] is a solution for real-time video crop. Available on select devices with an ultra-wide camera, it dynamically centers people on the camera preview, e.g. during video calls.

A state-of-the-art solution is Cloudinary API [2], that offers a large variety of crop modes as well as AI technology to gravitate video crop to the pre-determined ROI: faces or other user-specified objects. Despite its capabilities, this solution does not prioritize achieving an optimal frame size, which is the key feature of the proposed program. Moreover, it does not guarantee consistent inclusion of the entire subject within the frame or accommodate specific body capture orientations, such as portrait mode.

Reliance on AI for video processing, as highlighted in Adobe's documentation [5], may introduce artifacts upon recurrent processing of identical footage. Moreover, there currently exists no commercial or open-source program that replicates the unique approach of combining machine learning with direct mathematical cropping. The proposed

program offers a high degree of customization of processing parameters without the risk of significant artifacts.

3 Proposed Optimal Crop Algorithm

The proposed video processing algorithm, as shown in Figure 1, utilizes a two-phase architecture. In the scope of the initial video processing, it extracts body landmarks to generate bounding and frame box coordinates, and stores these 3 types of coordinates in separate JSON files with uniform structure, shown in Figure 1. This step, crucial due to its resource-intensive nature, ensures that landmark detection is only conducted once per video, thereby optimizing the cropping process for repeated crops of the same footage.

3.1 Detection

The program uses two MediaPipe detection solutions: Pose Landmark detection [9] and Object detection [8].

The program uses two MediaPipe detection solutions: Pose Landmark Detection and Object Detection. BlazePose, the underlying technology for Pose Landmark Detection, employs a lightweight convolutional neural network (CNN) architecture [1]. It combines heatmaps and regression to keypoint coordinates, enabling the detection of up to 33 body landmarks and the generation of a segmentation mask for a single person.

During inference, BlazePose adopts a detector-tracker setup. Initially, a body pose detector identifies the person in the frame. This is followed by a pose tracker network which predicts keypoint coordinates and refines the region of interest for accurate pose tracking. The detector focuses on detecting a relatively rigid body part, like the torso, using a fast on-device face detector as a proxy. This innovative method overcomes the limitations of traditional Non-Maximum Suppression algorithms, which often struggle with the complexity of human poses. The pose estimation network then predicts the location of 33 keypoints based on the alignment provided by the detector, effectively capturing complex human movements with high precision.

While segmentation mask is redundant in terms of human detection, it aids to more precise result, comparing to other human detection solutions, that return bounding boxes with excessive space on the edges [4, 10]. The landmarks are useful for cropping video based on the body capture orientations. Moreover, the chosen API succeeds in differentiating the most prominent person on the frame, which is useful for videos, where individual sport is performed with audience in the background. On the other hand, this mechanism is not fit for partner sports, as the set of landmarks will be calculated for just one person.

For partner sports, e.g. dancing, the MediaPipe Object Detection API is used. Detector output includes a name of object category e.g "human" and dimensions of the detected bounding box. The API also proved useful in the experiments with cropping a video of a person together with sporting equipment (e.g. cycling - the output video was cropped based on the combined position of the person and bicycle)

3.2 Bounding Box Calculation

In one-person mode, the bounding box is a rectangle that encloses the contour of the segmentation mask, returned by the detector, as shown in Figure 2. In case only a part of body is needed for the crop, the lower boundary of the segmentation mask bounding rectangle is cropped based on the y-coordinate of the relevant body landmark (Figure 3).

In case of partner sports, the final bounding box is obtained by summing up the bounding boxes of the relevant classes ("human" is default, classes with the names of sporting equipment are optional).

3.3 Frame Box Calculation

As the human moves, the dimensions of the corresponding bounding box can change from frame to frame. To ensure all video frames in the cropped output maintain constant dimensions, calculating the frame boxes is crucial. The exact size of the frame box depends on the chosen crop mode (Section 3.4). In case of a fixed frame, the output video size is defined by the coordinates, that enclose the area where a human was present at some point throughout the whole video. Otherwise, the largest width and height value among all bounding boxes define the size of an output video, with regard to if a specific aspect ratio was chosen. As per Figure 4, the frame box coordinates are calculated such that the bounding box is centered inside it. Figure 5 showcases the example of such positioning on a sample video frame.

Aspect Ratio

Adjusting to specific aspect ratios during cropping can indeed present a complex challenge, necessitating a variety of methods as noted by existing research [3]. Nonetheless, presented program simplifies this process significantly. It allows for selective inclusion of the surrounding environment by altering the dimensions of the frame box around the bounding box, thus eliminating the need for the complex methodologies typically required. This approach provides flexibility in determining the extent and specific areas to be included around the subject, facilitating a more intuitive and efficient cropping process.

Edge Cases

For frames like in Figure 6a, when person is moving towards the edge of the frame, bounding box centering results in frame box values exceeding the dimensions of in-

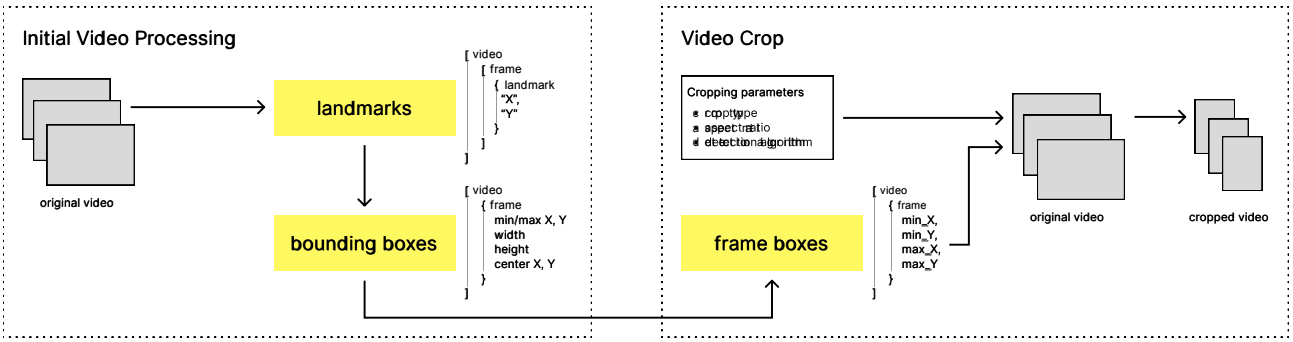


Figure 1: Architecture of the proposed solution. The first phase (**Initial Video Processing**) is executed once per unique video, and saves the output of processing (landmarks, bounding and frames boxes' coordinates). The second phase (**Video Crop**) utilizes the processed data to crop the original video based on the user-defined cropping parameters.

Figure 2: Segmentation mask with overlaid rectangle, which coordinates were calculated based on the edges of mask's contour. As a result, an optimal bounding box around the entire human body is found.

Figure 3: Bounding box, derived from the segmentation mask, cropped based on the y-coordinate of hips pose landmark.

put video. In this case, centering constraint is not included in the calculations.

For frames to be cropped and extracted from input video, each frame has to have frame box values defined. As shown in Figure 6b, if frame $n+1$ is missing frame box coordinates, these values are iteratively propagated from frame n and vice versa.

Stabilization

To ensure stable output video, crop-out values are filtered using Savitzky-Golay filter from SciPy library [13]. An array of each frame box coordinate's values in each frame is processed by savgol_filter( ) function.

3.4 Crop Modes

Crop modes are special crop settings implemented in the program, appropriate for specific type of movements in the input footage. Yoga videos, where person remains on the same place, could be cropped by **fixed frame**, which signifies the border, inside which all action is happening. For movements, that are primarily up and down, or left to right, **one-direction** cropping mode eliminates frame fluctuations in the secondary direction. Default **two-direction** mode can be combined with **zoom** option: in footage, where person distances from the camera, such mode zooms the frame in and out, so that person appears to be in the same distance.

4 Desktop Program

The solution is a Python-based desktop application that operates via a command-line interface, facilitating the cropping of video files through various parameters: input video(s) path, cropping mode, aspect ratio, body capture orientation (ranging from head-shot to full body), and optional graphical overlays (including detected landmarks, bounding, and frame boxes). Capable of batch processing entire directories, the program is designed for efficient handling of extensive video datasets.

5 Conclusion and Future Work

The solution effectively executes video cropping tasks across a wide array of video types, including scenarios featuring single or multiple subjects, with or without background activity. The qualitative evaluation of the desktop program is still in progress, with a primary focus on gathering user feedback regarding the appropriateness of different crop modes for different sports.

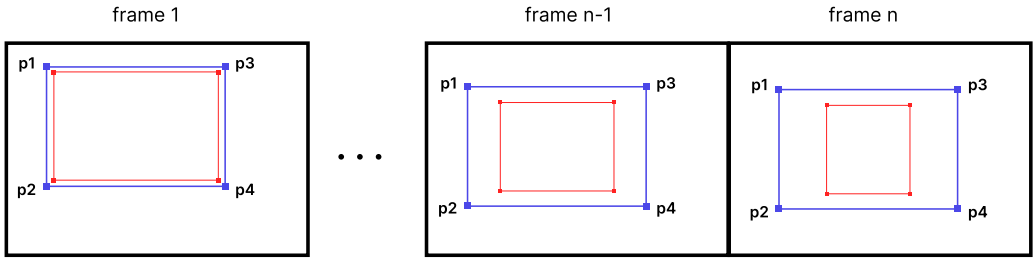


Figure 4: The dimensions of the frame box are determined by the largest values of width and height observed across all bounding boxes. For every frame that is cropped, the frame box is strategically positioned to ensure the bounding box remains centered within it.

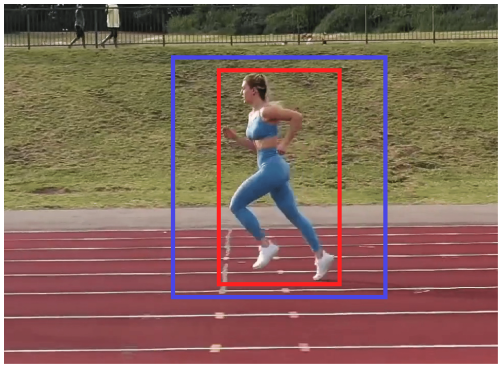


Figure 5: Bounding box and frame box on a sample frame.

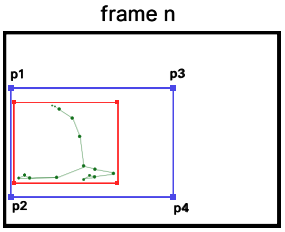


Figure 6a: Bounding box on the edge of input video, removal of centering constraint.

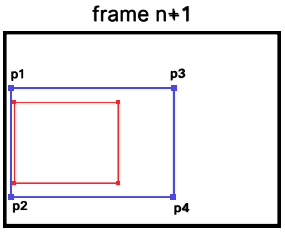


Figure 6b: Landmarks not detected, duplicate of previous frame box used.

While the program excels in its primary function of video cropping tailored to body dimensions, it lacks the comprehensive features of a full-fledged video editor, positioning it as a single-purpose tool ideal for batch processing, particularly in research contexts. It accepts user inputs through command-line arguments without providing a graphical user interface (GUI).

For convenient crop of videos, captured on smartphone, a logical improvement of the solution is development of a mobile application. Efforts will concentrate on integrating video cropping functionalities as well as devising a user-friendly interface that addresses the challenges of displaying complex cropping parameters on limited screen sizes, ensuring intuitive and visual editing workflows. Additional considerations include automatic video selection from device galleries, personalized cropping recommendations based on the video's characteristics or previous user settings, aiming to streamline the video cropping experience for end-users.

References

[1] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. Blazepose: On-device real-time body pose tracking, 2020.

[2] Cloudinary. Video resizing and cropping. `https://cloudinary.com/documentation/video_resizing_and_cropping`. Accessed on 07.03.2024.

[3] Zhongliang Deng, Yandong Guo, Xiaodong Gu, Zhibo Chen, Quqing Chen, and Charles Wang. A comparative review of aspect ratio conversion methods. In *2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008)*, pages 114–117, 2008.

[4] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[5] Adobe Inc. Automatically reframe video for social media channels. `https://helpx.adobe.com/premiere-pro/using/auto-reframe.html`. Accessed on 07.03.2024.

[6] Apple Inc. Crop clips in final cut pro for mac. `https://support.apple.com/cs-cz/guide/final-cut-pro/verb8e5db98/mac`. Accessed on 07.03.2024.

[7] Apple Inc. Use center stage on your ipad or studio display. `https://support.apple.com/en-us/HT212315`. Accessed on 07.03.2024.

[8] MediaPipe. Object detection task guide. https://developers.google.com/mediapipe/solutions/vision/object_detector/. Accessed on 07.03.2024.

[9] MediaPipe. Pose landmark detection guide. https://google.github.io/mediapipe/solutions/pose.html. Accessed on 07.03.2024.

[10] Duc Thanh Nguyen, Wanqing Li, and Philip O. Ogunbona. Human detection from images and videos: A survey. *Pattern Recognition*, 51:148–175, 2016.

[11] Manoj Ramanathan, Wei-Yun Yau, and Eam Khwang Teoh. Human action recognition with video data: Research and evaluation challenges. *IEEE Transactions on Human-Machine Systems*, 44(5):650–663, 2014.

[12] Yufei Xu, Jing Zhang, and Dacheng Tao. Out-of-boundary view synthesis towards full-frame video stabilization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4822–4831, 2021.

[13] Çağatay Candan and Hakan Inan. A unified framework for derivation and implementation of savitzky–golay filters. *Signal Processing*, 104:203–211, 2014.

Self-supervised Learning of Spatial Object Positioning in Football

Matúš Baran*
Supervised by: Igor Jánoš†

Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Bratislava / Slovakia

Abstract

We introduce a pretext task for self-supervised learning of feature extraction on an unlabeled dataset of football images. The task is based on predicting the relative distance between two random crops from the same image, which requires the model to understand the spatial positioning of the objects and players in the image. We evaluate the feature extractor trained with the proposed pretext task on the SoccerNet action spotting challenge and compare it to the existing self-supervised method SimCLR. We demonstrate the effectiveness and generality of the proposed pretext task for learning relevant features of the football domain.

Keywords: Self-supervised, Feature extractor, Football

1 Introduction

Football arguably belongs among the most favorite sports in the world with millions of fans and players. With technological advances and improvements in machine learning algorithms, the tasks performed by humans have been automatized and simplified and this applies also to the football domain. There were many attempts to create a model that would understand the game to predict the winner [31, 32, 2], analyze the players [24, 23], or even substitute the role of a referee [3].

The recent works in self-supervised learning methods made huge advances in the field of computer vision by closing the gap to supervised learning [20], some of them even surpassing the supervised method [5]. The self-supervised methods like MoCo [21] and MoCov2 [8] proved to be very effective in extracting relevant features from the image by contrasting the features. Other works showed that the missing annotations in the dataset can be replaced by introducing a pretext task such as image rotation [19] or temporal frames shuffling [27]. The purpose of the pretext task is to force the model to learn relevant features on the prior layers that can then be transferred to other downstream tasks.

*xbaranm@stuba.sk
†igor.janos@stuba.sk

We introduce a pretext task for self-supervised feature extractor learning on the unlabelled dataset. The task is based on the spatial understanding of the image and does not rely on the batch size. We apply this task by training a feature extractor for the football domain on the unlabeled dataset and validate it by transferring the trained model to the downstream football task.

We consider the action spotting challenge from SoccerNet [15] as an appropriate task to evaluate our feature extractors. The goal of the task is to identify 17 football actions like a goal, foul, ball out of play, etc. in broadcasted football videos. The task allows us to exchange the used feature extractor while preserving the rest of the solution architecture. So by substituting the feature extractors, we can evaluate them with the resulting performance of the task.

To show the effectiveness of our method we compare the lightweight feature extractor model trained with our pretext task to the lightweight model trained with the existing self-supervised method SimCLR, and also to a bigger pre-trained model with substantially more parameters.

Our contributions are as follows:

- We introduce a pretext task based on the spatial understanding of the image content by predicting the relative distance between two random crops for the self-supervised learning of the feature extractor.
- We trained multiple feature extractors using the existing self-supervised method SimCLR and our method which we evaluated and compared using the SoccerNet action spotting challenge.

2 Related work

Many previous works focused on creating a pretext task that would replace the missing annotations. Noroozi and Favaro [28] created a pretext task inspired by the puzzle game jigsaw in which the original image is divided into nine evenly big crops and shuffled. The goal of the model is to solve the jigsaw puzzle by which the model learns features that are as representative and discriminative as possible.

Another pretext task which is based on the nine-part grid is defined as predicting a relative position of the crops

[14]. The nine crops are taken from the original image while preserving the grid structure with a little variance. The model is always given the central middle crop with one of the eight remaining neighbor crops. The model then has to predict the relative position of the second crop by specifying one of the eight directions represented by the numbers one to eight. It is therefore a classification task where only one option is correct. The distance between the crops is always relatively small as the crops are next to each other in contrast to our method where the crops are randomly sampled. This prevents learning features that are spread along the whole image from one end to another.

The contrastive methods SimCLR [6] and SimCLRv2 [7] rely on attracting the positive pairs represented by augmented views from the same image and repelling the negative pairs represented by augmented views from different images. This is done by applying the contrastive loss on the features extracted from the views while maximizing the similarity of the features from positive pairs and minimizing the similarity of negative pairs. The effectiveness of this method highly relies on big batch sizes which require adequate computational power and resources. As Lin et al. mentioned [26], there are cases where negative pairs from different images can be more similar than the positive pairs from the same image. For example, the two crops from opposite corners of the same image can both capture diametrally different content, and forcing them to have similar feature representations could be misleading.

Giancola et al. [15] proposed a benchmark dataset for football action spotting. Later the authors extended the SoccerNet dataset [11] and provided a baseline using their own NetVLAD++ [18] model. The authors provide the annotated dataset along with annual challenges [16, 10] doing which they promote the use of neural networks in the football domain.

Action spotting is a challenge to identify certain football actions within the temporal window of their occurrence in the video. It is a popular challenge with many submissions [30, 22, 13, 9] competing for the best result. We consider the action spotting task as the appropriate form of evaluation of our feature extractor as it focuses on the most interesting and common actions in football.

3 Data collection

Despite the recent advances in football dataset annotation [17], manual annotations are still needed. Therefore we decided to attack the problem of insufficient size and number of annotated datasets in the football domain by using a self-supervised method and train the model on unlabeled football data. As the process of annotating is often costly and always very time-consuming, there will be no need for the dataset to contain the annotations. In this case, we trade off the missing annotations for a larger dataset size.

When training an unsupervised or self-supervised model, a large dataset is a must. Therefore getting as much valid data as possible was our top priority. We focused on the replays of professional football matches and extracted the frames from these videos. Football is a dynamic sport where a lot can happen in a nick of time so we choose the frequency of the extraction to be two frames per second. This resulted in the final 12,085,293 images in the unlabeled dataset. As the main source of the videos was YouTube, we named the dataset YF (YoutubeFootball).

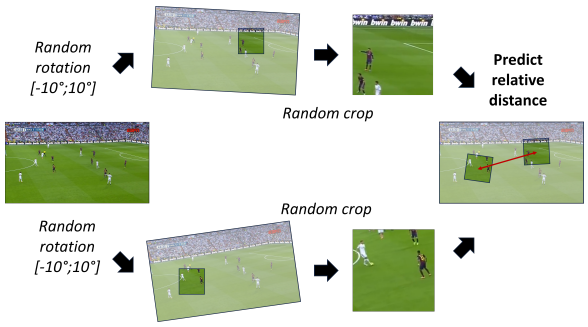


Figure 1: Illustration of our pretext task that is used for self-supervised training.

The images in the dataset do not strictly have to be consecutive as there is no additional information about which image is the start or the end of some video. So image *N + 1* does not have to be subsequent to image *N*. This fact constrains the pretext task to not rely on any temporal information which makes the pretext task more generic and applicable to other domains.

As we do not possess the author rights to the videos we can only publish the scripts for the image extraction and not the whole dataset.

4 Our pretext task

Most of the existing methods are trained and benchmarked on datasets [12, 25] that have very little in common with football. The fact that the YF dataset consists of football images only can be used as an advantage when creating the new self-supervised method.

Our pretext task focuses on understanding the spatial positioning of the objects in the images by predicting their relative distance from each other. This is done by extracting two random crops of the same size from the same image and measuring the relative distance between their centers.

Before the crops are taken from the image, the image is rotated by a random degree. For each crop, a new random number is used from the interval from -10 to 10 degrees. The rotation is done around the center of the image. After the rotation is applied, the resulting image is still rectangular, but a dark background is created to fill the blank spaces around the rotated edges. To end up with the image containing only the valid content of the image a crop that represents the largest possible rectangle that excludes the

Figure 2: After the rotation is applied, corner spaces around the image are filled with default dark color. To end up purely with valid data containing the content of the image a crop is performed, representing the largest possible rectangle with content omitting the filled spaces created by rotation.

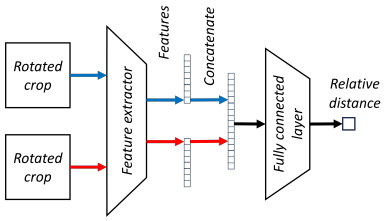


Figure 3: Architecture of our pretext task. The features are concatenated into the dense layer which outputs the relative distance.

dark background from the rotation needs to be performed. To better understand this process, figure 2 visually shows the adapted solution to effectively end up with valid data after augmenting the original image.

After the images are rotated and cropped to contain the biggest possible content, random coordinates are selected to represent the center of the final crop in each of the rotated images. The range that the coordinates are taken from is calculated so that the randomly taken coordinate is not located near the edge of the image which would result in an incomplete image crop since the part of the crop could exceed the rotated image. This technique ensures that the final crop will always contain valid data. On the other hand, the rotation of the image and the aforementioned cropping result also in omitting some valid parts of the original image that will not be used when performing the final crops. While this is true in most cases, in a case when the rotation degree is zero the full image is available for the final crop and no data is omitted before the final crop.

Since both of the crops are extracted from the images that could be rotated by a different degree, the coordinates of their centers are recomputed to match the exact same points in the original non-rotated image. The relative distance between the crops is computed as the distance between the centers of the crops divided by the size of the crop, all in pixel units. So if two crops were both from the non-rotated images(rotation angle zero degrees) and were right next to each other meaning they have one common edge, their relative distance would be exactly one.

This relative distance is created for every image during the training so the pseudo-labels are created on the fly and the task for the model is to predict this relative distance. Figure 1 illustrates our pretext task and figure 3 illustrates the architecture of the model using our pretext task.

Our method is different from the previous position prediction pretext task [14] as it offers more variations in the resulting pseudo-label because the predicted value is not limited by some set of values. Also, the crops are taken randomly and there is no restriction on their positioning, meaning that they can be next to each other, or one under the other, or anywhere in the image. The gap between them also varies so the model must learn not only local similarities when the crops are right next to each other but also be aware of the global context when the crops are on the opposite corners of the image. There is also no restriction on whether the crops can overlap or not.

To be able to accurately predict the distance between two parts of the image some knowledge about the context must be known that can be derived from the content of the two crops. In football, the positioning on the pitch is very important. It can say a lot about the style of the play of one team or the current situation in the game, whether is the team attacking or defending. The positioning of the players is very important also because of the football rules. Mainly because of one particular rule, which is offside [4]. In football, a player is offside if they are closer to the opponent's goal line than both the ball and the second-last opponent when the ball is played to them. So the understanding of the positioning is even more important in the football domain.

Therefore using our proposed pretext task the model should learn to understand the complex positioning of the players and the ball on the pitch. This however applies not only to the game itself but also to replays from other perspectives and other actions connected to the game as substitutions, in-game medical treatment, and many more. So when the model is trained to be relatively accurate when predicting the relative distance between two crops from the football image, it must possess some deeper knowledge and understanding of the football positioning itself. This implies selecting more valuable features from the early layers and in the end better feature extraction.

To make the task more challenging the rotation of the image by up to ten degrees is applied. When looking at the images in the figure 2 we can visually understand what

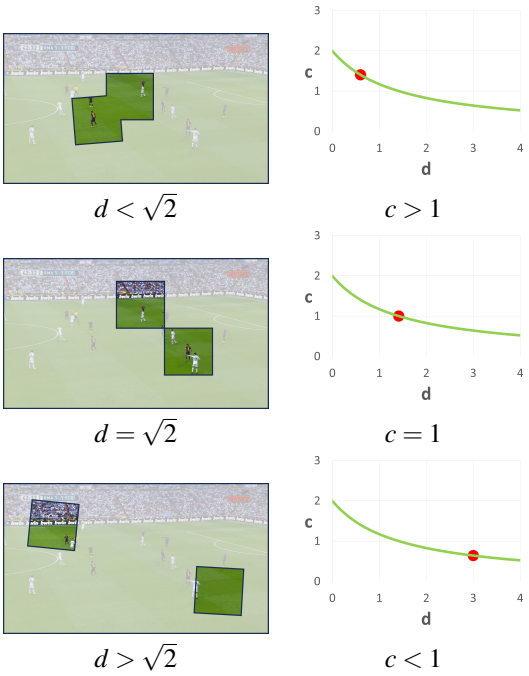


Figure 4: Illustration of possible placement of the crops, where d represents the distance and c represents the coefficient.

is happening in the picture even if it is slightly rotated. In convolutional neural networks, however, even a slight rotation can cause different outcomes as the convolutional filters are very sensitive to rotated input [29, 1]. By rotating the input crops, this sensitivity is attacked and the neural network is forced to learn and understand the images more in a way that humans understand them.

Another advantage of our pretext task is its generality. Since it does not rely on any particular information related directly to football it can be applied to other datasets as well.

4.1 Customised loss function

Predicting the relative distance of the crops from the image can be a demanding task when the crops are from opposite parts of the image since the content on one side could be wholly different than the content on the other side. There could be not so many if any clues in the crops for predicting the right distance between such crops. On the other hand, it is much easier to predict the distance if the crops are overlapping and have some common parts. The parts in common could hint that the crops are close to each other and by the size of the overlapping part, it could easily be determined how far away are the centers of the crops.

Because it is not always the same difficulty to predict the distance of the crops based only on the content of the crops without any context, we scale the loss calculated from the predicted distance based on the distance of the crops. If the crops are close to each other or even overlapping, the

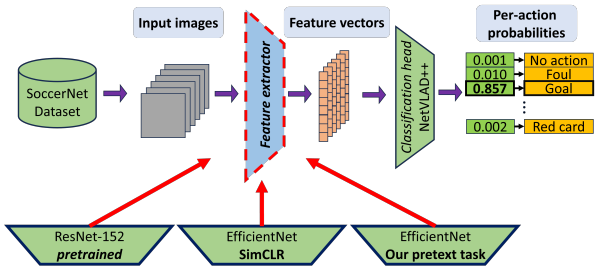


Figure 5: Action spotting pipeline with various feature extractor models. The classification head predicts the per-class probabilities for each action.

neural network should easily determine their relative distance and therefore it will be additionally penalized if it makes a mistake in such an "easy" case. If the crops are far away from each other it is way more difficult to predict the exact distance between the crops and therefore if the neural network makes a mistake in such a "hard" case the resulting mistake will be reduced.

Figure 4 shows three scenarios that can occur when creating the crops from the image. In the first case, the crops are overlapping and their relative distance is less than $\sqrt{2}$. In the second case, the crops have exactly one corner in common and their relative distance is equal to $\sqrt{2}$. In the third case, the crops have no area in common and their relative distance is greater than $\sqrt{2}$.

To adjust the loss or the mistake that the neural network makes, a coefficient is used which is calculated with the formula 1. The coefficient is dependent on the distance of the crops. The α and β are coefficients with default values $\sqrt{8}$ and $\sqrt{2}$ respectively and d is the relative distance of the crops. The final loss (l) is calculated as the product of the distance error (e) and the coefficient (c) as can be seen in the formula 2. The smaller the distance between the crops is the bigger the coefficient is and therefore the final loss will be also bigger. When the distance is bigger, the coefficient and the final loss will be smaller.

$$c = \frac{\sqrt{\alpha}}{d + \sqrt{\beta}} \tag{1}$$

$$l = e * c \tag{2}$$

The default values for the coefficients are set according to the illustration in the figure 4. In the second case when the crops are diagonally next to each other, the computed coefficient will have value 1 and therefore will not affect the final loss. This serves as a reference scenario where it should be reasonably difficult to predict the distance of the crops. If the crops are closer to each other, the coefficient will be greater than 1 and if the crops are further from each other the coefficient will be less than 1.

5 Evaluation metric

To be able to evaluate the trained feature extraction models and compare our method to the existing self-supervised method we need a qualitative metric that will give us some score for both models. As the pretext tasks used in these methods were different we could not use the training or validation loss as a valid metric for comparison. Instead, we used the action spotting task from SoccerNet which takes the broadcast videos of professional football matches and evaluates the precision of identifying specific football actions.

The model must predict the exact timestamp when the action occurs and the prediction must land within a tolerance δ around the ground truth anchor. The tolerance varies from 5 to 60 seconds with 5-second steps. Recall, precision, and Average Precision (AP) are computed for each given class and a mean Average Precision (mAP) is computed across all classes. An average-mAP is computed across all δ tolerances. The average-mAP metric, together with the *average-mAP visible* for visible actions and *average-mAP unshown* for actions that happen out of the camera range, are used for the evaluation of the models' performances in the SoccerNet action spotting.

The process of training a classification model for action spotting is illustrated in figure 5. The process consists of extracting the features from the videos and training the classification head on the extracted features. The feature extraction is a separate process that allows for modifying it by substituting the feature extractor which then yields different feature vectors.

No architectural or other changes are needed for the classification head which is every time trained from scratch on the given features. The result achieved by the classification head therefore relies on the extracted features. So when the result of a classification head trained on features extracted by one model is better than the result of a classification head trained on features extracted by the second model, we can say that the first feature extraction model is better than the second.

Figure 5 shows the integration of feature extractors trained with the SimCLR method and also our pretext task into the SoccerNet training pipeline. The values of per-action probabilities on the output of the two figures are illustrative but they symbolize that the classification head trained on different features yields different predictions which end up in different accuracy and precision.

By comparing the average-mAP of the classification heads, which is the metric used in SoccerNet action spotting, we were able to compare the performance and ability of the feature extractors to extract the relevant features from the football videos. As the architecture of the classification head remains always the same, its average-mAP is used as the qualitative metric for evaluating the feature extractors.

6 Results

We trained multiple feature extractors on the YF dataset using the existing self-supervised method SimCLR and our pretext task. The augmentations used in the SimCLR method were random horizontal flip, random resized crop, color jitter, random grayscale, and Gaussian blur, similar to the original paper. We trained the SimCLR models with a learning rate of 10^{-4}, batch size of 120, and cosine annealing scheduler without restart. We used the NT-Xent loss with a temperature of 0.7.

As for our pretext task, we used the same batch size as with SimCLR, but we used a constant learning rate of 10^{-4} together with adjusted MSE loss as discussed in the section 4.1.

The training time of one feature extractor trained on the subset of the YF dataset was about one week for both the SimCLR and our method. Training on the whole dataset took two to three weeks for each feature extractor. All training runs were executed using one NVIDIA RTX3090 GPU.

Throughout the training, we performed evaluations on the SoccerNet action spotting task, which we used as a metric for the evaluation of feature extractors for the football domain.

As can be seen in table 1, the feature extractors trained on the YF dataset did not outperform the pretrained feature extractors from the ImageNet, however, the feature extractor trained with the SimCLR method did not get behind by much, as the difference between the best model is only 2.71%. The feature extractor trained with our pretext task did not perform badly neither. The a_mAP score of 41.13% did prove that the method helps to learn to extract relevant features for the football domain, however, it does not reach the level of the existing self-supervised method. Further research focusing on finding the optimal hyperparameters of our pretext task could improve its performance.

In table 2 we show the per-class results of the NetVLAD++ model trained on features extracted by the best extractor trained with the SimCLR and our pretext task. For reference, we show also the results of the model trained on the features provided by SoccerNet that were extracted using ResNET-152 and features extracted with the pretrained Efficient-B0 model. The table shows that the model trained on the features extracted with the SimCLR model outperformed the pretrained EfficientNet-B0 in 4 classes and even outperformed the pretrained ResNET-152 in one class. The feature extractor trained with our method did not reach the best result in any class and the result margins were similar to the a_mAP results.

7 Future work

Our pretext task uses the customized loss function that contains two hyperparameters *alpha* and *beta*, which in-

Backbone	# params	Dataset	Train method	# Images seen /	unique	Head	a_mAP all	a_mAP visible	a_mAP unshown
ResNET-152*	60M	ImageNet	*pretrained*	- /	-	NetVLAD++	**52.73**	**59.07**	**36.59**
EfficientNet-B0	5.3M	ImageNet	*pretrained*	- /	-	NetVLAD++	52.17	58.79	35.61
EfficientNet-B0	5.3M	YF[000-002.sqsh]	SimCLR	110 625 000 /	375 000	NetVLAD++	**49.02**	53.87	**33.38**
EfficientNet-B0	5.3M	YF	SimCLR	69 806 310 /	11 634 385	NetVLAD++	48.84	**54.63**	33.00
EfficientNet-B0	5.3M	YF[000-002.sqsh]	**Our pretext**	56 625 000 /	375 000	NetVLAD++	39.13	44.74	29.96
EfficientNet-B0	5.3M	YF	**Our pretext**	255 956 470 /	11 634 385	NetVLAD++	41.13	46.14	30.41

Table 1: Best results of various feature extractors. The first run(marked with an asterisk *) is executed on the provided features from the SoccerNet. Other runs are executed with the use of a smaller model EfficientNet-B0. YF[000-002.sqsh] represents a small subset of the YF dataset.

Feature extractor	**# params**	**SoccerNet-v2**	visible	unshown	Ball out	Throw-in	Foul	Ind. free-kick	Clearance	Shots on tar.	Shots off tar.	Corner	Substitution	Kick-off	Yellow card	Offside	Dir. free-kick	Goal	Penalty	Yel. → Red	Red card
ResNET-152 [pretrained]	60M	52.7	59.1	36.6	74.9	58.5	73.4	69.2	36.0	39.2	40.0	56.7	70.1	68.5	64.5	43.8	57.9	79.7	54.9	3.9	5.2
EfficientNet-B0 [pretrained]	5.3M	52.2	58.8	35.6	70.2	58.7	66.7	67.3	38.7	36.5	40.0	54.2	68.7	67.5	63.2	44.3	58.4	80.9	62.8	5.4	3.4
EfficientNet-B0 [SimCLR]	5.3M	49.0	53.9	33.4	62.4	51.8	65.6	69.1	29.0	36.6	39.0	56.3	68.9	63.6	61.9	43.5	52.0	79.3	39.3	13.3	1.7
EfficientNet-B0 [Our pretext]	5.3M	41.1	46.1	30.4	36.0	39.6	49.5	60.3	22.0	33.6	36.6	52.2	66.3	58.9	54.0	38.3	39.7	77.0	33.6	0.6	1.0

Table 2: Mean average precision of the NetVLAD++ model on features extracted by various feature extractors on SoccerNet action spotting.

fluence the training process of the model. Initial values of these hyperparameters that were used are not optimized and future work could include finding the optimal values of these hyperparameters, which could lead to better performance of the method. Identically the optimal value for the maximal rotation of the image could improve the method and lead to better results.

The comparison of a lightweight feature extractor trained with our pretext task to the existing self-supervised method and a substantially bigger model showed the potential of our method. Models with more parameters tend to achieve better results because of their higher learning capacity, so possible future work includes training a bigger feature extractor using our method and comparing it to the ResNET-152 originally used in SoccerNet.

Since our pretext task does not rely on any information about the dataset or the football domain, it can be used in other domains and downstream tasks as well. An example can be the replay grounding task from SoccerNet which also uses extracted features from the SoccerNet dataset to identify replayed actions in broadcasted football matches or another non-football-related downstream task such as image classification benchmark in ImageNet.

All feature extractors in this work were evaluated on the SoccerNet action spotting using only the NetVLAD++ classification head. Using other models as a classification head can yield even better results in the SoccerNet action spotting challenge.

8 Conclusion

In this paper, we introduced a pretext task for self-supervised learning on an unlabelled dataset that is based on the spatial understanding of the image content. We trained multiple feature extractors with both our and an existing self-supervised method SimCLR which we evaluated on the SoccerNet dataset using the action spotting task.

With the same model representing the classification head and only varying the backbone, we showed that our method achieved an a_mAP of 41.13% in the action spotting task, which is 7.89% less compared to the existing self-supervised method SimCLR. The performance gap between the lightweight EfficientNet-B0 model trained with both SimCLR and our pretext task and a substantially bigger ResNET-152 model is relatively small compared to the number of parameters and learning capacity that they dispose of.

We hypothesize that using a bigger model together with our method can achieve even better results and overcome the pretrained ResNET-152 in the action spotting task. We show that our pretext task does not rely on any information about the dataset and therefore can be applied to other domains as well.

References

[1] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017.

[2] Azrel Aiman Azeman, Aida Mustapha, Nazim Razali, Aziz Nanthaamomphong, and Mohd Helmy Abd Wahab. Prediction of football matches results: Decision forest against neural networks. In

2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pages 1032–1035, 2021.

[3] Arik Badami, Mazen Kazi, Sajal Bansal, and Krishna Samdani. Review on video refereeing using computer vision in football. In *2018 IEEE Punecon*, pages 1–8, 2018.

[4] The International Football Association Board. Laws of the game 23/24. Accessed: 7-1-2024.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *CoRR*, abs/2006.09882, 2020.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. *CoRR*, abs/2006.10029, 2020.

[8] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020.

[9] Anthony Cioppa, Adrien Deliège, Silvio Giancola, Bernard Ghanem, Marc Van Droogenbroeck, Rikke Gade, and Thomas B. Moeslund. A context-aware loss function for action spotting in soccer videos. *CoRR*, abs/1912.01326, 2019.

[10] Anthony Cioppa, Silvio Giancola, Vladimir Somers, Floriane Magera, Xin Zhou, Hassan Mkhallati, Adrien Deliège, Jan Held, Carlos Hinojosa, Amir M. Mansourian, Pierre Miralles, Olivier Barnich, Christophe De Vleeschouwer, Alexandre Alahi, Bernard Ghanem, Marc Van Droogenbroeck, Abdullah Kamal, Adrien Maglo, Albert Clapés, Amr Abdelaziz, Artur Xarles, Astrid Orcesi, Atom Scott, Bin Liu, Byoungkwon Lim, Chen Chen, Fabian Deuser, Feng Yan, Fufu Yu, Gal Shitrit, Guanshuo Wang, Gyusik Choi, Hankyul Kim, Hao Guo, Hasby Fahrudin, Hidenari Koguchi, Håkan Ardö, Ibrahim Salah, Ido Yerushalmy, Iftikar Muhammad, Ikuma Uchida, Ishay Be'ery, Jaonary Rabarisoa, Jeongae Lee, Jiajun Fu, Jianqin Yin, Jinghang Xu, Jongho Nang, Julien Denize, Junjie Li, Junpei Zhang, Juntae Kim, Kamil Synowiec, Kenji Kobayashi, Kexin Zhang, Konrad Habel, Kota Nakajima, Licheng Jiao, Lin Ma, Lizhi Wang, Luping Wang, Menglong Li, Mengying Zhou, Mohamed Nasr, Mohamed Abdelwahed, Mykola Liashuha, Nikolay Falaleev, Norbert Oswald, Qiong Jia, Quoc-Cuong Pham, Ran Song, Romain Hérault, Rui Peng, Ruilong Chen, Ruixuan Liu, Ruslan Baikulov, Ryuto Fukushima, Sergio Escalera, Seungcheon Lee, Shimin Chen, Shouhong Ding, Taiga Someya, Thomas B. Moeslund, Tianjiao Li, Wei Shen, Wei Zhang, Wei Li, Wei Dai, Weixin Luo, Wending Zhao, Wenjie Zhang, Xinquan Yang, Yanbiao Ma, Yeeun Joo, Yingsen Zeng, Yiyang Gan, Yongqiang Zhu, Yujie Zhong, Zheng Ruan, Zhiheng Li, Zhijian Huang, and Ziyu Meng. Soccernet 2023 challenges results, 2023.

[11] Adrien Deliège, Anthony Cioppa, Silvio Giancola, Meisam Jamshidi Seikavandi, Jacob V. Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B. Moeslund, and Marc Van Droogenbroeck. Soccernet-v2 : A dataset and benchmarks for holistic understanding of broadcast soccer videos. *CoRR*, abs/2011.13367, 2020.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Julien Denize, Mykola Liashuha, Jaonary Rabarisoa, Astrid Orcesi, and Romain Hérault. Comedian: Self-supervised learning and knowledge distillation for action spotting using transformers, 2023.

[14] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015.

[15] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. *CoRR*, abs/1804.04527, 2018.

[16] Silvio Giancola, Anthony Cioppa, Adrien Deliège, Floriane Magera, Vladimir Somers, Le Kang, Xin Zhou, Olivier Barnich, Christophe De Vleeschouwer, Alexandre Alahi, Bernard Ghanem, Marc Van Droogenbroeck, Abdulrahman Darwish, Adrien Maglo, Albert Clapés, Andreas Luyts, Andrei Boiarov, Artur Xarles, Astrid Orcesi, Avijit Shah, Baoyu Fan, Bharath Comandur, Chen Chen, Chen Zhang, Chen Zhao, Chengzhi Lin, Cheuk-Yiu Chan, Chun Chuen Hui, Dengjie Li, Fan Yang, Fan Liang, Fang Da, Feng Yan, Fufu Yu, Guanshuo Wang, H. Anthony Chan, He Zhu, Hongwei Kan, Jiaming Chu, Jianming Hu, Jianyang Gu, Jin Chen, João V. B. Soares, Jonas Theiner, Jorge De Corte, José Henrique Brito, Jun Zhang, Junjie Li, Junwei Liang, Leqi Shen, Lin Ma, Lingchi Chen, Miguel Santos Marques, Mike Azatov, Nikita Kasatkin, Ning Wang, Qiong Jia, Quoc Cuong Pham, Ralph Ewerth, Ran Song, Rengang Li, Rikke Gade, Ruben Debien, Runze Zhang, Sangrok Lee,

Sergio Escalera, Shan Jiang, Shigeyuki Odashima, Shimin Chen, Shoichi Masui, Shouhong Ding, Sin-wai Chan, Siyu Chen, Tallal El-Shabrawy, Tao He, Thomas B. Moeslund, Wan-Chi Siu, Wei Zhang, Wei Li, Xiangwei Wang, Xiao Tan, Xiaochuan Li, Xiaolin Wei, Xiaoqing Ye, Xing Liu, Xinying Wang, Yandong Guo, Yaqian Zhao, Yi Yu, Yingying Li, Yue He, Yujie Zhong, Zhenhua Guo, and Zhiheng Li. Soccernet 2022 challenges results. In *Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports*, MM '22. ACM, October 2022.

[17] Silvio Giancola, Anthony Cioppa, Julia Georgieva, Johsan Billingham, Andreas Serner, Kerry Peek, Bernard Ghanem, and Marc Van Droogenbroeck. Towards active learning for action spotting in association football videos, 2023.

[18] Silvio Giancola and Bernard Ghanem. Temporally-aware feature pooling for action spotting in soccer broadcasts. *CoRR*, abs/2104.06779, 2021.

[19] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018.

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *CoRR*, abs/2006.07733, 2020.

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722, 2019.

[22] James Hong, Haotian Zhang, Michaël Gharbi, Matthew Fisher, and Kayvon Fatahalian. Spotting temporally precise, fine-grained events in video, 2022.

[23] Xin Hu. Football player posture detection method combining foreground detection and neural networks. *Scientific Programming*, 2021:4102294, June 2021.

[24] Mihnea Bogdan Jurca and Ion Giosan. A modern approach for positional football analysis using computer vision. In *2022 IEEE 18th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 275–282, 2022.

[25] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[26] Wenye Lin, Yifeng Ding, Zhixiong Cao, and Hai tao Zheng. Establishing a stronger baseline for lightweight contrastive models, 2023.

[27] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Unsupervised learning using sequential verification for action recognition. *CoRR*, abs/1603.08561, 2016.

[28] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *CoRR*, abs/1603.09246, 2016.

[29] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, 2015.

[30] João V. B. Soares and Avijit Shah. Action spotting using dense detection anchors revisited: Submission to the soccernet challenge 2022, 2022.

[31] Johannes Stübinger, Benedikt Mangold, and Julian Knoll. Machine learning in football betting: Prediction of match results based on player characteristics. *Applied Sciences*, 10(1), 2020.

[32] Ekansh Tiwari, Prasanjit Sardar, and Sarika Jain. Football match result prediction using neural networks and deep learning. In *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 229–231, 2020.

Capturing of Detailed and Very Large Photograph and Localization Within

Bc. Pavol Dubovec*
Supervised by: prof. Adam Herout PhD.†

Department of computer graphics and multimedia
Brno University of Technology
Brno / Czech Republic

Abstract

This paper presents a new technique for locating a photograph within a larger one, with the aim of enhancing the speed and accuracy of conventional methods. The proposed technique utilises a CNN architecture to extract multiple embeddings from the query image[1], which are then used to perform an approximate search within a database of embeddings from the large photograph. Two main models were trained on a large dataset. The first model used a triplet loss function, while the second model used a cross-entropy loss function. Conventional methods were used to determine the location of the images in the training set and to generate a large image. A database of embeddings was created by partitioning the large photograph with a certain sampling frequency (in pixels) using the trained model. The database is queried for K-nearest sub-query[2] embeddings. These embeddings are generated by partitioning the query image into equal-sized pieces as CNN inputs. The optimal homography model is determined through random sampling based on the positions of four sub-query images and their corresponding positions in the large image. The model homography with the lowest harmonic mean embedding distance is selected as the resulting position. The method demonstrates satisfactory accuracy and good speed on the generated test datasets. The best model achieved a top-1 accuracy of 97.71% and a top-3 accuracy of 99.17%. Future research will investigate the method's performance with increasing surface heterogeneity, the potential for automating video retrieval to obtain a large dataset of photos, and its effectiveness for photo localization in cases where conventional methods fail due to a lack of key points.

Keywords: Image Localization, Homography Estimation, Approximate Search

1 Introduction

This article discusses solutions to two common problems in computer vision and graphics: image localization and image stitching. The conventional method for addressing the localization problem involves detecting keypoints, extracting local features (descriptors) around these keypoints, matching the extracted features from the query image with features from a large image (map), and then using the matched keypoints to estimate homography. This homography can then be refined using bundle adjustment, which minimises the reprojection error. The methods themselves are explained in the sections 2 and 4. These methods rely on handcrafted keypoints and homography matrices, which use robust fitting methods such as RANSAC[7] or LMS[20]. However, they may perform poorly when the percentage of inliers falls below 50% [13]. These methods often use Hough transform [11, 2] to overcome this problem. This problem is particularly significant in cases where it is necessary to locate a very small picture within a much larger one, especially when there are very few common features.The aim of this study is to explore a novel approach to determine position based on a CNN-generated model to create an embedding database for a large image. This database is then used to locate a photography within the large photography. The process involves dividing the input image into smaller sub-queries, determining the embedding for each sub-query, and using a random sampling algorithm to create a homogra-

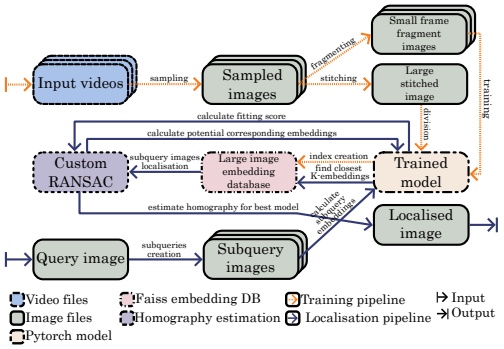


Figure 1: Model of training and localisation pipelines.

*xdubov02@stud.fit.vutbr.cz
†herout@fit.vut.cz
[1]query image – image to be localized
[2]sub-queries – patches of query image with same size as inputs of NN

phy model. Preliminary results indicate that this method is both feasible and efficient, demonstrating promising speed and accuracy. Section 2 provides a comprehensive review of the relevant literature. The used methodology for the creation of the dataset is explained in Section 3. The process of stitching to create a large photograph is clarified in Section 4. Section 5 describes the architecture of the trained neural network models. The construction of the large image embedding database is explicated in Section 6. The procedure for localizing the query image is detailed in Section 7, followed by a presentation and discussion of the results in Section 8. The paper concludes in Section 9, where a summary of the findings is provided along with potential avenues for future research.

2 Related Work

The main focus of this paper is on image localization, local feature descriptors, and image retrieval, all of which are discussed below. This paper aims to apply these three methods to the localisation of query photography within large photography.

2.1 Homography Estimation

Homography estimation is a technique used in computer vision and image processing to find the relationship between two images of the same scene, but captured from different viewpoints. We can divide this process into few cathegories mainly by number of sources:

- **Single-source homography estimation** – source images are usually acquired by the same device from different viewpoints or at different times.
- **Multi-source homography estimation** – source image data with two or more different types of imaging mechanisms for the same scene or object.

The focus of this work is to work with a single camera so that we can focus on single source homography estimation techniques. These techniques can be divided into two main groups: feature-based and deep-learning methods.

2.1.1 Feature-Based Methods

In the feature-based homography estimation method, the feature points in the image are first detected by a feature extraction algorithm and then the similarity metric for the matching is calculated. The parameters of the homography matrix are then solved using the mapping relationship of the matched feature points.

1. **Conventional** – Conventional homography estimation methods rely on hand-designed feature extractors. The process conventional homography estimation is divided into three main steps:
   (a) **feature detection** – In this step, distinctive features are identified in both images. These features could be corners, edges, or other notable structures in the image,
   (b) **feature matching** – Finds matches between these features. This involves comparing each feature in one image with all features in the other image and finding the best match. The result of this step is a set of corresponding feature points between the two images,
   (c) **homography matrix estimation** – Using corresponding feature points the homography matrix [3] is estimated.

   Common conventional descriptors include:

   - **SIFT**[13] – A descriptor that is invariant to image scale and rotation, and robust to changes in viewpoint, noise, and illumination. It detects and describes local features in images based on the histograms of the gradient orientations within a local region around the feature.
   - **BEBLID**[22] – an efficient binary descriptor. It represents a small part of an image using a binary string of zeros and ones. In various benchmarks, it has been shown to significantly enhance other binary descriptors, such as ORB or BRISK, while maintaining the same level of efficiency.

2. **Learning-Based** – These methods utilise neural networks to replace feature extraction or matching in traditional algorithms. Traditional methods are then used to estimate the homography transformation parameters at subsequent steps. Common learning-based descriptors include:

- **LIFT**[25] – A novel Deep Network architecture that implements the full feature point handling pipeline, that is, detection, orientation estimation, and feature description. This technique implements hard negative mining techniques over the entire image to obtain more accurate descriptors.
- **SuperGlue**[21] – Introduces a flexible context aggregation mechanism based on attention, enabling it to reason about the underlying 3D scene and feature assignments jointly. Matches two groups of local features by collectively finding correspondences while rejecting non-matchable points.
- **MatchFormer**[24] – Hierarchical extract-and-match transformer. Interleave self-attention to extract features and cross-attention to match features.

[3] 3x3 matrix that describes the transformation from one image to another

2.1.2 Deep Learning-Based Methods

Methods with a unified homography estimation pipeline, handled by a deep neural network model. Network understands and handles complex image correspondences. Common deep learning-based methods include:

- **RHWF** [4] [4] – Supervised method. Combines homography-guided image warping and the focus transformer. Image warping improves feature consistency. Focus Transformer uses the attention focusing mechanism to aggregate the intra-inter correspondence into global, non-local and local. Has a relatively small number of parameters. There is an increase in computational cost due to the use of homography-guided image warping and attentional manipulation.

- **MS2CA-HENet** [5] [10] – Unsupervised method. The method uses different input sizes at different stages to deal with different scales of homography transformations between images. Lower error can be achieved when there are large changes in displacement between corresponding points.

2.2 Image Retrieval

Image retrieval is the process of retrieving relevant images from a large database based on a query image or query terms. Image retrieval methods can be divided into 2 categories:

- **Content-Based Image Retrieval** (CBIR)

- **Text-Based Image Retrieval** (TBIR)

This paper focuses on content-based image retrieval (CBIR). This technique uses the visual content of a picture like colours, shapes, textures and spatial layout to represent and index the picture. In CBIR, the features of each image stored in the database are extracted and compared with the features of the queried image. It involves two steps:

1. **Feature extraction** – In this step, features such as colour histogram, texture, shape, etc. are extracted from the image.

2. **Similarity measurement** – After extracting the features, the similarity between the extracted features and the features of the query image is calculated.

The current focus of research is on deep learning methods. We can divide them into two main categories [18]:

- **Off-the-Shelf models** – Pre-trained deep learning models which are used as-is for image retrieval without further training or modification. However, they may not perform optimally for specific retrieval tasks due to domain shifts-differences between the data they were trained on and the new target data.

- **Fine-tuned models** – Pre-trained models that are further fine-tuned on a specific dataset related to the retrieval task. The model's weights need to be adjusted to better suit the particular characteristics of the new dataset, which will improve retrieval performance. However, this requires additional data and computational resources.

On the **off-the-shelf** side Mohedano et al. [15] proposed that both fully-connected layer and last convolutional layer can be used as feature extractors. **Fully-connected layer method** lack spatial information and a lack local geometric invariance. A. Razavian et al. [18] proposed very efficient single feed forward pass technique where features are used for direct similarity measurement without further processing. The need for more accurate image retrieval has led to a surge of multiple feed forward pass techniques. Although these techniques are more time-consuming, they can lead to more accurate results. Also discriminative features from the image patches better retain spatial information [18]. Multi-scale image patches could be obtained using sliding windows Y.Gong et al. [8] or spatial pyramid Y. Liu et al. [12]. These methods have problems with retrieval efficiency so Cao et al. [3] introduced merging image patches into larger regions with different hyper parameters. Random or dense creation of image patches may not be ideal so Zitnick et al. [29] proposed method where region proposals can be generated using object detectors instead. **The last convolutional layer method** preserves more structural details, which is particularly advantageous for instance-level retrieval [19]. The convolutional layer effectively organizes spatial information and generates location-specific features [28]. Razavian et al. [19] were the first to attempt spatial max pooling on the feature maps of an off-the-shelf DCNN model. They also apply max pooling on the convolutional features for retrieval to improve the discrimination of deep features. Yue et al. [17] and are the first to encode local features into VLAD [26] features. R. Arandjelović et al. [1] used VLAD as a layer plugged into the last convolutional layer. In addition to pooling agregation techniques, it is possible to embed the convolutional feature maps into a high-dimensional space to obtain compact features. Commonly used embedding methods include Bag of Words (BoW can be used with other metrics, such as Hamming distance [23]), Vector of Locally Aggregated Descriptors (VLAD [26]), and Fisher Vector (FV [7]).

[4] Recurrent Homography Estimation Using Homography-Guided Image Warping and Focus Transformer

[5] Multiscale Multi-stage based Content-Aware Homography Estimation method

3 Dataset creation

The study utilises a large collection of photographs depicting various indoor surfaces. The carpet dataset is the most extensive and commonly used dataset, and is considered the reference due to its 11 possible light conditions (data was collected at various times throughout the day and/or under artificial lighting). The dataset contains 21457 / 68961 / 142489 images based on the output size of the image used. It can be used to test independence for reflection symmetry in both directions of the axis. Other datasets include: laminate1, laminate2, carpet2, stone1, stone2, rusty_sheet_metal and wood. These are smaller (5000 – 10000 images). The dataset was created with several assumptions that made it easier to create, including:

- The camera scanning the ground is parallel to it,
- The camera scanning speed is constant,
- The objects to be captured should not move,
- The diversity of the dataset was mainly achieved by changing the lighting conditions.

The process of capturing the desired material or object involves using a custom-built cart that is designed to hold the camera parallel to the ground. This process must be repeated several times to capture the object in different lighting environments. By doing so, the dataset becomes more generalised and less impacted by lighting changes. The process of extracting fragments out of video frames consist of:

1. Sampling of videos by given sampling size,
2. Dividing large image into smaller fragments,
3. Localising fragments in sampled frames using conventional methods.

If a classification approach is used, the fragments will be grouped into classes based on their spatial position within the image. This means that there is a limiting factor to this approach, which determines how many surface datasets can be used for training (the number of classes should be finite and not very large). In the classification method, the name of the fragment file also means the class to which it belongs, and the two numbers are the x and y coordinates of the midpoint of that class. For triplet loss, each sample is created as a triplet. This triplet consists of an anchor, a positive image (from the same class) and a negative image (from a different class). Triplets are created by generating csv file from all image files that consists of paths to files and index of this file and class. This information is then used in semi-/hard negative mining. All fragments in datasets undergo custom augmentations, like shown in table 3.

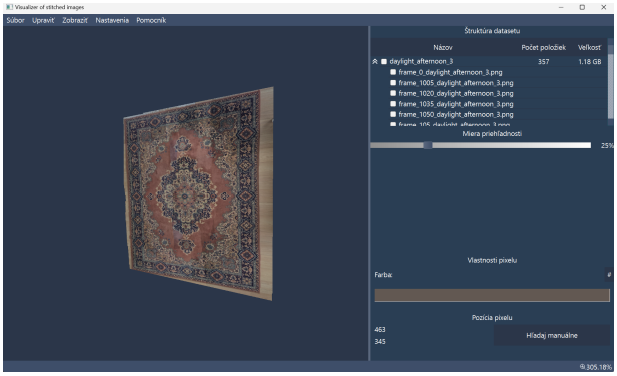


Figure 2: GUI application for visualising the homography of frames in a large image. This application is able to show the frames of the dataset in their correct positions.

Custom augmentations	
translation	$\pm 25px$
rotation	$\pm 180°$
homography	$\pm 25°$
center crop	224×224 pixels

4 Image stitching

To locate a query image in a large image, the first step is to retrieve the large image (map). The images for this map were created during the dataset creation process, explained in Section 3, specifically in the sampling section. This map is created using the image stitching technique, which involves the following steps:

1. **Keypoint Detection and Matching** – To detect the key points in each image, algorithm SIFT [13] is used. This is followed by the FLANN [16] key point matching algorithm, which allows the identification of correspondences between key points in different images.
2. **Homography Estimation** – RANSAC [7] algorithm is used to estimate the homography between pairs of images. Homography maps points in one image to corresponding points in another image.
3. **Image Warping and Blending** – Images are transformed to align them for stitching once homography has been estimated. Then, multiband image blending [27] is used to create one seamless large image.
4. **Extraction of Largest Inner Rectangle** – identifies and extracts the largest possible rectangle within the stitched image.

Numerous experiments have been conducted using various subdatasets to optimize image blending quality. Although most of the process has produced satisfactory results, a few minor artefacts require further investigation for refinement.

Figure 3: Examples of fragments from the same class.

5 Models architecture

The architecture of the models used in this study is based on the ResNet50 [9] model. The models are as follows:

- **Classification model** – The first model uses ResNet50 for classification tasks. The final layer is a fully connected layer with a softmax activation function, which outputs the probabilities for each class. In addition to the class probabilities, the model also returns the embeddings from the forward method. These embeddings are the output of the layer before the final fully connected layer. They represent high-dimensional learned features of the input data that the model uses for classification.

Model details for classification model	
Dimensions of embeddings	2048
Linear layers – carpet	**in**=2048, **out**=231
Linear layers – all	**in**=2048, **out**=914
Trainable parameters	24108389

- **Triplet model** – The second model also uses ResNet50, but it's trained with a triplet loss function. A triplet consists of an anchor, a positive, and a negative sample. The batch of data is used to extract the anchor, positive, and negative images. The anchor and positive images belong to the same class, while the negative image belongs to a different class. The model is used to obtain embeddings, which are vector representations of the images. Finally, the distances between the anchor and positive embeddings, and the anchor and negative embeddings, are calculated using a distance function. The model should select either hard or semi-hard negatives based on the chosen method. The valid triplets' embeddings are then selected for further processing. The calculation of the triplet loss involves the chosen embeddings. Finally, the loss is backpropagated and the model parameters are updated. The purpose of this design is to learn embeddings in such a way that the distance between an anchor image and a positive image (belonging to the same class) in the embedding space is smaller than the distance between the anchor image and a negative image (belonging to a different class).

Model details for triplet model	
Dimensions of embeddings	512
Linear layers – carpet / all	**in**=2048 **out**=512
Trainable parameters	24675111

6 Large image embedding database

A database of embeddings is created from the large image by traversing the entire image with a specified step in pixels and extracting the image to obtain its embedding. Currently, a step of 5 pixels in both axis directions is used. The created embeddings are then utilised to generate a Feiss index with the Feiss [6] library. In addition, the JSON file also stores the bounding box position of each index used in the matching process. Although this process is computationally and time-consuming, it is only performed once for a single large image.

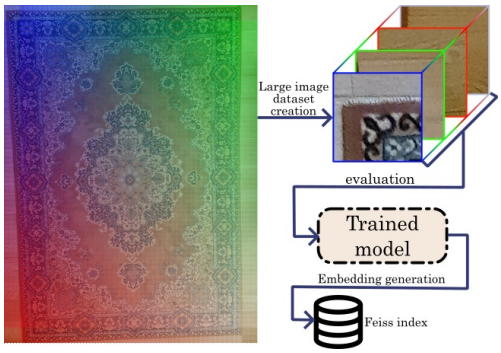


Figure 4: Feiss index creation process.

7 Localization

The final stage of the localisation pipeline is the localisation process. Its purpose is to identify a specific image within a larger one. The process begins by dividing the query image into smaller sub-queries, each of the same size as the inputs to our CNN model. If the image is large

Figure 5: Image showing all valid hypotheses for image patches, where small solid coloured squares are patches in the query image, dotted squares represent potential patch positions and the solid coloured squares in the large image represent the K-nearest embedding patch. The solid red rectangle represents the resulting homography without optimisations.

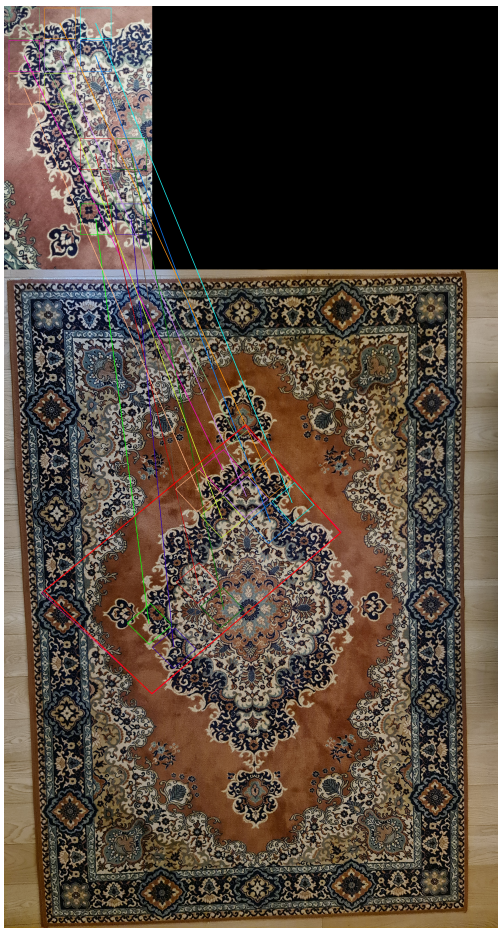


Figure 6: Result of the localisation. The red line is the original homography, while the pink line is the refined one. The solid squares are the sub-query inliers. The dotted ones are their localised potential correspondences.

enough to contain four non-overlapping sub-queries, the non-overlapping method is selected. This method divides the space between all the sub-queries equally. Otherwise, a method is selected where the sub-queries may overlap. It is important to ensure that the query image is smaller or at worst the same size as the large image to avoid computing homographies where the large image is inside a black area, resulting in missing information. For each sub-query, embeddings are extracted using one of the trained models. These embeddings capture the essential features of the sub-query. They are then used to search our database of embeddings from the large image. The user chooses K-nearest subquery embeddings to query the database. This allows us to find the most similar embeddings in a large index of FAISS large image embeddings for each sub-query. A random sampling algorithm with neighbourhood suppression is used to determine the best homography model based on the positions of four sub-query images and their corresponding positions in the large image. This generates a given number of homography hypotheses. Invalid homography hypotheses are marked to make the process-faster, including those with unwanted properties such as area change, angle change, and scale ratio change. The cosine distance is computed between the sub-query image embeddings and the potential homography hypothesis embeddings. The harmonic mean of all distances is then calculated to evaluate the hypotheses. This process is repeated for all valid hypotheses, and the homography hypothesis with the lowest global distance is selected as the best model. This homography represents the location of the query photo within the large image. The homography is refined by applying LK optical flow to the best hypothesis. Inliers are searched for within the potential sub-query centers. The potential midpoints and their corresponding closest real point are found using the second Feiss index with midpoints, this time with L2 distance. This process allows for the computation of a new refined midpoint position for each inlier sub-query, which can be used to compute a better homography. Homography refinement can

be performed multiple times to improve accuracy. If the new homography has a better distance, it will be chosen as the new one. Otherwise, the original homography will be selected. Despite the best possible training of the neural network, there are still inconsistencies that cannot be corrected even with a higher number of nearest neighbours. Further work will aim to improve the localisation model to avoid such inconsistencies.

8 Results

For clarity, the results are illustrated with figures and tables. All results are based on test data sets. Homographies were mainly tested with a these metric techniques: Average Corner Error [5] and Point Matching Error [14]. The main problem with them is that the points where the homographies are given have to be given manually.

Percentage accuracy of models					
classification model			**triplet loss model**		
train	**val**	**test**	**train**	**val**	**test**
99.21	97.78	97.71	98.72	95.35	94.11

Table 1: Accuracy of the trained models for each dataset.

query:(899x1599) map:(4800x3600)	This solution	SIFT + RANSAC
Create index database (once per new map)	9-21h (size dep.)	0s
Processing time (once for all of images)	7.9s	0s
Query time (for every query)	5,131s	0s
Localization (for every query)	5.471s	111.7s

Table 2: Average time in each part of localization pipeline.

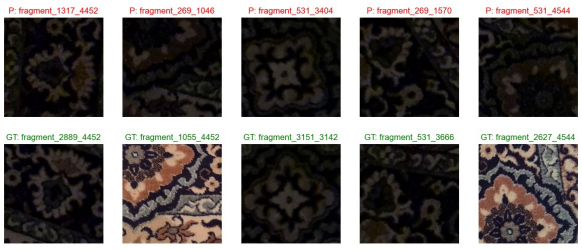


Figure 7: Examples of 10 misclassified images (CE). The two numbers represent the midpoint (x,y) of that particular fragment class.

9 Conclusions

In conclusion, this research presents a novel method for locating a specific photograph within a very large photograph. This method uses a convolutional neural network to extract embeddings from the query image, which are then used to perform an approximate search within a database of embeddings from the large photo. The results of this research demonstrate the effectiveness of this method, which is comparable to state-of-the-art localisation methods.

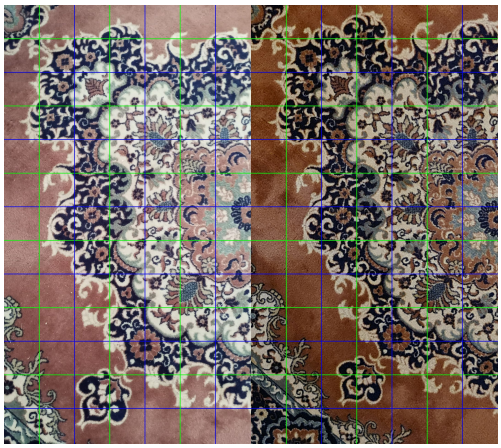


Figure 8: Query image next to the localised part of the large image after inverse homography transformation.

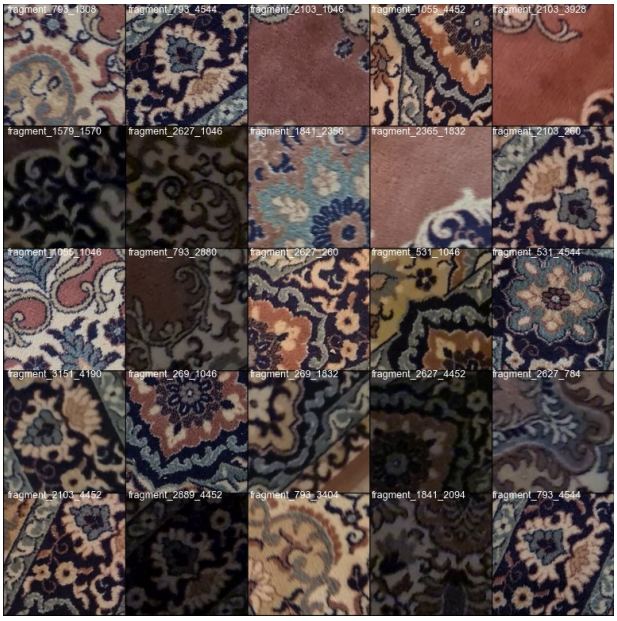


Figure 9: Examples of 25 images from testing dataset.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly super-

vised place recognition, 2016.

[2] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[3] J. Cao, L. Liu, P. Wang, Z. Huang, C. Shen, and H. T. Shen. Where to focus: Query adaptive matching for instance retrieval using convolutional feature maps. *CoRR*, abs/1606.06811, 2016.

[4] S.-Y. Cao, R. Zhang, L. Luo, B. Yu, Z. Sheng, J. Li, and H.-L. Shen. Recurrent homography estimation using homography-guided image warping and focus transformer. In *2023 IEEE/CVF Conf. on Comp. Vis. and Pattern Rec. (CVPR)*, pages 9833–9842, 2023.

[5] D. DeTone, T. Malisiewicz, and A. Rabinovich. Deep image homography estimation, 2016.

[6] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. The faiss library, 2024.

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981.

[8] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. *CoRR*, abs/1403.1840, 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[10] B. Hou, J. Ren, and W. Yan. Unsupervised multi-scale-stage content-aware homography estimation. *Electronics*, 12(9), 2023.

[11] P. V. Hough. Method and means for recognizing complex patterns, Dec. 18 1962. US Patent 3,069,654.

[12] Y. Liu, Y. Guo, S. Wu, and M. Lew. Deepindex for accurate and efficient image retrieval. In *ACM International Conference on Multimedia Retrieval (ICMR)*, pages 43–50, Shanghai, China, 06 2015.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, nov 2004.

[14] Y. Luo, X. Wang, Y. Wu, and C. Shu. Detail-aware deep homography estimation for infrared and visible image. *Electronics*, 11(24), 2022.

[15] E. Mohedano, A. Salvador, K. McGuinness, F. Marqués, N. E. O'Connor, and X. G. i Nieto. Bags of local convolutional features for scalable instance search. *Proceedings of the 2016 ACM on Int. Conf. on Multimedia Ret.*, 2016.

[16] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Int. Conf. on Comp. Vis. Theory and Applications*, 2009.

[17] J. Y.-H. Ng, F. Yang, and L. S. Davis. Exploiting local features from deep networks for image retrieval, 2015.

[18] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.

[19] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual instance retrieval with deep convolutional networks, 2016.

[20] P. Rousseeuw. Least median of squares regression. *Journal of The American Statistical Association - J AMER STATIST ASSN*, 79:871–880, 12 1984.

[21] P. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. *CoRR*, abs/1911.11763, 2019.

[22] I. Suárez, G. Sfeir, J. M. Buenaposada, and L. Baumela. Beblid: Boosted efficient binary local image descriptor. *Pattern Recognition Letters*, 133:366–372, 2020.

[23] F. Wang, W.-L. Zhao, C.-W. Ngo, and B. Merialdo. A hamming embedding kernel with informative bag-of-visual words for video semantic indexing. *ACM Trans. Multimedia Comput. Commun. Appl.*, 10(3), apr 2014.

[24] Q. Wang, J. Zhang, K. Yang, K. Peng, and R. Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching, 2022.

[25] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform, 2016.

[26] J. Zhang, Y. Cao, and Q. Wu. Vector of locally and adaptively aggregated descriptors for image feature representation. *Pattern Recognition*, 116:107952, 2021.

[27] Y. Zhao, W. Qian, and D. Xu. Fast multi-band blending using run-length encoding. In *2015 14th Int. Conf. on Computer-Aided Design and Comp. Graph. (CAD/Graphics)*, pages 224–225, 2015.

[28] L. Zheng, Y. Zhao, S. Wang, J. Wang, and Q. Tian. Good practice in cnn feature transfer, 2016.

[29] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 391–405, Cham, 2014. Springer International Publishing.

Rendering and beyond

Semantically Meaningful Vectorization of Line Art in Drawn Animation

Calvin Metzger*
Supervised by: Univ.Prof. Michael Wimmer †

Faculty of Informatics
TU Wien
Vienna / Austria

Abstract

Animation consists of sequentially showing multiple single frames with small mutual differences in order to achieve the visual effect of a moving scene. In limited animation, these frames are drawn as semantically meaningful vector images which could be referred to as clean animation frames. There are limited animation workflows in which these clean animation frames are only available in raster format, requiring laborious manual vectorization.

This work explores the extent to which line-art image vectorization methods can be used to automatize this process. For this purpose, a line-art image vectorization method is designed by taking into account the structural information about clean animation frames. Together with existing state-of-the-art line-art image vectorization methods, this method is evaluated on a dataset consisting of clean animation frames. The reproducible evaluation shows that the performance of the developed method is remarkably stable across different input image resolution sizes and binarized or non-binarized versions of input images, even outperforming state-of-the-art methods at input images of the default clean animation frame resolution. Furthermore, it is up to 4.5 times faster than the second-fastest deep learning-based method. However, ultimately the evaluation shows that neither the developed method nor existing state-of-the-art methods can produce vector images that achieve both visual similarity and sufficiently semantically correct vector structures.

Keywords: vectorization, line-art, animation, deep-learning

1 Introduction

In principle, animation consists of sequentially showing single frames in order to achieve the visual effect of a moving scene. *Limited animation* is an animation technique in which frames are not completely redrawn (like in full animation), but where the moving parts (also called *cels*) are reused over frames.

The hand-drawn limited-animation production process is composed of four phases. Based on the storyboard produced in the first phase, animators repeatedly draw and improve rough key frames in the second phase. These keyframes are line drawings only drawn for critical moments in a scene and contain mostly cels. In the third phase, the rough keyframes cleaned of any spurious lines or obsolete text markers and vectorized. To achieve the visual effect of fluidity, a large number of frames in between the keyframes are drawn. Finally, in the fourth phase, the clean frames are colored and enriched with special effects and a background image.

In order for the limited animation production process to proceed as quickly and as accurately as possible, clean frames need to be drawn as vector images. In the event of clean animation frames being only available in raster format, it is necessary to manually vectorize the images before they can be used efficiently. Automatizing this process is challenging, as the resulting line-art vector image needs to be semantically meaningful, i.e., the arrangement, topology and parameterization of graphical primitives (i.e., Bézier curves) need to make sense and be close to how artists would draw. An example of such a process is depicted in Figure 1

In order to alleviate this issue, this work will attempt to answer the **Research Question 1 (RQ1)**: To what extent is it possible to automatically vectorize clean animation frame line art in a manner that is semantically meaningful?

To answer RQ1, the Research Objective 1 (RO1) is to create a method for line-art vectorization that takes clean animation frame raster images as input and outputs the corresponding semantically meaningful vector image. This method is based on a deep learning model tailored to the qualitative structure of clean animation frames as input and output images, as traditional heuristics-based algorithms [15, 19, 11] tend to produce vector images that visually resemble the raster image closely, but contain semantically meaningless vector primitives.

Accordingly, the Research Objective 2 (RO2) is to perform an evaluation that ascertains the extent to which the developed method and existing state-of-the-art line-art image

*calvin.metzger@student.tuwien.ac.at
†wimmer@cg.tuwien.ac.at

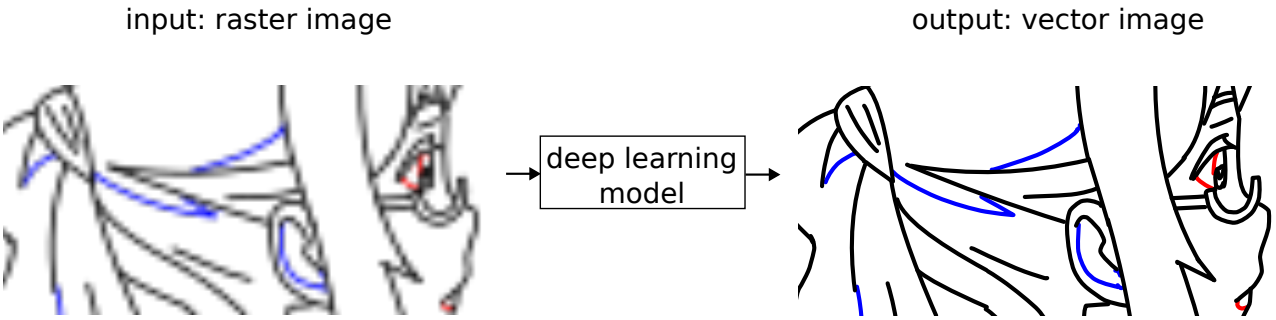


Figure 1: Overview of the research objective. The objective is to automatically convert clean animation frame line-art raster images into vector images. Zooming into the figure reveals the structural difference between the input and the output image. Input and output images are provided by Tonari Animation. Note that the output image is taken from the gold standard test dataset. For a genuine reconstruction result of the developed line-art vectorization method, refer to Figure 5.

vectorization methods are able to vectorize clean animation frames.

The code for this work is publicly available at `https://github.com/nopperl/marked-lineart-vectorization`.

2 Related Work

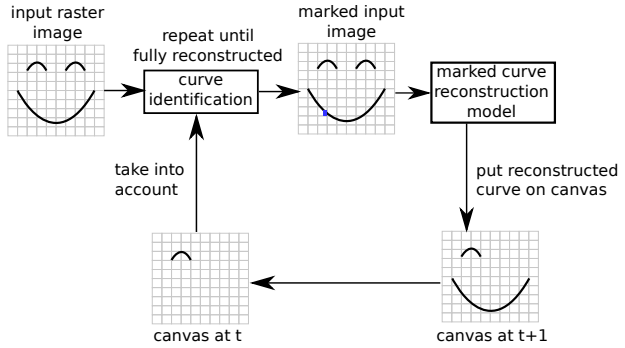
This section details existing work on image vectorization, specifically for the case of line art. Since there is a non-injective relation between vector images and raster images, converting a raster image into a vector image is a non-trivial task. Hence, state-of-the-art methods primarily utilize learned models to achieve this. While there exist methods based solely on heuristic optimization [15, 19, 11, 1, 21], they do not produce the intended output for this task, as the resulting vector primitives rarely resemble the primitives an artist would draw. Additionally, they require manual hyperparameter tuning for each individual image. Furthermore, each method relies on strong assumptions on the input image, such as exceeding a specific resolution, a low signal-to-noise ratio or containing only specific junctions.

While image vectorization is not yet a solved task, there have been some recent advances in deep learning for vector images. Reddy [13] introduce Im2Vec, an encoder-decoder architecture consisting of a Convolutional Neural Network (CNN) encoder and a Recurrent Neural Network (RNN) decoder. The CNN encodes the image into a latent feature vector, while the RNN is used to decode this feature vector into a fixed-length sequence of vector shapes based on multiple Bézier curves. It can be trained to vectorize raster images without vector supervision. This would be very useful in the context of line-art vectorization. The ability to train the model without vector supervision stems from its usage of a differentiable rasterizer [8]. In the general case, there are two main limitations of Im2Vec: The pixel resolution has to be defined at training time and the model does not scale well to higher resolutions. Additionally, the outputs sometimes contain degenerate features or semantically useless parts. Furthermore, Im2Vec only works on a specific type of image, such as emojis or icons. Finally, there were no experiments in the paper to output more than 4 shapes. Hence, it is doubtful whether it is possible to train the RNN decoder to output the large number of Bézier curves required for a clean animation frame.

The virtual sketching framework introduced by Mo et al. [10] is similar to Im2Vec in that it is trained without vector supervision to vectorize raster images. Other than that, it differs from Im2Vec in multiple ways. The main difference is that it constrains the output to only produce quadratic Bézier curves. Also, it is an iterative model, i.e. the curves are sequentially added to a canvas in a differentiable manner. After a given number of curves is drawn, the loss is computed and propagated through all the steps. These two differences make the model more suitable for professional line art. However, since the iterative model is trained mainly by computing a perceptual loss [6] of the whole output image with the input image, the results are not semantically meaningful vector images.

A different approach is to incorporate parts of traditional optimization-based methods. Based on earlier work [1], the state-of-the-art method by Puhachov et al. [12] uses a learned ensemble model to detect curve keypoints (such as junctions, start/end points and corners). Together with the input image, these keypoints are used by a geometric flow algorithm to find connections between keypoints and compute their geometry. It achieves remarkably good results, but has a more narrow aim than the proposed work. The algorithm focuses on retaining the correct stroke connectivity in the presence of noise, in their case for scanned pencil drawings. However, clean animation frames are not noisy and the curves are more narrow and densely connected, forming one large connected component for curves.

On the other hand, there do exist works that attempt to fully learn a line-art vectorization model using (partially) vector supervision, which makes it easier to produce semantically meaningful vector images [18, 4, 2]. Of note is a method to generate technical drawings by Egiazarian et al. [3]. It uses the Transformer [16, 14] architecture and is constricted to only handle 10 curves per image. To handle



(a) The method unrolled at time step $t+1$

Figure 2: Overview of the proposed method. The method iteratively reconstructs a given raster line-art image as a vector image. At time step $t=0$, an algorithm identifies a new curve to reconstruct and places a marker on it. This information is then passed to a learned marked-curve reconstruction model to reconstruct the curve in vector format using cubic Bézier curve parameters. This output is added to a canvas, which is taken into account when identifying the curve to reconstruct at $t+1$.

images with a larger amount of curves, each image is split into fixed-size tiles. The tiles are processed independently by using the Transformer model to predict vector primitives to match the curves in the image. The resulting primitives are then refined using a physics-inspired algorithm by aligning them to the black pixels in the raster image. Afterwards the primitives of all tiles are merged using a simple heuristic algorithm. While the model produces good results on technical line drawings, the authors also demonstrate that it generalizes to other line art. It is limited by the assumption that there are less than 10 curves within a tile and the reliance on the heuristic merging algorithm.

3 Method

This section describes a method to automatically convert line-art raster images into vector images. The method is visualized in Figure 2 and consists of two parts: the main part is a learned model that takes as input a raster line-art image and a mark on a curve in this image and outputs a cubic Bézier curve which fits the marked curve, which is described in Section 3.1. The second part is a lightweight algorithm that uses this model iteratively to reconstruct all curves in an image, which is described in Section 3.2.

The method is designed in an iterative manner in order to handle the large amount of Bézier curves in the considered line-art images. Additionally, this structure is more amenable to manual fixing of the output, since missing curves can easily be reconstructed by invoking the curve reconstruction part with a marker on the curve in question.

3.1 Marked-Curve Reconstruction Model

The marked-curve reconstruction model architecture is depicted in Figure 3 and was designed by following the principle that reducing the complexity of the task the model needs to solve increases the probability that the model actually converges to a suitable state.

This is achieved by three design decisions. The most important design decision is to have the model reconstruct only a *single* curve instead of all curves per invocation. The other two decisions are based on the input and the output of the model and are explained below.

3.1.1 Input and Output

The input of the model is a line-art raster image. Additionally, this image contains one marker pixel placed on a curve to reconstruct. Importantly, this means that the *location* of the curve is already established. This information can be used to reduce the task complexity for the model by centering the input image on the mark.

The raster input images are represented using the Red-Green-Blue (RGB) color model, i.e., each pixel is represented using three floating-point numbers in $[0,1]$.

The output of the model is defined as the parameters of a cubic Bézier curve with a fixed stroke width. The parameters are defined by the start point, the end point and two control points, resulting in a vector of length 8. This output structure is sufficient to represent the output data domain considered in this work, i.e., clean animation frames.

3.1.2 Model Architecture

The architecture of the marked-curve reconstruction model is depicted in Figure 3. It consists of an encoder neural network that turns the input image $\mathbf{x}$ into a latent vector $\mathbf{z}$ of length L, and a decoder neural network that turns this latent vector into cubic Bézier curve parameters $\mathbf{o}$.

Since the input is an image, the encoder is a convolutional neural network. A global average pooling layer [9] is used at the end to produce a latent vector of predefined length L independent of the input image size. The hyperparameters of the encoder layers are displayed in Table 1.

Note that, as described above, the encoder architecture is designed to handle variably sized input, with these variables being denoted in Table 1. The batch size B is used to process multiple observations in parallel and increase the effectiveness of batch normalization by decreasing the variance. The image width W and height H need to be a multiple of 2, but can be otherwise freely chosen. The latent vector length L needs to correspond to the length used for the input vector of the decoder. For this work, the hyperparameters are set to $B=32$, $W=H=512$ and $L=128$. $L=128$ was chosen after early experiments with smaller resolutions. W and H are set to a square multiple of 2 approaching the maximum clean animation frame resolution

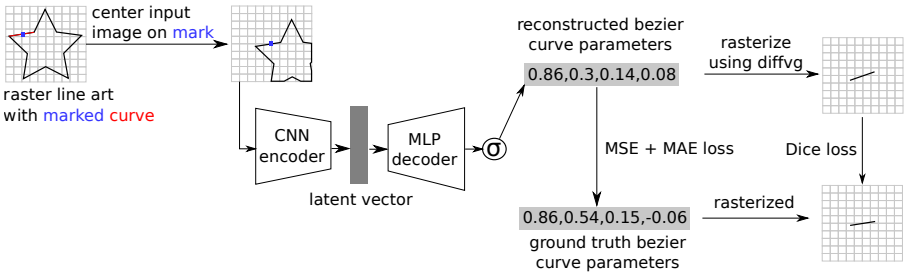


Figure 3: Architecture overview of the marked-curve reconstruction model. Note that for brevity, lines with two points are shown instead of cubic Bézier curves with four points.

of 1280×720 pixels. Note that the width and height deliberately do not correspond to the exact resolution of clean animation frames in the dataset to show that the model does not overfit to a specific resolution. B is maximized under the constraint of a limited amount of Graphics Processing Unit (GPU) memory.

The decoder is summarized in Table 2 and is a 2-layer multi-layered perceptron (MLP), which turns the latent vector of length L into a vector of length $P*2$, where P is the number of cubic Bézier curve parameters. Since cubic Bézier curves are parameterized by a start point, an end point and two control points, $P = 4$. Hence, the output is restricted to $[0,1]$ using the sigmoid function. The x-coordinates of the cubic Bézier curve points are then scaled with the image width, while the y-coordinates are scaled with the image height.

3.1.3 Training

The model is trained using supervised learning with a combination of a raster-based loss for visual similarity and a vector-based loss for semantic correctness. This task-derived loss combination is an important distinction from related work [13, 10, 3].

The vector loss follows Egiazarian et al. [3] and is an even combination of mean absolute error (MAE) and mean squared error (MSE). Defining a raster-based loss is more difficult, since the model outputs the cubic Bézier curve in vector format, which needs to be rasterized in a differentiable manner. The differentiable rasterizer introduced by Li et al. [8] is used for this. The raster output image is then compared to the rasterized ground truth image, with all curves aside from the marked curve removed. Using this, the dice loss function in Equation (1) can be used as loss. Note that o is the model output and y is the ground truth raster image.

$$\text{dice}(o,y) = 1 - \frac{2yo+1}{y+o+1} \tag{1}$$

The model is trained using the widely used Adam [7] optimizer with a learning rate of $\eta = 5*10^{-4}$.

3.2 Iterative Curve Reconstruction Algorithm

The marked-curve reconstruction model introduced in Section 3.1 is the main part of the line-art image vectorization method, but reconstructs only a single curve without color or stroke width information given a marked curve on the line-art raster image. In order to vectorize an entire line-art raster image, an algorithm has to be defined around the model that performs three tasks detailed in the following sections.

3.2.1 Color and Stroke Width

For the first task, recall that the marked-curve reconstruction model does not output color information. Since color carriers significant meaning in clean frames, it is necessary for the algorithm to produce the correct color information for all predicted curves.

This can easily be done for clean animation frames as they are drawn according to a color scheme which is known a priori. Hence, the image can be simply segmented according to these colors. For the dataset considered in this paper, these segments already exist. Then, the curve colors of each segment are set to black and each segment is individually input into the marked-curve reconstruction model. The color of its output can then be set to the segment color.

In the same vein, the marked reconstruction model does not output stroke width information. However, in clean animation frames, all curves share the same stroke width by design. Hence, it is possible to assume a constant stroke width for the input image and to apply it to all reconstructed curves.

3.2.2 Curve Identification

In order to indicate to the marked-curve reconstruction model which curve needs to be reconstructed, the second task consists of sampling a pixel lying on a curve not already reconstructed given the input image (more specifically an input image segment, as described in Section 3.2.1) and a canvas image containing already reconstructed curves. In the case of clean line-art images considered in this work, this can simply be done by sampling a random black pixel.

layer	output shape	# params	filter size	kernel size	stride	padding
2-d conv	$(B, 32, W/2, H/2)$	896	32	3	2	1
2-d conv	$(B, 64, W/4, H/4)$	18496	64	3	2	1
2-d conv	$(B, 128, W/8, H/8)$	73856	128	3	2	1
2-d conv	$(B, 256, W/16, H/16)$	295168	256	3	2	1
2-d conv	$(B, 512, W/32, H/32)$	1180160	512	3	2	1
2-d conv	$(B, L, W/32, H/32)$	589952	L	3	1	1
avg pool + squeeze	(B, L)	0	L	$W/32$	-	-

Table 1: Summary of the layers of the encoder neural network of the marked-curve reconstruction model.

layer	output shape	# params	size
linear	$(B, L/2)$	8256	$L/2$
batch norm	$(B, L/2)$	$2(L/2)$	
Rectified Linear Unit (ReLU)	$(B, 2P)$		
linear	$(B, 2P)$	520	$2P$
sigmoid	$(B, 2P)$	520	

Table 2: Summary of the layers of the encoder neural network of the marked-curve reconstruction model.

3.2.3 Marked-Curve Reconstruction Model Invocation

In order to vectorize the entire line-art image, the marked-curve reconstruction model has to be invoked iteratively until all curves are reconstructed. This is done in multiple steps, which are laid out in Algorithm 1.

Note, that Line 5 in Algorithm 1 constitutes an intuitive stopping criterion enabled by the progressive canvas image subtraction from the remaining image. Since missing a few curves is not a significant issue and errors in the model output are to be expected, the stopping criterion is set to $T = \lfloor B * 0.1 \rceil$, where B is the number of black pixels in the original image.

4 Dataset

The dataset used in this work consists of two parts: a human-generated dataset of 20,564 clean line-art images and a synthetic dataset. The human-generated dataset consists of 139 vector images provided by Tonari Animation, 425 vector images from the SketchBench benchmark, and 20,000 amateur sketches from the TU Berlin collection. The size of this dataset is increased using four data augmentation techniques: curve mirroring, curve rotation, curve reversion and curve dropout. The synthetic dataset is used to further increase the size of the training data. For that, images with a low number of randomly sampled cubic Bézier curves are generated and combined with the human-generated dataset at a 1:5 ratio. The entire dataset consists of Scalable Vector Graphics (SVG) vector images and corresponding rasterized Portable Network Graphics (PNG) images, with a uniform color for the background (white) and the curves (black).

5 Evaluation

To answer the RQ1, this section provides a quantitative and qualitative evaluation of the extent to which the line-art vectorization method developed in this work and comparable state-of-the-art methods are able to automatically vectorize clean animation frame line art. It is performed on a held-out portion of the dataset consisting of 10 Tonari animation frames.

5.1 Quantitative Evaluation

To perform the quantitative evaluation, the methods are applied to vectorize a test dataset consisting of evaluation dataset images and their results are compared using metrics which quantify the difference between the ground truth (i.e., the gold standard) and the vectorization results.

In detail, the vectorization methods are given a raster image $\mathbf{X}_{\text{raster}}$ as input and produce an output vector image $\hat{\mathbf{Y}}$, where $\hat{\mathbf{Y}} = (\hat{\mathbf{y}}_j)_{j=0}^{n}$ is a sequence of cubic Bézier curves of arbitrary length n and each cubic Bézier curve $\hat{\mathbf{y}} = (\hat{y}_i)_{i=1}^{8}$ is a sequence of 8 numbers, which represent the curve parameters (i.e., the start point, end point and two control points). The metrics measure how well $\hat{\mathbf{Y}}$ matches the ground truth vector image $\mathbf{Y}$ corresponding to the input image $\mathbf{X}_{\text{raster}}$, where again $\mathbf{Y} = (\mathbf{y}_j)_{j=0}^{m}$ is a sequence of cubic Bézier curves of length m.

Intersection-over-Union (IoU) Following related works [3, 10, 5], the visual similarity of the output image to the ground truth image is measured using the IoU defined in Equation (2). Note, that TP refers to the true positives, FP to the false positives and FN to the false negatives, which are calculated by binarizing the rasterized output image $\hat{\mathbf{Y}}_{\text{raster}}$ and input image $\mathbf{X}_{\text{raster}}$.

$$J = \frac{TP}{TP + FP + FN} \tag{2}$$

Curve error One method of measuring the correctness of the vector structure of the output $\hat{\mathbf{Y}}$ is to calculate its distance to the ground truth image $\mathbf{Y}$. This *curve error* is defined in Equation (3) as the sum of the distance of each curve in the output image to the corresponding ground truth curve. Hungarian ordering is used to establish curve

correspondence, i.e. the ground truth curve with the minimum distance is paired to each output curve. The sum of absolute errors defined in Equation (4) is used as distance function $d(\hat{\mathbf{y}}_i, \mathbf{y}_j)$.

$$\text{curve error}(\hat{\mathbf{Y}}, \mathbf{Y}) = \text{mean}_{i=0}^{n} \left(\min_{j=0}^{m} d(\hat{\mathbf{y}}_i, \mathbf{y}_j) \right) \quad (3)$$

$$d(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i=0}^{8} |\hat{y}_i - y_i| \quad (4)$$

Curve ratio Given an output image $\hat{\mathbf{Y}}$ that visually resembles the ground truth $\mathbf{Y}$, a simple measure of matching vector structures is to consider the ratio number of output curves and ground truth curves $n/m \in [0, n]$. In the case of perfectly matching vector structures, $n = m$ and $n/m = 1$.

Curve length Another method to measure how well the output vector structure matches the ground truth is to calculate the average curve length in pixels and compare it to the ground truth.

Curve distance Following Yan et al. [20], holes between curves are measured using the the minimum distance of each curve endpoints to each other curve endpoints. In detail, the metric is defined by Equation (5), where $\mathscr{E} = [0, 1, 6, 7]$ defines the indices of the start and the end point parameters of a curve. The closer the value is to the ground-truth baseline, the closer the vector structure can be considered to match the ground truth, while values that are higher than the baseline indicate more unintentional holes.

$$\text{mean}_{i=0}^{n} \left(\min_{j=0}^{n} \sum_{k \in \mathscr{E}} |\hat{y}_k^i - \hat{y}_k^j| \right) \quad (5)$$

Efficiency Furthermore, the runtime (in seconds) and GPU memory usage is measured to evaluate the efficiency of the algorithms.

5.1.1 Results

Table 4 shows the performance of the following line-art image vectorization methods: the method developed in this work (`marked`), the traditional algorithm by Weber [19] (`autotrace`), the vectorization algorithm combining deep learning and heuristic optimization by Puhachov et al. [12] (`polyvector-flow`), the deep learning-based algorithm using raster supervision by Mo et al. [10], (`virtual-sketching`), and the deep learning-based algorithm using vector supervision by Egiazarian et al. [3] (`deepvectechdraw`).

The methods are applied on the Tonari clean animation frame test dataset rasterized at a resolution of 512px, while preserving the aspect ratio. The Intersection-over-Union (IoU), curve error and runtime metrics can be easily interpreted: While the arrow in the column name indicates whether larger or smaller numbers represent better performance, the results of the best and the second-best performing method on the metric are indicated using bold and italics fonts, respectively.

For the remaining metrics, recall that the average curve length and the average curve distance should be close to the ground truth values, which are listed in Table 3. The curve ratio is calculated with the number of curves listed in the same table.

Table 4 shows that the line-art vectorization method developed in this work outputs vector images that resemble the input raster image the closest. It achieves this with the second-smallest curve error behind the method by Puhachov et al. [12] and with a curve distance that is close to the ground truth, just behind the method by Mo et al. [10]. Interestingly, it uses roughly half the curves of the ground truth, with curves on average being nearly twice as long. Finally, it is also the fastest deep learning-based method, while requiring the least amount of dedicated GPU memory.

Note that the traditional method by Weber [19] significantly outperforms all other methods on the runtime. On the other hand, it has the highest curve error and lowest IoU, suggesting ill-fitting outputs. The method by Puhachov et al. [12] also achieves a surprisingly low IoU, but also the best curve error.

The two deep learning-based methods by Mo et al. [10], Egiazarian et al. [3] approach the IoU of the method developed in this work, albeit with a significantly higher curve error and runtime. Additionally, the method by Egiazarian et al. [3] outputs the lowest amounts of curves, but the curves of the method by Mo et al. [10] are still longer on average, suggesting that this method produces more curves that do not fit the ground truth curves.

In general, most methods produce output images that surprisingly do not cover the input image well. This suggests that no method reproduces clean animation frames to the extent required by the task considered in this work.

5.1.2 Results with higher resolution input images

The methods by Weber [19], Puhachov et al. [12] performed unusually low on the evaluation in Table 4. A potential cause for this was identified as the low resolution of input images at 512px. To investigate this hypothesis, the evaluation was rerun with input images rasterized at twice the resolution, i.e., 1024px, while preserving the aspect ratio. Keep in mind that this is significantly higher than the standard resolution of clean animation frames considered in this work. Hence, performance increases of methods at this resolution will likely not materialize when they are applied to real-world clean animation frames, which usually will only be available at a lower resolution.

Figure 4 compares the evaluation results of the two resolutions sizes. Note that, since metrics measured in pixels

		curves median	curve length median	curve distance median
tonari	512-0.512	205.00	2.56	1555.82
	1024-1.024	205.00	5.12	1725.81
sketchbench	512-0.512	208.00	12.43	2355.22
	1024-1.024	208.00	24.85	2726.03

Table 3: Selected metrics of the vector images in the test dataset. This information can be used as baseline for the corresponding metrics in Table 4.

		autotrace	polyvector-flow	virtual-sketching	deepvec-techdraw	marked (ours)
IoU ↑	median	0.02	0.12	*0.29*	0.28	**0.30**
curve ratio	median	0.23	1.35	0.30	0.19	0.43
curve length	median	1.00	0.55	11.16	9.06	8.19
curve distance	median	891.00	439.18	1442.91	917.50	1361.28
curve error ↓	median	20.37	**14.05**	20.08	17.58	*16.76*
runtime ↓	median	**0.35**	14.82	22.99	97.73	*9.49*

Table 4: Comparison of the performance of the marked line-art image vectorization method and four prior works on the Tonari test subset at a resolution of 512px. If possible, the result of the best and the second-best performing method for the metric is indicated using bold and italics fonts, respectively.

scale linearly with the resolution size, they are normalized by the resolution size. It is clear that all prior methods except AutoTrace [19] perform significantly better than at 512px resolution. The method by Puhachov et al. [12] even reaches an IoU well over 0.5, i.e., its outputs cover more than half of the input image correctly on average. This is dampened by a high curve error and curve distance, indicating incorrect vector structures. The method by Egiazarian et al. [3] performs similarly well, with a lower IoU but better curve error and curve distance, seemingly striking a different balance between visual resemblance and semantically correct vector structures.

Interestingly, the metrics of the method developed in this work stay remarkably stable at the increased resolution. This is especially remarkable for the runtime, which significantly and predictably changes for all other methods.

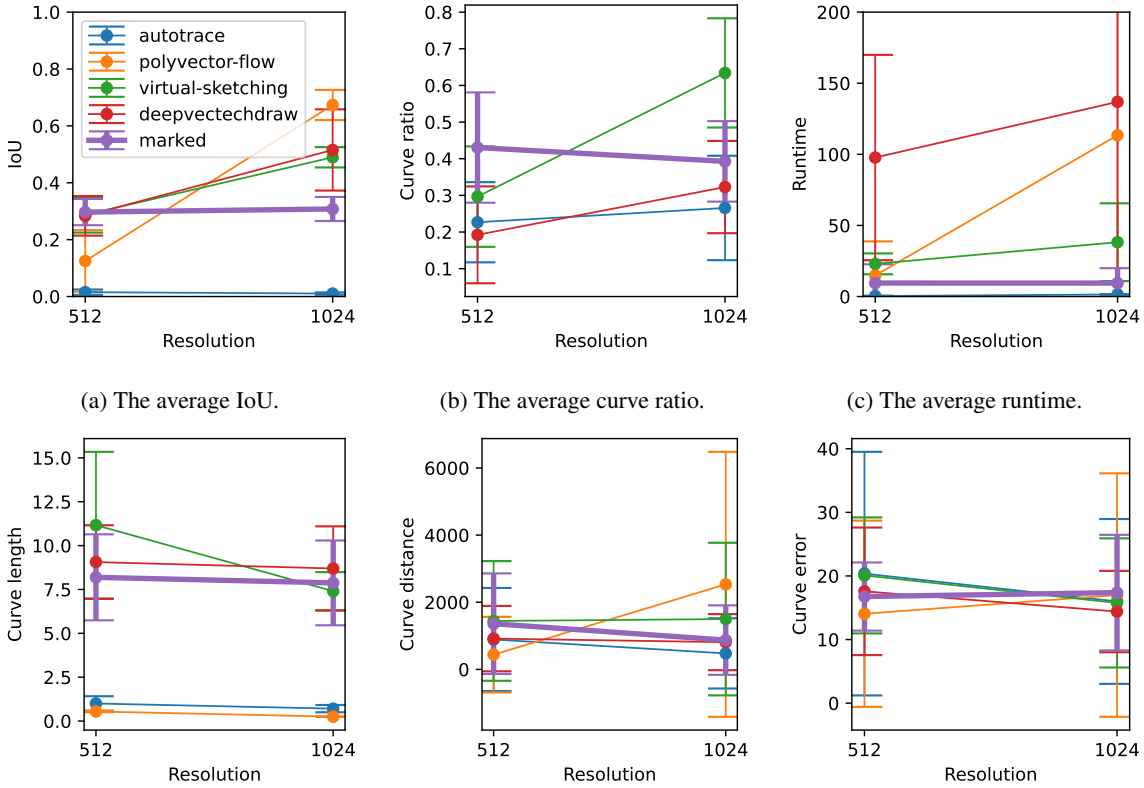
One potential reason for this remarkable input image resolution invariance of the method developed in this work is the selection of reconstruction curves using marks, which explicitly *forces* the model to reconstruct curves which other methods might not have detected. This can be the case for curves that are too thin or contain some spots at low resolutions.

5.2 Qualitative Evaluation

For a visual comparison, Figure 5 shows the best output of each method for an example clean animation frame by Tonari Animation. The input image has a resolution of 512px and is binarized for the methods by Weber [19], Puhachov et al. [12], Mo et al. [10], since that leads to higher-quality outputs. Since the main objective is to not only achieve visual similarity but also match the semantically correct vector structure of the ground truth vector image, Figure 5 attempts to visualize the underlying vector structure. Following Guo et al. [5], Mo et al. [10], Puhachov et al. [12], this visualization is achieved by representing curves using mutually exclusive colors. Furthermore, the images are zoomed in to lay bare minute differences. Indications for a correct vector structure are a constant color for continuous curves and a similarity to Figure 5a.

All methods appear to be visually correct at first glance, with varying quality and the methods by Egiazarian et al. [3], Mo et al. [10] not performing favourably. However, looking into details reveals significant deficiencies. The method developed in this work arguably produces the most closely matching vector structure, with most curves faithfully reconstructed following their appearance. On the other hand, curves are often slightly too short, leaving undesirable holes. Furthermore, there is a bias towards low curvature.

The method by Mo et al. [10] is similar to the method developed in this work in that it faithfully reconstructs curves, but fails to preserve the constant stroke width. The methods by Weber [19], Puhachov et al. [12] do not faithfully reconstruct curves, with multiple curves often merged into a single curve or altogether missing. This leads to a visually clean output – even without a significant amount of holes in the case of AutoTrace [19]. However, the produced vector structure is far from the ground truth in Figure 5a.



(a) The average IoU.

(b) The average curve ratio.

(c) The average runtime.

(d) The average curve length.

(e) The total curve distance.

(f) The average curve error.

Figure 4: Metrics for the line-art image vectorization methods evaluated on images with 512px and 1024px resolution, respectively. Points denote the median of the metric, while vertical bars denote the inter-quartile range (IQR). Horizontal lines show the trend of the metric. The metrics for the method developed in this work are emphasized. Note that they are not significantly affected by the image resolution and none decreases with lower resolutions.

6 Conclusion

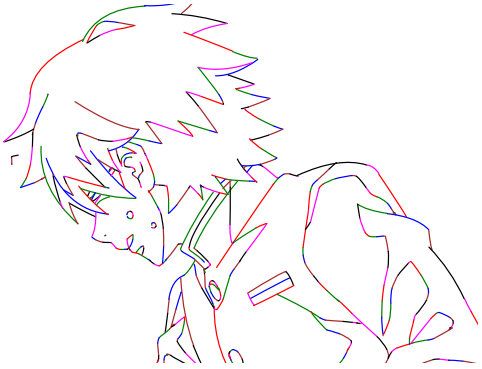
The objective of this work was to ascertain to what extent it is possible to automatically vectorize clean animation frame line art in a semantically meaningful way. In order to answer the RQ1, Section 3 proposed a clean animation frame line-art image vectorization method and Section 5 evaluated it together with prior work on an evaluation dataset provided by Tonari Animation. It could be shown that while the proposed method outperforms prior work at the default input image resolution, ultimately no line-art image vectorization method is able to satisfactorily vectorize clean animation frames, especially failing to properly reconstruct details and primitives with high curvature. Hence, no method studied in this work is of practical use in the limited-animation workflow. In order to achieve the goal of automatizing the tedious step of vectorizing clean animation frames, the curve reconstruction needs to be significantly more accurate.

Advantages of the developed method include remarkable robustness to input image resolution and binarization, resource efficiency and flexibility for manual fixing. Limitations include a significant amount of small holes in reconstructed curve sequences, limited semantic correctness and a bias towards lower curvature.

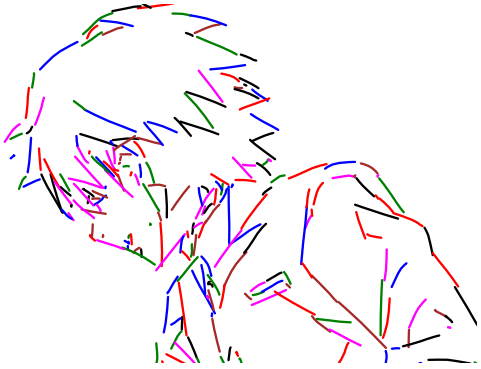
6.1 Future Work

There are numerous opportunities to improve on the presented work. The dataset could be improved by collecting a larger amount of high-quality data or performing more advanced data augmentation or feature extraction. A further promising improvement is to finetune a large vision-language model such as CogVLM [17] instead of training a small CNN based encoder-decoder model from scratch in order to utilize their emergent capabilities.

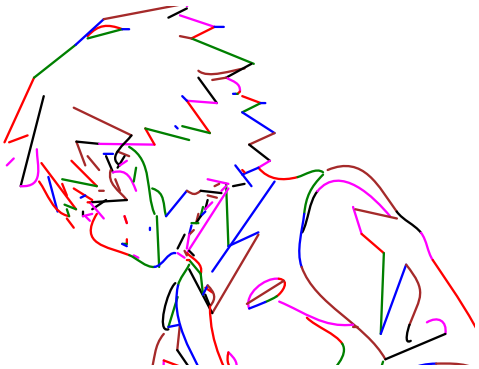
Furthermore, there exist other tasks to which the developed model could be extended. These include the generation of inbetween frames based on keyframes or clean animation frame colorization. Moreover, the model output could be constrained to exhibit temporal consistency, i.e., to consist of curves that remain consistent across frames of the same scene.



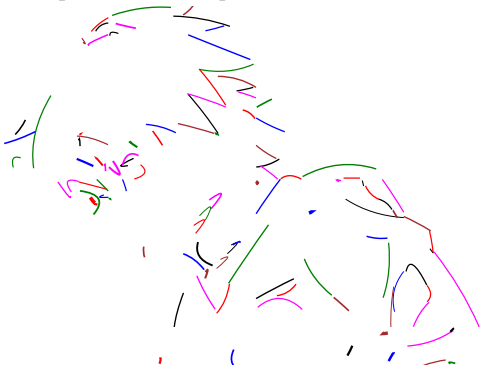
(a) Ground truth.



(b) Output of the developed method.

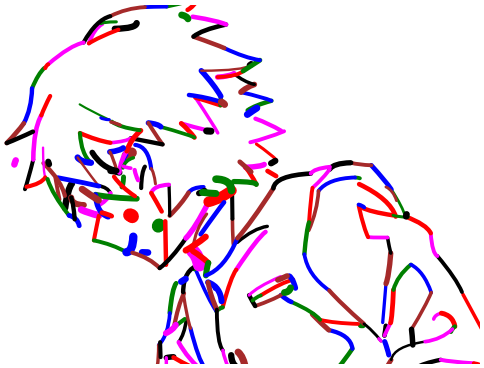


(c) Output of AutoTrace [19].



(d) Output of Egiazarian et al. [3].

(e) Output of Puhachov et al. [12].



(f) Output of Mo et al. [10].

Figure 5: The output vector image given a Tonari clean animation frame in raster format as input of each line-art image vectorization method studied in this work. The vector structure behind the images is revealed by representing each curve with a mutually exclusive color and a high zoom level.

References

[1] Mikhail Bessmeltsev and Justin Solomon. Vectorization of line drawings via polyvector fields. *ACM Trans. Graph.*, 38(1):9:1–9:12, 2019. doi: 10.1145/3202661. URL https://doi.org/10.1145/3202661.

[2] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, and Yi-Zhe Song. Vectorization and rasterization: Self-supervised learning for sketch and handwriting. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5672–5681. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00562. URL https://openaccess.thecvf.com/content/CVPR2021/html/Bhunia_Vectorization_and_Rasterization_Self-Supervised_Learning_for_Sketch_and_Handwriting_CVPR_2021_paper.html.

[3] Vage Egiazarian, Oleg Voynov, Alexey Artemov, Denis Volkhonskiy, Aleksandr Safin, Maria Taktasheva, Denis Zorin, and Evgeny Burnaev. Deep vectorization of technical drawings. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIII*, volume 12358 of *Lecture Notes in Computer Science*, pages 582–598. Springer, 2020. doi: 10.1007/978-3-030-58601-0_35. URL https://doi.org/10.1007/978-3-030-58601-0_35.

[4] Jun Gao, Chengcheng Tang, Vignesh Ganapathi-Subramanian, Jiahui Huang, Hao Su, and Leonidas J. Guibas. Deepspline: Data-driven reconstruction of parametric curves and surfaces. *CoRR*, abs/1901.03781, 2019. URL http://arxiv.org/abs/1901.03781.

[5] Yi Guo, Zhuming Zhang, Chu Han, Wenbo Hu, Chengze Li, and Tien-Tsin Wong. Deep line drawing vectorization via line subdivision and topology reconstruction. *Comput. Graph. Forum*, 38(7):81–90, 2019. doi: 10.1111/cgf.13818. URL https://doi.org/10.1111/cgf.13818.

[6] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, volume 9906 of *Lecture Notes in Computer Science*, pages 694–711. Springer, 2016. doi: 10.1007/978-3-319-46475-6_43. URL https://doi.org/10.1007/978-3-319-46475-6_43.

[7] 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*, 2015. URL http://arxiv.org/abs/1412.6980.

[8] Tzu-Mao Li, Michal Lukác, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph.*, 39(6):193:1–193:15, 2020. doi: 10.1145/3414685.3417871. URL https://doi.org/10.1145/3414685.3417871.

[9] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.4400.

[10] Haoran Mo, Edgar Simo-Serra, Chengying Gao, Changqing Zou, and Ruomei Wang. General virtual sketching framework for vector line art. *ACM Trans. Graph.*, 40(4):51:1–51:14, 2021. doi: 10.1145/3450626.3459833. URL https://doi.org/10.1145/3450626.3459833.

[11] Gioacchino Noris, Alexander Hornung, Robert W. Sumner, Maryann Simmons, and Markus H. Gross. Topology-driven vectorization of clean line drawings. *ACM Trans. Graph.*, 32(1):4:1–4:11, 2013. doi: 10.1145/2421636.2421640. URL https://doi.org/10.1145/2421636.2421640.

[12] Ivan Puhachov, William Neveu, Edward Chien, and Mikhail Bessmeltsev. Keypoint-driven line drawing vectorization via polyvector flow. *ACM Trans. Graph.*, 40(6):266:1–266:17, 2021. doi: 10.1145/3478513.3480529. URL https://doi.org/10.1145/3478513.3480529.

[13] Pradyumna Reddy. Im2vec: Synthesizing vector graphics without vector supervision. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*, pages 2124–2133. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPRW53098.2021.00241. URL https://openaccess.thecvf.com/content/CVPR2021W/SketchDL/html/Reddy_Im2Vec_Synthesizing_Vector_Graphics_Without_Vector_Supervision_CVPRW_2021_paper.html.

[14] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992. doi: 10.1162/NECO.1992.4.1.131. URL https://doi.org/10.1162/neco.1992.4.1.131.

[15] Peter Selinger. Potrace: a polygon-based tracing algorithm, September 2003. URL https://www.mathstat.dal.ca/~selinger/potrace/potrace.pdf.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[17] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. Cogvlm: Visual expert for pretrained language models. *CoRR*, abs/2311.03079, 2023. doi: 10.48550/ARXIV.2311.03079. URL https://doi.org/10.48550/arXiv.2311.03079.

[18] Yizhi Wang and Zhouhui Lian. Deepvecfont: synthesizing high-quality vector fonts via dual-modality learning. *ACM Trans. Graph.*, 40(6):265:1–265:15, 2021. doi: 10.1145/3478513.3480488. URL https://doi.org/10.1145/3478513.3480488.

[19] Martin Weber. AutoTrace, September 2002. URL https://autotrace.sourceforge.net/. accessed on 2022-12-03.

[20] Chuan Yan, David Vanderhaeghe, and Yotam Gingold. A benchmark for rough sketch cleanup. *ACM Transactions on Graphics (TOG)*, 39(6), November 2020. ISSN 0730-0301. doi: 10.1145/3414685.3417784. URL https://doi.org/10.1145/3414685.3417784.

[21] Zibo Zhang, Xueting Liu, Chengze Li, Huisi Wu, and Zhenkun Wen. Vectorizing line drawings of arbitrary thickness via boundary-based topology reconstruction. *Computer Graphics Forum*, 41(2):433–445, 2022. doi: https://doi.org/10.1111/cgf.14485. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14485.

A Appendix

This section contains additional information related to the implementation of the vectorization algorithm.

Algorithm 1: Iterative Curve Reconstruction.

Input: A raster line-art image.
Output: A vector line-art image.

```
1  Segment input image by color;
2  foreach image segment do
3      canvas = an empty vector image of the same size
         as the input image;
4      remaining = image segment;
5      while number of black pixels in remaining > T;
         do
6          Compute marker by applying curve
             identification on the remaining image;
7          Centered image = center the remaining
             image on the marker;
8          reconstructed curve = invoke the
             marked-curve reconstruction model using
             the centered image;
9          Inverse the center location of the curve by
             using the mark location;
10         Add the reconstructed curve to the canvas
             image;
11         remaining = remaining - rasterized canvas
             image;
12     end
13     Set color of all curves in the canvas image to the
         segment color;
14 end
15 Merge the canvas images;
16 return Merged canvas images
```

Time Evolution Simulation of the Quantum Mechanical Wave Function in 3D Space

Zoltán Simon*
Supervised by: Dr. Balázs Csébfalvi

Department of Control Engineering and Information Technology
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics

Abstract

In quantum mechanics, the wave function describes the state of a physical system. In the non-relativistic case, the time evolution of the wave function is described by the time-dependent Schrödinger equation. In 1982, D Kosloff and R Kosloff proposed a method to solve the time-dependent Schrödinger equation efficiently using Fourier transformation. The computational physics research group, led by Géza I. Márk in the Nanotechnology Department, Institute for Technical Physics and Materials Science, Centre for Energy Research, located in Budapest, in collaboration with Belgian researchers, developed a simulation method based on three-dimensional wave packet dynamics for the study of electron dynamics in nanosystems. A simplified, interactive, two-dimensional version for educational purposes was published in 2020. In this work, we demonstrate two improvements of the wave packet dynamical simulation software: (i) the use of the Graphical Processing Unit (GPU), which results in a vast (up to 50x) increase in simulation speed, and (ii) the introduction of advanced visualization techniques which are helpful to correctly interpret massive 4D space-time wave function data sets obtained from the simulation.

Keywords: Quantum Mechanics, Wave Packet Dynamics, Ray Tracing, Simulation

1 Introduction

In the first quarter of the 20th century Quantum Mechanics (QM) opened a whole new window to understand our universe. Tamás Geszti, in his book [18] writes: *"learning QM is part of the process of understanding the world, and the person who masters it, understands the world better"*. QM can be used efficiently to model the behavior of atomic particles. It describes how electrons behave in the orbitals around the atomic core and explains chemical reactions. It can be used to model the structure of molecules. In nanotechnology, it is crucial to make quantum mechanical calculations to predict –and, in many cases, explain– the behavior of different nanostructures. One exciting field of study is the science of single-layer materials [23]. These are also known as 2D materials. One such carbon structure is called graphene [5, 16, 14]. This single-layered structure conducts heat and electricity very efficiently, thus raising high hopes in many when it comes to possible use-cases. Inspired by the previously enumerated fields of application, we set the goal to study the behavior of quantum systems by computer simulation. Such simulations are beneficial for scientists. They use such methods to accurately model the interaction between particles and various potential fields. In order to accomplish this goal, we choose a method that uses the Fast Fourier Transform (FFT) to efficiently calculate the time development of the quantum mechanical wave function. In QM, the wave function describes the state of a physical system. In the non-relativistic case, the time evolution of the wave function is described by the time-dependent Schrödinger equation [15]. In 1982, D Kosloff and R Kosloff proposed a method [8] to solve the time-dependent Schrödinger equation efficiently using Fourier transformation. In 2020, Géza István Márk published a paper [11] describing a computer program for the interactive solution of the time-dependent and stationary two-dimensional (2D) Schrödinger equation. Some details of quantum phenomena are only observable by calculating with all three spatial dimensions. Géza István Márk and his colleagues have already used 3D calculations in their research work [19, 9]. The difference is that their implementation uses solely the Central Processing Unit (CPU) of a computer. For visualization of the resulting probability density so far, they used the isosurface method. Our contribution mainly lies in leveraging the parallelization potential of the modern Graphical Processing Unit (GPU), thus significantly boosting the calculation speed by approximately a factor of 50 on our test hardware. We also apply state-of-the-art volumetric visualization techniques to create pleasing and comprehensible visuals to analyze the probability density evolution in 3D space.

*zoltan.simon@edu.bme.hu

2 Theoretical background

By examining atomic particles, scientists have observed that such particles exhibit wave-like behavior, and in bounded systems, they can absorb or release energy only in discrete quanta. These matter waves have complex amplitudes, and can interfere with themselves.

Equation 1 is called the Schrödinger equation. It is a linear partial differential equation, and it is the governing equation of QM published by Erwin Schrödinger [15] in 1926. Linearity is a requirement for matter waves since, by the definition of superposition, a general equation that aims to describe the behavior of matter waves must be satisfied not only by simple waves but also by the linear combination of these waves.

$$\frac{d}{dt}\Psi(\vec{r},t) = -\frac{i}{\hbar}\hat{H}\Psi(\vec{r},t) \tag{1}$$

In equation 1, we can see that on the left side, we basically take the first derivative of the wave function with respect to the time and on the right side we let the $\hat{H} = -\frac{\hbar^2}{2m}\Delta + V(\vec{r})$ Hamiltonian operator [6] affect the wave function. $\hbar$ is the reduced Planck constant. By specifying an initial state and solving this differential equation, we can predict the time development of a quantum mechanical wave function.

The wave function is a complex valued function. Experiments show that the square of the absolute value of the complex amplitude is the probability density associated with the particle being found in a given infinitesimally small portion of space at a given time. For convenience and to be sound with probability theory, we normalize the amplitude of the wave function so that the probability of the particle being found "somewhere" in space equals $\mathbb{P} = 1$.

$$\int_{\mathscr{V}} |\Psi(\vec{r},t)|^2 \, d^3r = 1 \tag{2}$$

3 Calculation method

Back in 1982, D Kosloff and R Kosloff proposed a method [8] to solve the time-dependent Schrödinger equation efficiently using Fourier transformation. The advantage of this algorithm compared to the Finite Difference in Time Domain (FDTD) methods[22, 10] is the high numerical stability of the time evolution step. In the adopted FFT method, no signs of divergence are present even after a large number of simulation steps. The time development step of the algorithm has a time complexity of $\mathscr{O}(N \log N)$ since it only uses six FFT runs ($\mathscr{O}(N \log N)$ each) and three element-wise multiplication between tensors ($\mathscr{O}(N)$ each). The amount of FFT runs and multiplications can be reduced further if we do not want to read the results of the time development in each step. A significant speed-up can be reached by using a parallelized implementation of the FFT algorithm as we did by using an efficient GPU implementation. In the following part, we would like to explain the FFT method in detail. The formal solution of equation 1 can be written in the form of equation 3.

$$\Psi(\vec{r},t) = e^{-\frac{i}{\hbar}\hat{H}(t-t_0)}\Psi(\vec{r},t_0) \tag{3}$$

where $\Psi(\vec{r},t_0)$ is a specified initial state and $\Psi(\vec{r},t)$ is the state after some $\delta t = t - t_0$ time. The problematic part is the Hamiltonian operator in the exponent. The kinetic and potential operators can not be commuted in general. Hence, the exponential can not be factored. We can decompose the exponential by the symmetrical unitary product [4, 3] as shown in the form 4.

$$e^{-\frac{i}{\hbar}\hat{H}\delta t} = e^{-\frac{i}{\hbar}(\hat{K}+\hat{V})\delta t} \approx e^{-\frac{i}{\hbar}\hat{K}\delta t/2}\, e^{-\frac{i}{\hbar}\hat{V}\delta t}\, e^{-\frac{i}{\hbar}\hat{K}\delta t/2} \tag{4}$$

The error of this approximation is $\mathscr{O}(\delta t^3)$; therefore, we have to be careful with selecting a small enough time resolution. When the potential energy is localized, the $\hat{V}$ operator is a simple multiplication by $V(\vec{r})$ function; thus the middle part of the product can be calculated in the form of equation 5.

$$e^{-\frac{i}{\hbar}\hat{V}\delta t}\Psi = e^{-\frac{i}{\hbar}V(\vec{r})\delta t}\Psi \tag{5}$$

The $\hat{K}$ kinetic operator involves calculating the spatial derivative of the wave function. We can use the Fourier transform, to make the conversion between real space and momentum space. Relation 6 holds for the derivative of an arbitrary f function and its Fourier transform.

$$ik\mathscr{F}\{f\} = \mathscr{F}\{f'\} \tag{6}$$

Taking the derivative in real space means multiplication by ik imaginary wave number in momentum space. We work with the $\Delta = \nabla \cdot \nabla$ Laplace operator, so we have to multiply by $(ik)^2 = -k^2$. By exploiting the linearity of the Fourier transform, we arrive at formula 7 for the kinetic energy part of the Hamiltonian function.

$$\hat{K}\Psi = \frac{p^2}{2m}\Psi = -\frac{\hbar^2}{2m}\Delta\Psi = -\frac{\hbar^2}{2m}\mathscr{F}^{-1}\{-k^2\mathscr{F}\{\Psi\}\} \tag{7}$$

where $\mathscr{F}^{-1}$ is the inverse Fourier transform. In momentum space, the k wave number is trivially given as it can be thought of as the very coordinate the function is parameterized with.

Actually, in equation 4, the $\hat{K}$ kinetic energy operator is in the exponent multiplied by $-\frac{i}{\hbar}\delta t/2$. Using the knowledge gathered from equation 7, we can now write equation 8.

$$e^{-\frac{i}{\hbar}\hat{K}\delta t/2}\Psi = \mathscr{F}^{-1}\left[e^{-\frac{ik^2\hbar\delta t}{4m}}\mathscr{F}\left[\Psi\right]\right] \tag{8}$$

Having a discrete data set, Discrete Fourier transform (DFT) can be efficiently implemented using the Fast Fourier Transform (FFT) algorithm. The output of the simulation is the wave function. The probability density can be obtained by calculating the square of the absolute value of the wave function for each grid cell. Making use of formulas 4, 5, and 8 and plugging them into the formal solution of the Schrödinger equation we can create algorithm 1 for the time development of the wave function.

Algorithm 1 Time advance algorithm

$\Psi \leftarrow$ initial state of the wave function
$V \leftarrow$ localized potential
$\delta t \leftarrow$ time resolution
$N_t \leftarrow$ number of time steps
$P_V \leftarrow e^{-\frac{i}{\hbar}V(\vec{r})\delta t}$
$P_K \leftarrow e^{-\frac{ik^2\hbar\delta t}{4m}}$
for $i \in [0, N_t)$ **do**
  $\Psi^{(1)} \leftarrow FFT^{-1}[P_K\, FFT[\Psi]]$
  $\Psi^{(2)} \leftarrow P_V\, \Psi^{(1)}$
  $\Psi \leftarrow FFT^{-1}\left[P_K\, FFT\left[\Psi^{(2)}\right]\right]$
  Visualize $|\Psi|^2$
end for

3.1 Defining Gaussian wave packets

In the algorithm, first, we have to specify an initial state of the wave function. Erwin Schrödinger introduced the concept of the Wave Packet (WP). A WP is a wavefront that propagates and reflects as a classical particle would do and also exhibits all the wave-like behavior described by QM. It bridges the gap between classical and quantum physics. The term Wave Packet Dynamics (WPD) refers to the process of modeling QM systems by initializing WPs and observing the propagation, reflection, scattering, and interference of the WP. In our work, we use Gaussian WPs. In this case the probability density of the WP has Gaussian distribution [21], hence the name. The definition of such wave function can be written in the form of equation 9.

$$\psi(\vec{r}) = \left[\frac{2}{\pi a^2}\right]^{\frac{D}{4}} exp\left[i\vec{k_0}\cdot\vec{r}\right] exp\left[-\frac{|\vec{r}-\vec{r_0}|^2}{a^2}\right] \quad (9)$$

where $\vec{r_0}$ is the initial position (with the highest probability density), $\vec{k_0}$ is the initial wave vector, and D is the dimension, which is $D := 3$ in our simulation. We can obtain the width of the Gaussian WP as $\Delta r = \frac{a}{2}$.

If we do not want to visualize the probability density in each iteration, we can further optimize the calculation by merging the first step of the nth iteration and the last step of the $(n-1)$th iteration. If we omit the visualization step, we can do one forward FFT then perform a multiplication between the moment space wave tensor and the P_K^2 kinetic propagator calculated for a whole δt interval, instead of the one used in Algorithm 1 calculated for $\delta t/2$ interval.

4 Our implementation

Using the Fourier method, we created a Python application simulating the time development of the quantum mechanical wave function. We use ray tracing to visualize the resulting volumetric probability density. The visualization requires the sampling of a 3D data set on a discretized grid. This makes it impossible to fully reconstruct the wave function that we simulated using only a finite resolution to begin with. In order to fight sampling artifacts, we deploy a state-of-the-art triquadratic reconstruction filter recently proposed by Balázs Csébfalvi [2].

4.1 GPU parallelization and Just-In-Time compilation

The Fourier method described in section 3 opens up the possibility to implement the simulation on the GPU. Using GPU acceleration is one of our contributions to the already existing implementation used at the Nanotechnology Departement, Institute for Technical Physics and Materials Science, Centre for Energy Research. The Compute Unified Device Architecture (CUDA) toolkit is often used for parallel computational tasks implemented on the GPU [12]. It comes with a powerful GPU based FFT implementation. To use CUDA with Python, we selected the CuPy wrapping library [13] that provides abstraction over CUDA. We have used Numba to access Just-In-Time compilation (JIT) features. JIT means that for some parts of the otherwise interpreted source code, the compiler performs a runtime translation to native code. This feature is especially useful when iterating over large arrays.

4.2 Performance test

We measured the performance of our application. We used a personal laptop to run and test the program. The system specification of our computer is summarized in figure 1. First, we tried a CPU-only version of our simulator to compare the results with the GPU accelerated implementation. The results of the comparison can be found in figure 2. Here, we tested three different configurations with varying resolutions. We measured the average iteration count per second. The test shows that by using GPU acceleration, we obtained significant speed up over the CPU-only implementation.

CPU	AMD Ryzen 5 6600H 3.30 GHz
GPU	NVIDIA GeForce RTX 3050 Ti Laptop GPU
RAM	16 GB
OS	MS Windows 11 64-bit

Figure 1: System specification of the used test hardware

Input size	CPU only [iter/s]	GPU accel. [iter/s]
128^3	1.1	11.5
256^3	0.09	6.5
512^3	0.01	0.5

Figure 2: Results of a performance test using a CPU-only and the GPU accelerated version

5 Results

5.1 General approach and the system of units

We used our software to perform various WPD simulations. In this section, we present the results of some of these simulations. Our simulator uses Hartree atomic units [7]. Every quantity in the upcoming part should be interpreted as such. This unit system makes it convenient to deal with quantities at atomic scale.

5.2 Double-slit experiment

First, we simulated electron scattering experiments. Scattering of a particle happens when the WP of the particle passes through some kind of a barrier with holes in it. In our simulation, we can model the barrier as a localized potential. The WP arrives from one side of the barrier. While passing through this barrier, it scatters, and some of the WP gets reflected. The portion of the WP that passed through –suffering scattering– continues forward and consequently arrives to a measuring device[1]. In our simulation, our measuring device is a virtual canvas where we measure the probability density. A simple scattering scenario is the double-slit experiment. Here the barrier is a potential wall with two narrow parallel slits. The WP passes through these slits.

We performed the simulation using a distance between the barrier and the measuring canvas of $L = 30$ Bohr radii, a distance between the two slits of $d = 4.0$ Bohr radii, and a WP wavelength of $\lambda = \frac{2\pi}{3} \simeq 2.1$ Bohr radii. The width of each slit was a small enough value of 1.0 Bohr radii. Snapshots of the double-slit simulation can be seen in figure 3, where we used ray tracing to visualize the probability density and the potential. The corresponding interference pattern is visualized in figure 4 on a canvas of size 60×60 Bohr radius.

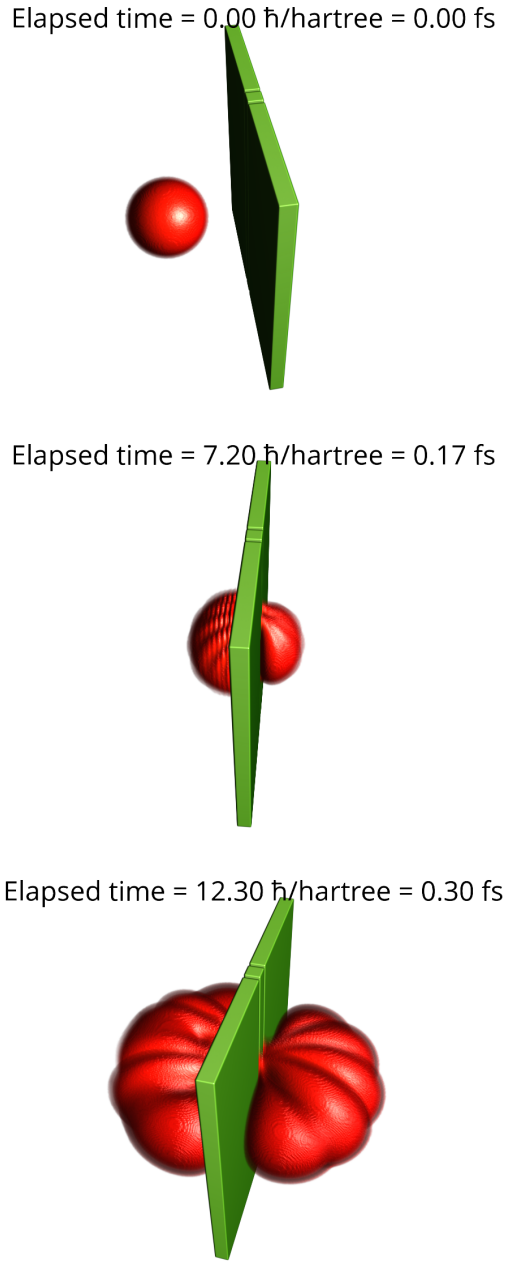


Figure 3: Double-slit experiment: the wave packet passes through the slits in the potential barrier

5.3 Diffraction by optical grating-like potential

Many different forms of diffraction can be explained using QM. The scale at which diffraction happens ranges from the scale of subatomic particles to larger molecules. Measuring diffraction patterns is a useful tool in the hands of scientists. It provides information about the object that caused the diffraction. The previously presented double-slit experiment is a 2D phenomenon because the localized potential is independent of the z coordinate. In the third dimension, there is free propagation. To make use of all three simulated dimensions, we also modeled diffraction on diffraction gratings. In optics, a diffraction grating is a periodic 2D structure that diffracts light [17]. In QM, similar gratings can also be utilized to diffract wave packets. The holes between the potential nodes behave like the holes in the double-slit experiment. We put 11 nodes in each direction, forming a rectangular grid. Each node has a Gaussian potential distribution and a maximal potential of $V_{max} = 8$ Hartree. The distance between adjacent grid points is $d = 4$ Bohr radii. The canvas distance is $L = 30$ Bohr radii, and the wavelength of the WP is $\lambda \simeq 2.1$ Bohr radii. Note that the kinetic energy of the WP $E = \frac{p^2}{2m} = \frac{h^2}{2\lambda^2} \simeq 4.5$ Hartree is less than V_{max}. Oth-

[1] In scattering and diffraction experiments we can make the distinction between a near field and far field solution.

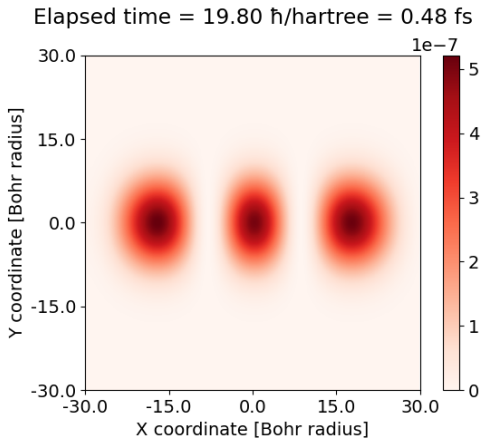


Figure 4: Simulated interference pattern of double-slit experiment

erwise, the grating would not impact the propagation of the WP sufficiently. In figure 5, we visualize subsequent stages of the scattering.

During the simulation, many interesting interference patterns arise. We show some of these for the 4 Bohr radii lattice constant case in figure 6, and for the 8 Bohr radii lattice constant case in figure 7.

5.4 Many-body interactions

One interesting use case of a higher-dimensional WPD simulator is that the higher-dimensional space can be used to model the interactions between multiple lower-dimensional particles. For example, our 3D simulation is capable of the simulation of three 1D particles. To do this, we have to define a special interaction potential. To create such potential, we have to think about the coordinates in the higher-dimensional configuration space as the coordinates of multiple lower-dimensional particles. If the potential energy affecting all particles can be expressed as a $V(x_a, x_b, x_c)$ function of the location of particle A and B and C, then we can reinterpret this function as the $V(\vec{r})$ localized potential function used in the potential propagator in equation 5. Note that here $\vec{r}$ becomes (x_a, x_b, x_c). To model the interaction between the three 1D particles, we initialized a hard interaction potential that takes its maximum inside an ε radius around each particle otherwise it is constant zero. To prevent blotting of the Gaussian WP we also added a harmonic oscillator potential. This helps because the Gaussian WP is the eigenstate of the harmonic oscillator. The potential for a harmonic oscillator is given in the form of equation 10.

$$V(x) = \frac{m\omega^2}{2}x^2 \tag{10}$$

Here m is the oscillating mass and ω is the angular frequency of the oscillation. We created a scenario where particle A starts at 25 Bohr radii away from the center of

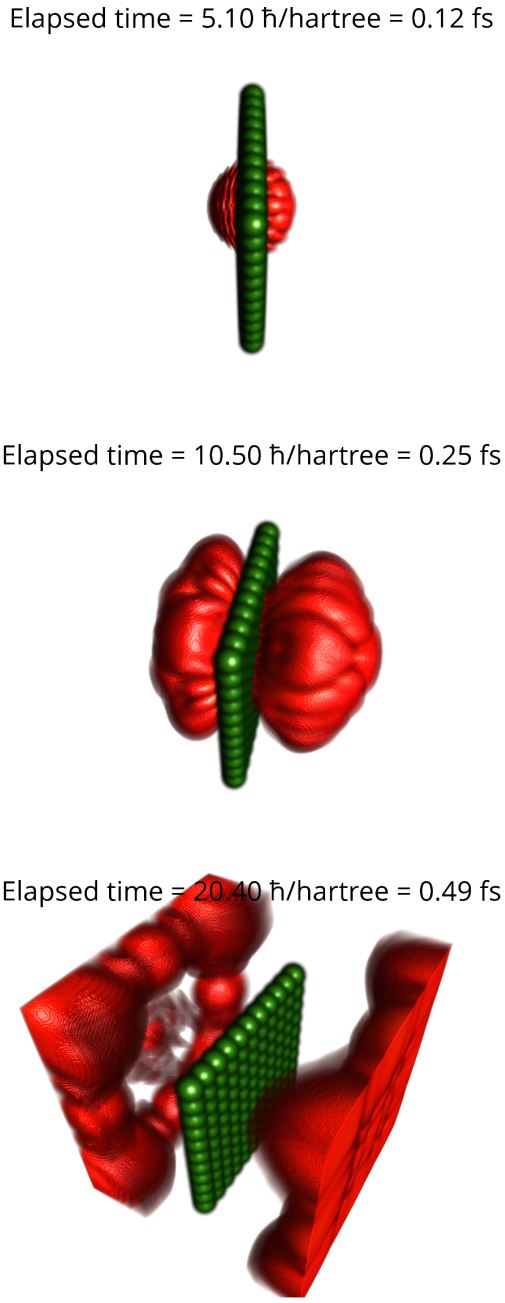


Figure 5: Diffraction grating experiment: the grating has a lattice constants of 4 Bohr radii

the oscillator where the potential energy is maximal; thus, it accelerates towards the other two particles (B and C), consequently transferring the momentum to particle C on the far right. The angular frequency of the oscillator was selected to be $\omega = \frac{2\pi}{40} \simeq 0.1571 \frac{\text{rad}\cdot\text{Hartree}}{\hbar}$.

We placed a finite potential barrier in the middle of the oscillator. We chose the thickness of this barrier so that approximately half of the wave packet of particle B tunnels through the barrier, giving its momentum to particle C on the next side of the wall. This causes C to start moving

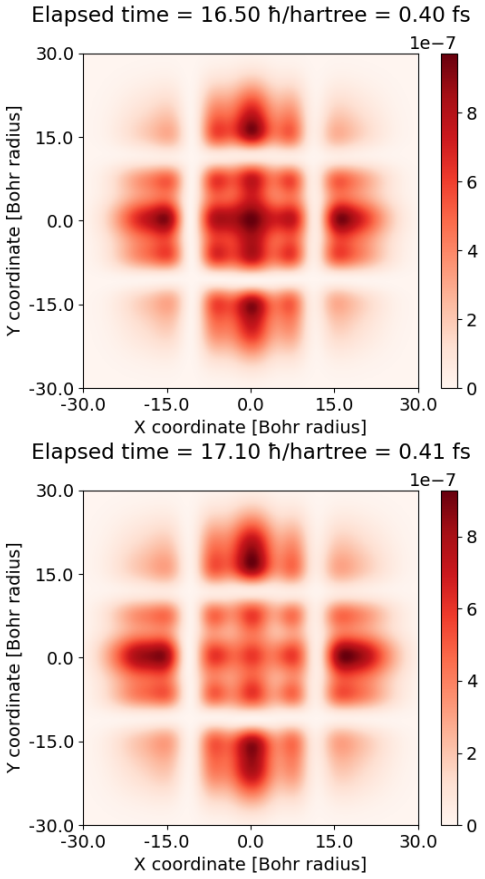


Figure 6: Interference pattern forming on the measurement canvas during diffraction grating simulation using lattice constant of 4 Bohr radii

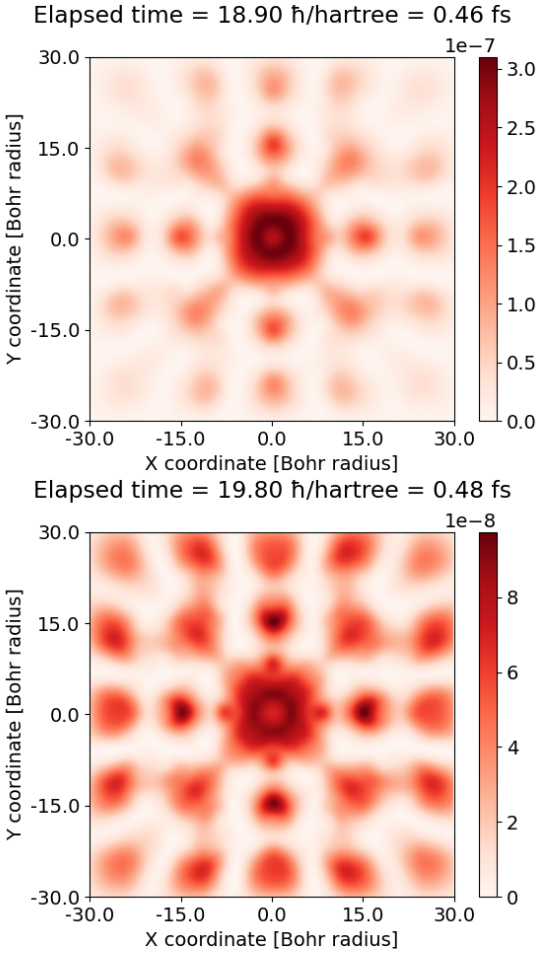


Figure 7: Interference pattern forming on the measurement canvas during diffraction grating simulation using lattice constant of 8 Bohr radii

with a probability of approximately $\frac{1}{2}$. What we just described is called the entanglement of the states of particles A, B, and C. Let's perform a measurement to determine the location of particles A, B, and C right after the previously described sequence of interactions! If we would measure particle A to be located in the middle of the harmonic oscillator, that means that it gave its momentum to particle B and B has tunneled through the finite potential barrier. If B tunneled, that also means that beyond the barrier, it collided with particle C, consequently transferring all of its kinetic energy to C. On the contrary, if the result of the measurement determining the location of particle A would have shown that particle A bounced back from B, that means that B did not tunnel through the barrier. This also means that particle C did not receive any kinetic energy and stayed stationary right beyond the barrier. The measurement of the state of one entangled particle determines the outcome of the measurement of the other entangled particles. Real-life experiments are sound with this thought experiment [20]. The probability density plot can be observed in figure 8.

6 Discussion

In our work, we wrote about simulating the time development of the quantum mechanical wave function in 3D space. Our accomplishments are the following

- We adopted a simulation method that uses the Fourier transform as a subroutine to efficiently calculate the solution of the time-dependent Schrödinger equation.
- As an improvement over Géza István Márk's implementation, we ported the Fast Fourier Transform to the Graphical Programming Unit, thus reaching a major speed-up of a factor of 50 for some cases.
- We combined state-of-the-art volume visualization techniques to enhance the visual quality of the resulting probability density images.
- We used our simulator software to run various Wave Packet Dynamical simulations ranging from basic diffraction scenarios up to simulation of lower-dimensional particles in configuration space.

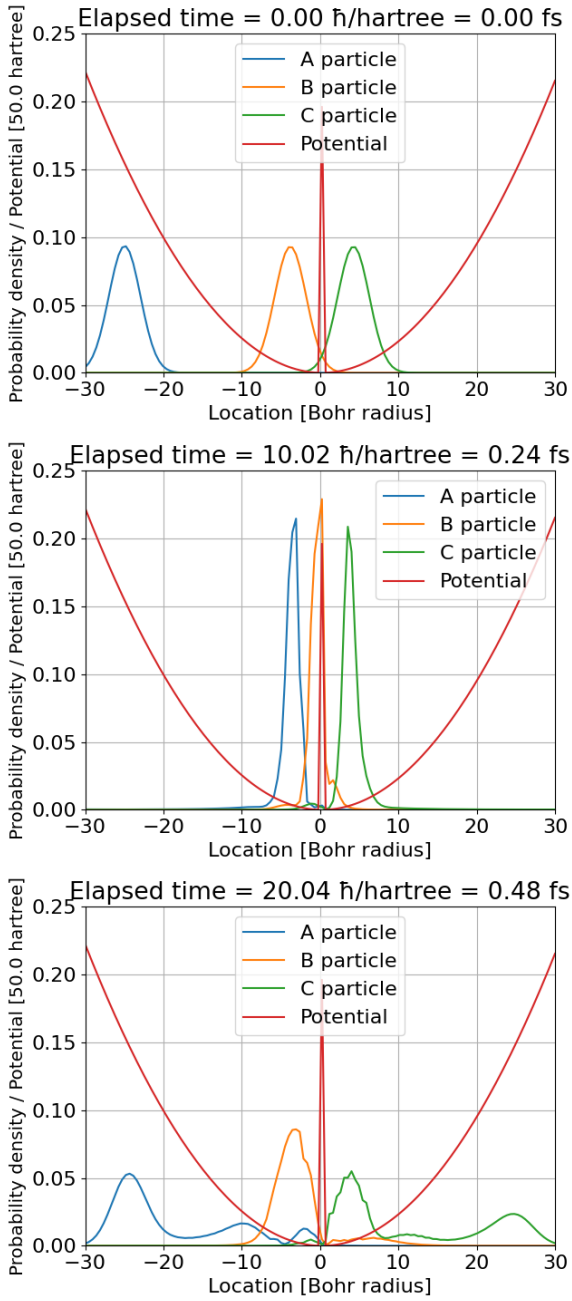


Figure 8: Stages of interactions between 1D particles in harmonic oscillator with finite potential barrier: initial state; particle A colliding with particle B; particle C reaching maximal potential on the right side of the oscillator with approximately 0.5 probability while it's in superposition with the state of staying stationary next to the central barrier

We see our work as a successful entry into the quantum mechanical wave packet dynamics world and a good starting point for further research. In the future, we want to make it possible to calculate the eigenstates of the localized potential. This would require the calculation of the Fourier transform in the time domain to obtain the energy state of the system. Then, iteratively converge towards the eigenstate. There is also a possibility of incorporating electromagnetism into the Hamiltonian operator. As we have simulated 1D particles in configuration space, we could use a higher-dimensional space to model the interaction between multiple multidimensional particles. One interesting path to go down on is to build a machine learning solution that is able to initialize a localized potential field that guides the wave function into a desired state. This particular idea is inspired by the marvelous work of Barnabás Börcsök, who presented his paper about controlling 2D laplacian eigenfluids [1] at the Central European Seminar on Computer Graphics in 2023. From a visualization point of view, there are also many possibilities to improve. There is room for even better reconstruction filters. We are very hopeful about the future research potential of this topic and are very eager to continue the fruitful work.

7 Acknowledgements

I am immensely grateful for the support of my supervisor, Dr. Balázs Csébfalvi, and for all the valuable insights and advice of the researchers at the Centre for Energy Research: Dr. Géza István Márk and Dr. Péter Vancsó. This work has been supported by the OTKA K-145970 project.

References

[1] Barnabás Börcsök. Controlling 2d laplacian eigenfluids. *Central European Seminar on Computer Graphics*, 2023.

[2] Balázs Csébfalvi. One step further beyond trilinear interpolation and central differences: Triquadratic reconstruction and its analytic derivatives at the cost of one additional texture fetch. *Computer Graphics Forum*, 42(2):191–200, 2023.

[3] M.D Feit, J.A Fleck, and A Steiger. Solution of the schrödinger equation by a spectral method. *Journal of Computational Physics*, 47(3):412–433, 1982.

[4] J. A. Fleck, J. R. Morris, and M. D. Feit. Time-dependent propagation of high energy laser beams through the atmosphere. *Applied physics*, 10(2):129–160, Jun 1976.

[5] A. K. Geim. Graphene: Status and prospects. *Science*, 324(5934):1530–1534, 2009.

[6] William Rowan Hamilton. *On a general method of expressing the paths of light, & of the planets, by the coefficients of a characteristic function*. Printed by P.D. Hardy Dublin, Dublin, 1833.

[7] D. R. Hartree. The wave mechanics of an atom with a non-coulomb central field. part i. theory and methods. *Mathematical Proceedings of the Cambridge Philosophical Society*, 24(1):89–110, 1928.

[8] D Kosloff and R Kosloff. A fourier method solution for the time dependent schrödinger equation as a tool in molecular dynamics. *Journal of Computational Physics*, 52(1):35–53, 1983.

[9] Geza Mark, Péter Vancsó, Laszlo Biro, Dmitry Kvashnin, Leonid Chernozatonskii, Andrey Chaves, Khamdam Rakhimov, and Philippe Lambin. *Wave Packet Dynamical Calculations for Carbon Nanostructures*, pages 89–102. 01 2016.

[10] Frederick Ira Moxley, Tim Byrnes, Fumitaka Fujiwara, and Weizhong Dai. A generalized finite-difference time-domain quantum method for the n-body interacting hamiltonian. *Computer Physics Communications*, 183(11):2434–2440, 2012.

[11] Géza I. Márk. Web-schrödinger: Program for the interactive solution of the time dependent and stationary two dimensional (2d) schrödinger equation, 2020.

[12] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. In *ACM SIGGRAPH 2008 Classes*, SIGGRAPH '08, New York, NY, USA, 2008. Association for Computing Machinery.

[13] Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. Cupy: A numpy-compatible library for nvidia gpu calculations. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

[14] C. N. R. Rao, Kanishka Biswas, K. S. Subrahmanyam, and A. Govindaraj. Graphene, the new nanocarbon. *J. Mater. Chem.*, 19:2457–2469, 2009.

[15] E. Schrödinger. An undulatory theory of the mechanics of atoms and molecules. *Phys. Rev.*, 28:1049–1070, Dec 1926.

[16] Meryl D. Stoller, Sungjin Park, Yanwu Zhu, Jinho An, and Rodney S. Ruoff. Graphene-based ultracapacitors. *Nano Letters*, 8(10):3498–3502, 2008. PMID: 18788793.

[17] George W. Stroke. *Diffraction Gratings*, pages 426–754. Springer Berlin Heidelberg, Berlin, Heidelberg, 1967.

[18] G. Tamás. *Kvantummechanika*. Elméleti fizika. Typotex, 2007.

[19] Péter Vancsó, Géza I. Márk, Philippe Lambin, Alexandre Mayer, Yong-Sung Kim, Chanyong Hwang, and László P. Biró. Electronic transport through ordered and disordered graphene grain boundaries. *Carbon*, 64:101–110, 2013.

[20] D. Vasilyev, F. O. Schumann, F. Giebels, H. Gollisch, J. Kirschner, and R. Feder. Spin-entanglement between two freely propagating electrons: Experiment and theory. *Phys. Rev. B*, 95:115134, Mar 2017.

[21] Xinhua Zhang. *Gaussian Distribution*, pages 425–428. Springer US, Boston, MA, 2010.

[22] Min Zhu and Yi Wang. Rk-ho-fdtd scheme for solving time-dependent schrodinger equation. *The Applied Computational Electromagnetics Society Journal (ACES)*, 36(08):968–972, Oct. 2021.

[23] Houlong L. Zhuang and Richard G. Hennig. Computational discovery, characterization, and design of single-layer materials. *JOM*, 66(3):366–374, Mar 2014.

Utilizing Measured Reflectance for Real-time Rendering in Game Engines

Lukáš Cezner*
Supervised by: Vlastimil Havran†

Department of Computer Graphics and Interaction
Czech Technical University in Prague
Prague / Czech Republic

Abstract

Having an appropriate reflectance model is a crucial part of achieving realistic rendering. Currently, the vast majority of renderers rely on physically-based analytic models with a few parameters. In this paper, we consider another approach and explore the possibility of using measured reflectance data to render 3D objects covered with real-world materials in real-time graphics. Specifically, we created an implementation for two major game engines, Unity and Unreal Engine 5, and compared it with the analytic model. For these purposes, more than 200 samples of materials were measured, and an application was developed to process the measured data.

Keywords: Bidirectional reflectance distribution function, real-time rendering, game engine, direct lighting

1 Introduction

One of the main tasks in computer graphics is to compute realistic images. This task can be achieved using the rendering equation, which incorporates a bidirectional reflectance distribution function (BRDF, see Section 1.2). This function is often described as an analytic model, a polynomial with certain parameters.

In the following sections, we present a different method for expressing a BRDF, which involves an interpolation of values obtained from real-world materials. We describe a workflow that consists of measuring the reflectance of a material, processing the measured data, and rendering the surface with these BRDF values.

1.1 Spherical coordinate system

The spherical coordinate system represents a vector $\vec{v} = (x, y, z)$ using two angles θ, ϕ and a radial distance $r = ||\vec{v}||$. The angle θ is characterized as the angle between the vector $\vec{v}$ and the basis vector z, while ϕ denotes the angle between the basis vector x and the projection of the vector $\vec{v}$ on the xy plane. The conversion between a Cartesian and a spherical coordinate system can be described as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \cdot \sin\theta \cdot \cos\phi \\ r \cdot \sin\theta \cdot \sin\phi \\ r \cdot \cos\theta \end{bmatrix}, \quad \begin{bmatrix} \theta \\ \phi \\ r \end{bmatrix} = \begin{bmatrix} \arccos\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right) \\ \arctan\left(\frac{x}{y}\right) \\ \sqrt{x^2+y^2+z^2} \end{bmatrix}. \tag{1}$$

On many occasions, we used the spherical coordinate system to describe the direction $\vec{\omega}$ (a unit vector, $r = 1$). An illustration of this situation is shown in Figure 1.

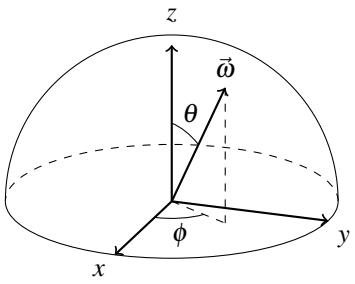


Figure 1: A direction vector $\vec{\omega}$ in spherical and Cartesian coordinate system.

1.2 Bidirectional Reflectance Distribution Function

The bidirectional reflectance distribution function (BRDF) is a mathematical representation of how light is reflected from the surface of an opaque material. For a specific wavelength of light, this is a function of two direction vectors $\vec{\omega}_{in}, \vec{\omega}_{out}$ in spherical coordinates ($\vec{\omega}_{in} = (\theta_{in}, \phi_{in}), \vec{\omega}_{out} = (\theta_{out}, \phi_{out})$) that represent the incoming (light) direction $\vec{l}$ and the outgoing (view) direction $\vec{v}$. The value of the BRDF is determined by the ratio of the reflected radiance L_{out} in direction $\vec{\omega}_{out}$ to the incoming irradiance E_{in} from direction $\vec{\omega}_{in}$ [14]:

$$f(\vec{\omega}_{in}, \vec{\omega}_{out}) = \frac{dL_{out}(\vec{\omega}_{out})}{dE_{in}(\vec{\omega}_{in})} = \frac{dL_{out}(\vec{\omega}_{out})}{dL_{in}(\vec{\omega}_{in}) \cdot \cos\theta_{in}} \quad [\mathrm{sr}^{-1}], \tag{2}$$

Physically plausible BRDFs must obey two restrictions: Helmholtz reciprocity and energy conservation.

*cezneluk@fel.cvut.cz
†havran@fel.cvut.cz

Helmholtz reciprocity defines the relationship between a light ray and its corresponding reverse ray that the value of the BRDF function must be the same:

$$\forall \vec{\omega}_{in}, \vec{\omega}_{out} \in \Omega : f(\vec{\omega}_{in}, \vec{\omega}_{out}) = f(\vec{\omega}_{out}, \vec{\omega}_{in}). \quad (3)$$

Energy conservation is a requirement that the overall outgoing energy cannot exceed the incoming energy:

$$\forall \vec{\omega}_{in} \in \Omega : \int_{\Omega} f(\vec{\omega}_{in}, \vec{\omega}_{out}) \cdot \cos\theta_{out} \, d\vec{\omega}_{out} \leq 1. \quad (4)$$

1.3 Rendering equation

The rendering equation [8] describes the total outgoing radiance in direction $\vec{\omega}_{out}$ at a specific point on the surface:

$$L_{out}(\vec{\omega}_{out}) = L_{emit}(\vec{\omega}_{out}) + \int_{\Omega} f(\vec{\omega}_{in}, \vec{\omega}_{out}) \cdot L_{in}(\vec{\omega}_{in}) \cdot \cos\theta \, d\vec{\omega}_{in}. \quad (5)$$

where L_{emit}, L_{in} are the emitted and the incoming radiance. Due to the integral over a hemisphere and an incoming radiance in it (resulting in the need of a recursive evaluation of the equation), the exact value of this function is computationally demanding. Therefore, this function must be approximated, even more so with real-time rendering.

2 Related work

In this section, we will describe two fields of study that are related to this work: BRDF data sets and analytical representation of BRDF.

2.1 BRDF data sets

The most well-known data set of measured BRDFs is the MERL BRDF database, produced by Matusik et al. [11] They used a spherically homogeneous sample, a stationary camera, and a light on a turntable. MERL database contains 100 different isotropic materials, each of which consists of 1,458,000 samples.

One of the newer data sets worth mentioning was produced by Dupuy and Jakob [4]. They invented an adaptive parameterization, using which they can measure and store only important parts of the BRDF 4D domain. By employing this method, their database currently includes 62 various materials.

2.2 Analytical models of BRDF

Currently, the vast majority of renderers rely on analytic models. One of the fundamental models is the Phong illumination model [15], which lacks both energy conservation and Helmholtz reciprocity. These problems were later solved by Lafortune and Willems [10], who represent the model as:

$$f(\vec{\omega}_{in}, \vec{\omega}_{out}) = \frac{k_d}{\pi} + k_s \cdot \frac{n+2}{2\pi} \cdot (\max\{\vec{v} \cdot \vec{r}, 0\})^n, \quad (6)$$

where $k_d \in [0,1]$, $k_s \in [0,1]$ ($k_d + k_s = 1$) are coefficients of the diffuse and specular part, $\vec{r}$ is a vector of ideal reflection of $\vec{l}$, and $n \in [0,\infty)$ is a parameter that defines the shininess of the material.

Today, more complex models are used. A frequently used model is, for example, the model developed by Walter et al. [21] It is based on the Cook-Torrance model, which, unlike Walter's model, does not satisfy energy conservation.

$$f(\vec{\omega}_{in}, \vec{\omega}_{out}) = k_d \cdot f_d(\vec{\omega}_{in}, \vec{\omega}_{out}, \lambda) + k_s \cdot f_s(\vec{\omega}_{in}, \vec{\omega}_{out}) \quad (7)$$

The specular part of the model is based on microfacet theory and is decomposed into three parts: Fresnel function F, geometric attenuation G, and distribution function D:

$$f_s(\vec{\omega}_{in}, \vec{\omega}_{out}) = \frac{F \cdot G \cdot D}{4 \cdot (\vec{n} \cdot \vec{l}) \cdot (\vec{n} \cdot \vec{v})}, \quad (8)$$

where $\vec{n}$ is a normal of the surface. For Fresnel function F in real-time graphics, the Schlick approximation [16] is used. Geometric attenuation G is influenced by the chosen distribution function F, for which several variants have been introduced. The most used is GGX:

$$D = \frac{\alpha^2 \cdot \max\{0, \vec{h} \cdot \vec{n}\}}{\pi \cdot \cos^4\theta_h (\alpha^2 + \tan^2\theta_h)^2}, \quad (9)$$

$$G \approx G_1(\vec{v}) \cdot G_1(\vec{l}), \quad (10)$$

$$G_1(\vec{x}) = \max\left\{0, \frac{\vec{x} \cdot \vec{n}}{\vec{x} \cdot \vec{h}}\right\} \cdot \frac{2}{1 + \sqrt{1 + \alpha^2 \cdot \tan^2\theta_x}}, \quad (11)$$

where $\vec{h}$ is a half vector ($\vec{h} = \frac{\vec{l}+\vec{v}}{||\vec{l}+\vec{v}||}$), θ_h is an angle between $\vec{h}$ and normal $\vec{n}$, and θ_x is an angle between $\vec{x}$ and normal $\vec{n}$. The parameter α defines the roughness of the material.

3 BRDF Measurements

Acquiring measured BRDF data is the first step in the workflow. We used MiniDiff v2 [17] (shown in Figure 2), a portable contact scatterometer created by Synopsys (previously LightTec). It supports the measurement of BRDF for isotropic materials at four angles of incidence for light sources: $\theta_{in} \in \{0°, 20°, 40°, 60°\}$. For each angle of light, it produces measurements of the RGB reflectance values in the range of $\phi_{out} \in [0°, 360°)$ and $\theta_{out} \in [0°, 75°]$ with a precision of 1°. Therefore, the measurement of a single sample consists of 324,000 BRDF values.

A limited number of material samples can be properly measured as a result of the construction of this instrument. Any sample that is not solid, is not homogeneous, is squashy, has bumps (such as plaster), contains tiny holes (like most fabrics), or is partially transparent (like certain types of plastic) will produce invalid results.

With these constraints, 216 measurements of materials were produced, mainly papers and swatches, but also

Figure 2: MiniDiff v2 with calibration samples.

metal, plastics, felt wool, and stiff foam. Some samples are wood (plywood, chipboard, planed wood), cloth, and leather. Anisotropic materials were measured for two ϕ_{in} angles of light that are approximately perpendicular and stored as independent measurements. The miniatures of all material samples are shown in Figure 9.

4 Processing

During the processing stage, two primary tasks need to be performed: extrapolation and export to the look-up table (LUT) image data. For this reason, we have created an application that also enables us to visualize data and verify its validity. This application was developed as modular and general as possible: internally it works with measurements as a point cloud, and loading data from a new file format can be easily added.

4.1 Extrapolation

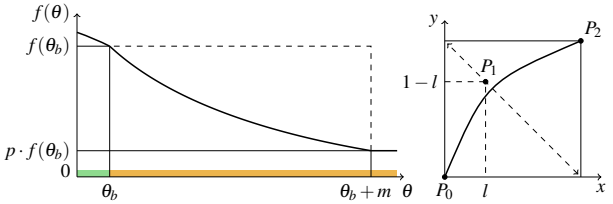
BRDF data for grazing angles ($\theta_{out} > 75°$) are unavailable, so we must extrapolate them from the measured range. Specifically, for each 3D texture slice corresponding to a particular θ_{in}, the values $f(\theta_e)$ are calculated from the value $f(\theta_b)$ of the nearest known measurement (in our scenario $\theta_b = 75°$) as an interpolation of the scale of $f(\theta_b)$ from 1 to the parameter $r \in [0,\infty)$:

$$\begin{aligned} f(\theta_e) &= f(\theta_b) \cdot ((1-\alpha) + \alpha \cdot p), \\ \alpha &= b\left(\min\left\{\frac{\theta - \theta_b}{m}, 1\right\}, l\right), \alpha \in [0,1], \end{aligned} \tag{12}$$

where m represents the maximum expected distance between the known measurement at θ_b and the calculated value at θ_e (in our scenario $m = 90° - 75° = 15°$). Figure 3a shows an example of extrapolated BRDF values. The function $y = b(x,l)$ can be described as finding the coordinate y of a point with the specific coordinate x on the restricted quadratic Bezier curve, illustrated in Figure 3b, specified by $l \in [0,1]$. This curve has a fixed starting point $\vec{P}_0 = (0,0)$, an ending point $\vec{P}_2 = (1,1)$, and a parameterized control point $\vec{P}_1 = (l, 1-l)$:

$$y = b(x,l) \text{ if } \exists t : \begin{bmatrix} x \\ y \end{bmatrix} = 2 \cdot (1-t) \cdot t \cdot \begin{bmatrix} l \\ 1-l \end{bmatrix} + t^2. \tag{13}$$

The equation for this function is solved by testing the roots of the variable t.



(a) Chart of BRDF values by θ (extrapolation area is orange).

(b) Used Bezier curve.

Figure 3: Extrapolation of BRDF data with the parameters $p = 0.2$ and $l = 0.3$.

This extrapolation was developed to be easily computed. If parameter $p = 1$, this extrapolation produces the same results as clamping values outside the measurement region, which is preferable for materials that exhibit mainly diffuse reflection, because it is expected that the BRDF value will not depend much on the outgoing direction $\vec{\omega}_{out}$. For glossy materials, it is recommended to set the parameter $p < 1$, because decreasing the value of the BRDF with increasing θ_{out} (consequently increasing the distance from the direction of ideal reflection) will roughly estimate the shape of the reflection lobe.

4.2 Export to LUT image

During rendering, the isotropic BRDF data are stored as a 3D texture constructed from several 2D textures, slices with the fixed third texture coordinate. The isotropic BRDF has three independent parameters and has $\phi_{in} = 0$. Therefore, each slice represents the BRDF values for a specific θ_{in} in an equirectangular projection with a spherical coordinate system $\vec{\omega}_d = (\theta_d, \phi_d)$ aligned with the direction of specular reflection, i.e. the direction $\theta_d = 0°$, $\phi_d = 0°$ corresponds to $\theta_{out} = \theta_{in}, \phi_{out} = \phi_{in} + 180°$. Slices are arranged in row-major order in an image file with ascending θ_{in} in a single composite image.

For rendering in Unreal Engine, it is necessary to generate a single texture (referred to as an atlas) that merges all measured BRDF samples being used (see Section 5.3). An example of an atlas is illustrated in Figure 5. The layout of an atlas is the same as that for a single BRDF, but instead of the slice itself, the atlas contains a regular grid. Each cell in the grid corresponds to a slice of a single BRDF texture with the same θ_{in}. Cells within the grid are organized similarly to slices in 3D texture, following a row-major order.

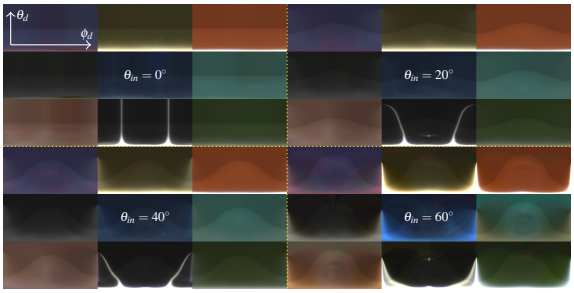


Figure 5: An example of an exported atlas image for 9 different materials.

4.3 Previewing BRDF data

The developed application supports two ways of previewing the measured BRDF data: as a reflection lobe for a specific angle of light $\vec{\omega}_{in}$ and rendering of a 3D object illuminated by an environment map.

The reflection lobe, illustrated in Figure 4a, can be described as a deformed sphere where the distance between the points from the origin corresponds to a specific BRDF value in a particular direction. This visualization method is beneficial for verifying the validity of specific values and gaining insight into the general form of a BRDF.

In contrast, rendering of a 3D object illuminated by an environment map, shown in Figure 4b, can be practical for comparing the real-world and rendered versions of the material. It is computed as a numerical integration of the rendering equation over a hemisphere, where the input radiance is sampled from the environment map in a particular direction.

5 Rendering in game engines

We implemented a shader for two major game engines, Unity and Unreal Engine 5, which compute direct lighting using a 3D LUT image created in the previous step.

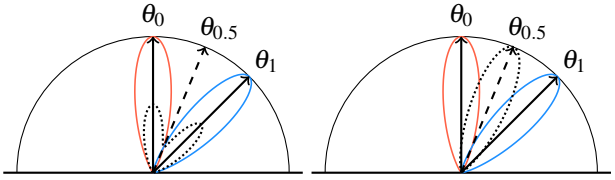
5.1 BRDF evaluation

In practical terms, we assume that the initial slice of the texture corresponds to $\theta_{in,0} = 0$, with each subsequent slice increasing its angle linearly (i.e. $\theta_{in,i+1} = \theta_{in,i} + \Delta$; $\Delta \in \mathbb{R}^+$, specifically in our case $\Delta = 20°$). Consequently, the value of a BRDF point in a normalized texture coordinate can be represented as:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \frac{\phi_d}{2\pi} \\ \frac{\theta_d}{\pi} \\ \frac{\theta_{in}}{\theta_{in,m}} \end{bmatrix}, \tag{14}$$

where $\theta_{in,m}$ is an angle of the last 2D slice in the 3D texture.

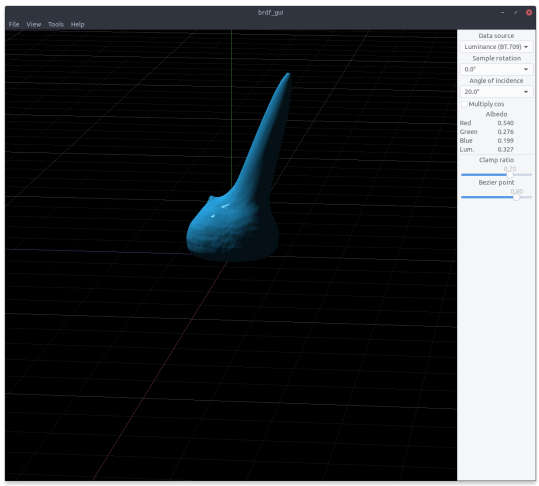
To read the BRDF data from the texture, native trilinear interpolation is used. In the selected coordinate system described in Section 4.2, two points with the same (u,v) in a different 3D texture slice (different texture coordinate w) describe the change of the lobe around the specular reflection, which is more valuable information than the linear interpolation of the BRDF values with fixed $\vec{\omega}_{out}$, because it is expected that the main difference between the 3D texture slices will be in the specular part of the BRDF. For example, in Figure 6, interpolation with fixed $\vec{\omega}_{out}$ produces two smaller lobes, which does not represent the correct behavior.



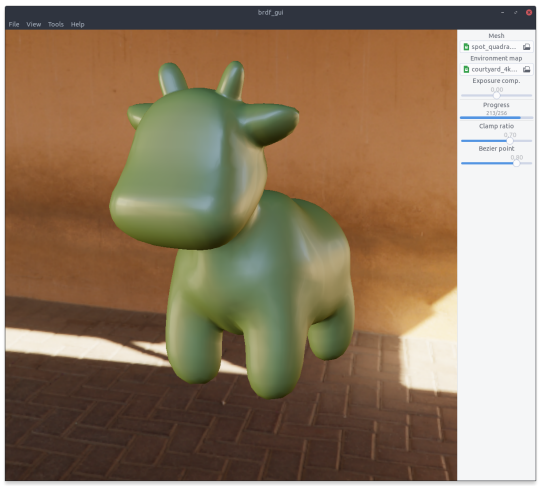
(a) Interpolation between slices parameterized with $\vec{\omega}_{out}$.

(b) Interpolation between slices parameterized with $\vec{\omega}_d$.

Figure 6: Comparing of interpolations of BRDF values (dotted line) of the cosine lobe $(\vec{r} \cdot \vec{v})^{20}$ ($\vec{r}$ is a direction of ideal reflection) between slices with specified θ_{in}.



(a) A reflection lobe.



(b) 3D object illuminated by an environment map.

Figure 4: Implemented methods for previewing BRDF data.

This approach is inappropriate for materials that exhibit substantial retroreflective properties, but such cases are not within the scope of this work.

BRDF is evaluated in a tangent space of the specified point on a 3D object surface. This tangent space, illustrated in Figure 8, is derived from the normal vector $\vec{n}$ and the tangent vector $\vec{t}$, which is formed by projecting the incoming direction $\vec{\omega}_{in}$ onto a plane perpendicular to the normal $\vec{n}$.

To evaluate the BRDF values from the 3D texture, it is necessary to transform the view direction $\vec{\omega}_{out}$ into $\vec{\omega}_d$ in the texture coordinate system. This can be done through the rotation matrix from normal $\vec{n}$ to the direction of specular reflection $\vec{\omega}_r = (\theta_{in}, \pi)$:

$$S = (R_z(\pi) \cdot R_y(\theta_{in}))^T = \begin{pmatrix} -\cos\theta_{in} & 0 & -\sin\theta_{in} \\ 0 & -1 & 0 \\ -\sin\theta_{in} & 0 & \cos\theta_{in} \end{pmatrix}. \tag{15}$$

After rotation, the spherical coordinates of the view direction $\vec{v}$ are related to $\vec{\omega}_r$, therefore they are $\vec{\omega}_d$. Hence, the normalized texture coordinates are calculated using this formula:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \frac{f_\phi(S\cdot\vec{v})}{2\pi} \\ \frac{f_\theta(S\cdot\vec{v})}{\pi} \\ \frac{f_\theta(\vec{l})}{\theta_{in,m}} \end{bmatrix}, \tag{16}$$

where the function f maps Cartesian coordinates to spherical, $\vec{l}$ and $\vec{v}$ represent the light and the view direction in the introduced tangent space.

To perform testing and performance evaluations, the same testing scene was established in both game engines. This scene contains six models (specifically Stanford Bunny [18], Stanford Armadillo [9], Phlegmatic Dragon [7], Stanford Dragon [3], Utah Teapot [13] and Spot [2]) with six different materials. The rendered image from this scene is shown in Figure 7.

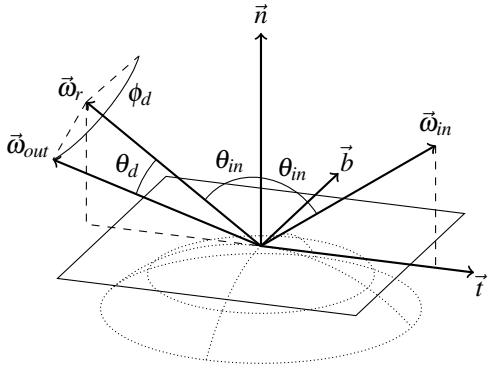


Figure 8: The tangent space used for BRDF evaluation.

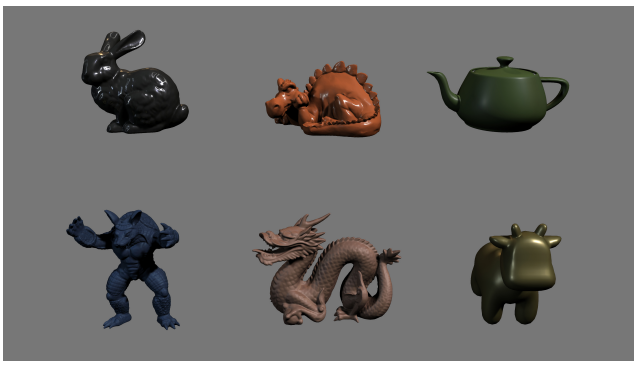
5.2 Unity

Unity [20] has three rendering pipelines: build-in, universal (URP), and high definition (HDRP). The shader for rendering measured BRDF was implemented for the built-in render pipeline because modifying a BRDF evaluation in other pipelines is not officially supported and requires a bit of reverse engineering [22].

The built-in rendering pipeline uses forward rendering by default, and therefore the implementation of the shader was straightforward. Each material has an input texture with BRDF data and evaluates the lighting in the fragment shader for each light that affects the object [19].

5.3 Unreal Engine 5

Unreal Engine [6] has a pipeline based on deferred shading. A shader, editable by a user, writes the necessary data for lighting (such as position, normal, and BRDF parameters) to the G-buffer (an off-screen framebuffer) [12]. In the G-buffer, it is not feasible to store the whole texture with measured BRDF data. Instead, it is necessary to store only an index of a sample in the atlas.



(a) Render with measured BRDF materials.

(b) Render with the Lafortune-Phong materials.

Figure 7: The test scene rendered in Unity (it appears almost identical in Unreal Engine). Differences between the analytical model and measured BRDFs are mostly noticeable in a specular part, where the highlights of the Lafortune-Phong model do not have soft endings.

During the second pass, lighting is calculated for every pixel on the screen based on data stored in the G-buffer. This pass is a part of the rendering engine, and making changes to the BRDF evaluation requires directly editing the source code of the engine. It involves modifying the shader and structure of the G-buffer, editing the UI components in the material editor, and incorporating some logic for manipulation with the atlas [1].

6 Discussion

6.1 Performance

The rendering speed was evaluated in the test scene in both game engines. The shader using measured BRDF was compared with the shader using the Lafortune-Phong analytic model [10]. This choice was determined due to differences in default shader models between game engines. Additionally, default shaders provide support for indirect lighting, which the current shader implementation does not offer.

Performance measurements were performed on a Linux PC with Intel i5-9600K @ 4.5 GHz CPU and Nvidia GeForce GTX 1660 GPU. Both game engines use the Vulkan API for rendering. The measured BRDF shader is slightly slower (approx. $2-3\%$).

	Unity 2022.1.19f1	Unreal Engine 5.1.1
Measured BRDF	10.05 ms (99.5 fps)	12.37 ms (80.9 fps)
Lafortune-Phong	9.72 ms (102.9 fps)	12.12 ms (82.5 fps)

Table 1: Average rendering time of the test scene with three directional lights on 4K resolution. The average was calculated from 15 seconds run with 5 seconds warm-up.

6.2 Drawbacks

The primary disadvantage of this method is the increased memory usage caused by the requirement to store the LUT texture. Each measurement from the created dataset took around 1 MB of VRAM ($361 \cdot 181 \cdot 4$ pixels in RGB9_E5 format, 4 bytes per pixel), but a LUT texture with denser measurements (which means higher resolution of the texture) will require significantly larger amounts of memory.

Another drawback is the restriction of rendering only direct lighting. For performance reasons, game engines make some assumptions that cannot be easily fulfilled with tabularized BRDF data. For example, Lumen Global Illumination in Unreal Engine uses a single simplified analytic model [5].

Figure 9: Miniatures of all measured samples in the dataset.

6.3 Future work

One of the possible improvements is to expand the implemented shader to support anisotropic materials. It will require manually rotating the scatterometer or another scatterometer with anisotropic support and extending the LUT texture to four dimensions. Because 4D textures are not generally supported, performing linear interpolation along a single axis needs to be realized within a shader by additional steps. Furthermore, memory consumption will increase even further, necessitating the implementation of some form of data compression.

7 Conclusion

In the preceding sections, we explained the utilization of measured reflectance for real-time rendering in computer graphics. We described the process from acquiring measured BRDF data, and processing them, to computing the radiance of a pixel in a shader. A total of 216 material samples were measured, and a tool was developed to process and visualize the data. Although the method outlined may have some limitations, in specific scenarios, it may be more appropriate than a generic analytic model.

References

[1] 0ne3y3. New shading models and changing the GBuffer, 12 2022. https://dev.epicgames.com/community/learning/tutorials/2R5x/unreal-engine-new-shading-models-and-changing-the-gbuffer [Accessed 2024-03-04].

[2] Keenan Crane, Ulrich Pinkall, and Peter Schröder. Robust fairing via conformal curvature flow. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013.

[3] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 303–312, New York, NY, USA, 1996. Association for Computing Machinery.

[4] Jonathan Dupuy and Wenzel Jakob. An adaptive parameterization for efficient material acquisition and rendering. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 37(6):274:1–274:18, November 2018.

[5] Epic Games. ShadingModels.ush, Unreal Engine source code. https://github.com/EpicGames/UnrealEngine/blob/5ca9da84c694c6eee288c30a547fcaa1a40aed9b/Engine/Shaders/Private/ShadingModels.ush#L343 [Accessed 2024-03-04].

[6] Epic Games. Unreal Engine, 1998–2024. https://www.unrealengine.com/en-US.

[7] Jiří Filip, Radek Holub, Vlastimil Havran, Jaroslav Křivánek, and Daniel Sýkora. Phlegmatic Dragon, 2007. https://web.archive.org/web/20220710054500/https://dcgi.fel.cvut.cz/cgg/eg07/index.php?page=dragon [Accessed 2022-07-10].

[8] James T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, page 143–150, New York, NY, USA, 1986. Association for Computing Machinery.

[9] Venkat Krishnamurthy and Marc Levoy. Fitting smooth surfaces to dense polygon meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 313–324, New York, NY, USA, 1996. Association for Computing Machinery.

[10] Eric Lafortune and Yves Willems. Using the Modified Phong Reflectance Model for Physically Based Rendering. Technical Report CW 197, Department of Computing Science, K.U. Leuven, 11 1994.

[11] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3):759–769, July 2003.

[12] Michael Muir. Unreal Engine Lighting. https://dev.epicgames.com/community/learning/tutorials/Le7b/unreal-engine-lighting [Accessed 2024-03-04].

[13] Martin Newell. Utah Teapot, 1975. https://graphics.cs.utah.edu/teapot/ [Accessed 2024-03-09].

[14] F.E. Nicodemus, J.C. Richmond, J.J. Hsia, W.I. Ginsberg, and T. Limperis. Geometrical considerations and nomenclature for reflectance. *Applied Optics*, 9:1474–1475, 1977.

[15] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, 6 1975.

[16] Christophe Schlick. An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum*, 13(3):233–246, 1994.

[17] Synopsys. *MiniDiff V2 User's Manual*, 2021.

[18] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, page

311–318, New York, NY, USA, 1994. Association for Computing Machinery.

[19] Unity Technologies. Rendering paths in the Built-in Render Pipeline. https://docs.unity3d.com/2022.1/Documentation/Manual/RenderingPaths.html [Accessed 2024-03-03].

[20] Unity Technologies. Unity, 2005–2024. https://unity.com/.

[21] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. page 195–206, 2007.

[22] Bronson Zgeb. Custom Lighting in URP with Shader Graph, 2021. https://bronsonzgeb.com/index.php/2021/10/04/custom-lighting-in-urp-with-shader-graph/ [Accessed 2024-03-04].

Multiresolution Mesh Rendering Engine Practicalities and Performance

Maxwell W. Pettett*
Supervised by: Rafał K. Mantiuk†

Department of Computer Science and Technology
University of Cambridge
United Kingdom

Abstract

A multiresolution mesh is a structure that allows multiple levels of resolution of a mesh to be sampled in different regions. They are used to accelerate the construction of view-dependent Levels of Detail (LODs) for real-time rendering, generally for complex objects that may span large depths (e.g. terrain). Nanite, introduced in Unreal Engine 5, is an example of a full multiresolution pipeline. We describe our mesh-shader based multiresolution rendering engine in Vulkan, with two implementations to extract view dependent LODs. The first implementation is based on the approach established by Nanite. Our alternative implementation has no intermediate buffers at the cost of less fine-grained control over regions of the multiresolution we explore. We finally evaluate the two methods against each other and traditional LOD chains, emphasising practicality and performance.

Keywords: Modeling and Geometry processing, Real-time Graphics, Rendering

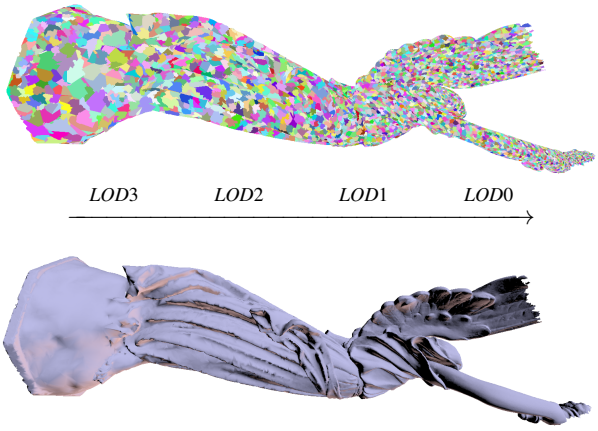


Figure 1: View-dependent LOD generated from the Stanford Lucy model (28 million source polygons) [11]. The statue's base is visibly lower quality than the top.

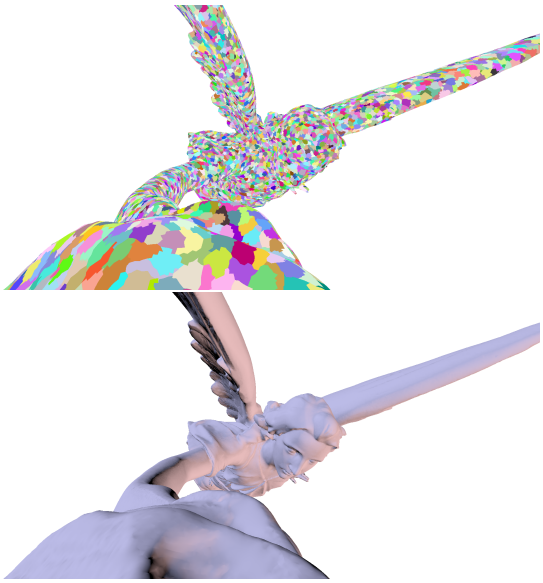


Figure 2: Figure 1's view from the camera. Note that cluster sizes are mostly uniform, excepting those close to the camera which have reached maximum resolution.

1 Introduction

A common desire for higher-fidelity scenes in modern rendering engines has brought higher and higher resolution meshes to real-time applications. Handheld photogrammetry applications have made sourcing such meshes simpler and more commonplace. Varying mesh resolution is typically used in real-time rendering engines to maintain performance in complex scenes. This is traditionally implemented with a series of coarser and coarser approximations of the mesh, a Level of Detail (LOD) chain. However, LOD chains are limited in flexibility. Each object can only be rendered at a single resolution, despite the possibility that the same object spans large depths (e.g., terrain), and, therefore, there is no single optimal LOD.

A *multiresolution mesh* is a data structure that stores geometry information at multiple levels of resolution. It is an alternative to, and generalisation of, LOD chains, with fine-grained control over rendering that can tackle their disadvantages. However, fidelity improvement may

*mp2015@cam.ac.uk
†rafal.mantiuk@cl.cam.ac.uk

be deemed too expensive if the cost to calculate the view-dependent LOD is higher than that of rasterisation at a higher mesh resolution, so methods to render multiresolution meshes must be fast and scalable. Our main point of comparison as a modern multiresolution pipeline is Unreal Engine 5's Nanite, which is embedded in the engine and so difficult to study or extract.

This work leverages the introduction of mesh shaders on modern hardware, which operate by emitting small clusters of triangles rather than vertices. This paper opens with background on multiresolutions, the error functions required to generate view-dependent LODs, and mesh shaders. The implementation section then describes two methods for rendering view-dependent LODs. Our first implementation, *DAG Explore*, is based on ideas Nanite's *persistent threads*[7]. In contrast, the second implementation, *Task Select*, is developed in this work. It relies on mesh shading to insert LOD logic into our draw calls, without a compute pass or intermediate data. We then evaluate their performance and practicality.

2 Background

Many forms of multiresolution exist with different characteristics and drawbacks. Hoppe introduced **progressive meshes** [6] in 1996. A progressive mesh is a multiresolution mesh encoded as a low-resolution base mesh, and the vertex splits required to raise resolution. Quick-VDR [12] expanded on progressive meshes with an initial coarse-grained selection before vertex local transformations.

Further techniques, such as BDAM [2], or Adaptive Tetrapuzzles [3], focus more on the coarse-grained selection, using spatially based partitions for 2D and 3D surfaces, respectively. Their partitions contain geometry in patches that can be substituted, moving further from vertex transformation and decimation techniques. Batched Multi Triangulation [1] extends geometry patches to a generic framework for multiresolutions based on a Directed Acyclic Graph (DAG) of patches, the approach our renderer will be based on. Ponchio's thesis is an excellent comparison of the above methods [10].

2.1 The Multiresolution DAG

This section introduces the multiresolution mesh as a DAG of **clusters**, uniformly sized patches of triangles, described in detail in [1]. Figure 1 shows clusters selected from a multiresolution of a high-resolution mesh. A requirement of a multiresolution scheme is to ensure that clusters of neighbouring resolution levels can be interleaved without seams introduced by mesh simplification. To illustrate the difficulty of this problem, let us consider a simple scheme:

1. Start from a set of clusters that partition a mesh.
2. Recursively, merge pairs of clusters together and simplify their contents. Edges on the boundary of the pairs are *locked*, so are not moved by the simplifier.

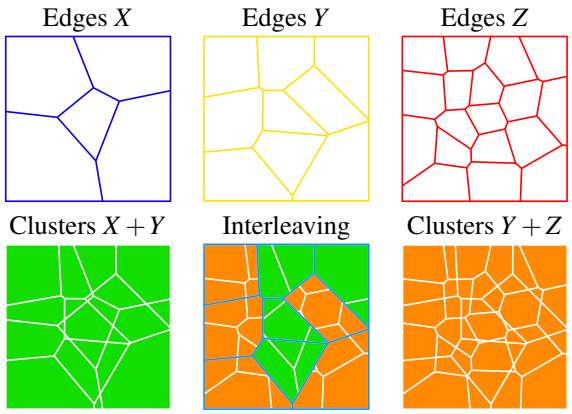


Figure 3: Example of locked edges forming clusters allowing for interleaving. 3 sets of locked edges (X,Y,Z) are merged into 2 sets of clusters $(X+Y,Y+Z)$, which can be interleaved using their shared locked edges (Y).

This scheme would form a tree of variable resolution clusters of the mesh. However, in doing this it will lock some edges from the highest resolution to the lowest, restricting the flexibility of mesh simplification. At the extreme, it will bisect the mesh with a high-resolution ring of edges, harming the quality of lower resolution clusters. To avoid this artifact, we need an alternate method that allows edges to be *unlocked*.

The multiresolution mesh scheme we use contains two sets of locked edges at each level of detail, one set for compatibility with each of the lower and upper levels. Figure 3 shows two adjacent levels and a selection of clusters from both, made possible due to their shared set of locked edges.

Such selections can only be made if we can guarantee they will approximate the original mesh, so containing no overlaps or holes. We use a DAG to encode relations between clusters to allow us to make confident selections. Nodes in the DAG represent clusters in the multiresolution, from all levels. Edges represent dependence between clusters, a relation of mutual exclusion, i.e. overlap. This property is transitive, therefore we only include dependence between adjacent levels on our DAG [1].

Our method of generating DAGs follows the Nanite *Cluster - Group - Simplify - Recluster* scheme [7]:

1. Start from a set of clusters that partition a mesh.
2. Partition clusters into **groups**, collections of around 4 adjacent clusters.
3. Simplify within groups, locking border edges.
4. Partition each group into *two* new clusters, which become the *two* parents of the group. Return to 2.

This is a generalisation of the previous method, replacing a one-parent-two-children relationship with two-parents-four-children. A DAG for a small mesh generated with our program is shown in Figure 4, in which these structures can be identified. Smaller clusters are more flex-

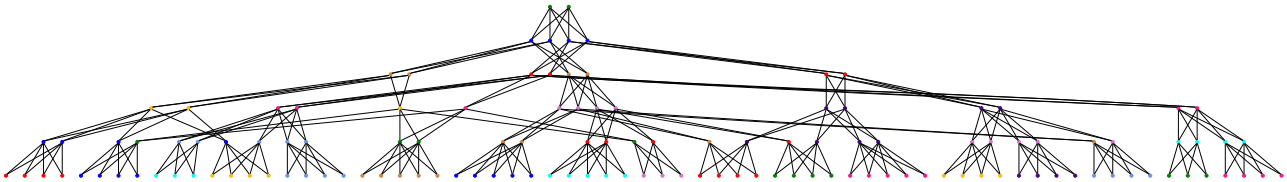


Figure 4: A DAG for a multiresolution encoding of a simple sphere mesh. Nodes represent clusters, edges represent dependencies between clusters. Nodes of the same colour share the same two parents, and so make up a group.

ible, but maintain a worse area to perimeter ratio, resulting in more locked edges at each stage and less efficient simplification; we choose to use clusters of around 300 triangles. There is a chance the new parent clusters do not overlap with all child clusters, resulting in some false dependencies in the DAG. However, parents sharing identical sets of children allows for efficient DAG traversal.

A valid selection of clusters will cover the area of the mesh, with no clusters overlapping. Ensuring the selection contains no overlapping clusters requires that no two clusters in the selection are dependent on each other. To cover the full mesh area, every path from a root to a leaf must contain a single selected node. Our DAG structure guarantees this selection will contain no seams [1].

A **dicut** is a cut into two subsets such that any cut edges connecting the two subsets share the same direction. If we select the leaf nodes of a dicut subset that includes the root, we have a valid set of clusters. This is due to two factors. First, the selected clusters will not contain overlapping geometry, as they satisfy all dependency relations. Second, we have no holes, as our selected clusters descend from the (two) roots. The root clusters cover the entire mesh area, so the sum of all descended clusters (satisfying dependency relations) area must also cover it [1].

The easiest DAG to imagine is a traditional LOD chain. Each layer's node is dependent on the next, as they overlap in area (the entire mesh), and we sample by selecting a leaf of a dicut set (any single node).

2.2 View-dependent LOD from a DAG

To generate a view-dependent LOD, it is useful to define an error function on clusters that allows us to estimate their **screen-space errors**. We then can compare this error to a user-defined threshold, , that defines the target mesh resolution. This screen-space error is projected from an object-space error δ of a cluster. The exact definition of the object-space error varies per-implementation, but we will use the average edge length of a cluster, similar to batched multi-triangulation [1]. This represents triangle density within a cluster, and is comparable between clusters as their triangle counts are roughly constant.

To convert error from object-space to screen-space, we assign each cluster a bounding sphere, with centre c and radius r, a spherical volume in object-space that bounds the cluster. We then use a method similar to [2] to project the object-space error of some cluster i to screen-space, $\text{err}(i)$, for eye position e.

$$\text{err}(i) = \frac{(\delta_i + r_i)^2}{||c_i - e||^2}, \tag{1}$$

An important feature of the error function is that it is monotonically decreasing down the DAG.[1] To ensure the screen-space area of clusters is monotonic, we assign each cluster's bound such that it contains all bounds of their children, turning the DAG into a nested boundings volume hierarchy [1]. Object-space error, clusters' average edge length, also monotonically decreases as clusters double their triangle density at each level.

2.3 Mesh Shaders

This paper references **mesh shaders**, a concept shared between modern graphics APIs that readers may not be familiar with. We will use the Vulkan implementation and terminology. Mesh shaders attempt to solve some shortcomings of using the traditional graphics pipeline for procedural geometry. The traditional pipeline includes tessellation, geometry, and vertex stages, that can be used for procedural geometry, but each with a limited view and control of parts of the source data.

The common way to program procedural geometry has shifted away from the graphics pipeline with the wide adoption of compute shaders, due to their flexibility and good support. Mesh shaders attempt to bring this flexibility to the graphics pipeline by stripping out everything other than the fragment stage of the pipeline, and adding a **mesh stage**. The mesh stage has all the semantics and capabilities of a compute shader, with the additional ability to emit triangles, up to a maximum primitive limit per workgroup [8]. These will output our clusters, and save writing to an intermediate index buffer.

Additionally, a similar **task stage**[2] is added, which, instead of emitting triangles, can emit mesh shaders. We will utilise this to insert LOD logic directly into the draw call. This grants us a large amount of flexibility for programming procedural geometry, although it is not as powerful as the ability to generically launch threads on the GPU.[3]

[1] Root nodes have the highest error, as they have the least polygons.
[2] Known as the Amplification stage in DX12.
[3] See **Work Graphs** [9], that will allow generic kernel invocation.

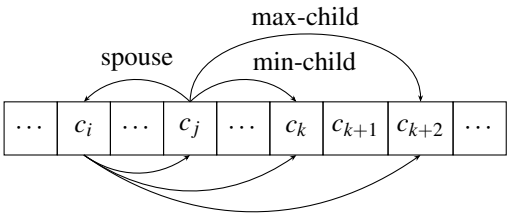


Figure 5: A diagram of our cluster structure. Pointers within the diagram from cluster c_j correspond to a cluster $c_j = [\text{spouse} = i,\ \text{min-child} = k,\ \text{max-child} = k+2]$.

3 Implementation

This section will look at the two approaches described and examined in this paper. Older view methods sample their multiresolutions on a separate CPU core, referred to as out of core [2, 1], however compute and task shaders allow us to do this work efficiently on a GPU.

Both implementations to render a view dependant LOD of a mesh are supplied with:

- A *cluster buffer*, containing the DAG of clusters that make up the multiresolution (Fig. 5).
- Instance information, containing the model matrix of the instance of a multiresolution we are drawing.
- Camera information, the view-projection matrix.
- A screen-space error target, τ, the maximum screen-space error a cluster can have to be drawn.

The first approach, *DAG Explore* (§3.1), aims to output all clusters that should be drawn into a buffer, searching the DAG for suitable clusters recursively from the root. DAG Explore is similar to Nanite's Persistent Threads implementation for generating a view dependent resolution.

The second approach, *Task Select* (§3.2), aims to use the programmable task stage of the mesh pipeline (§2.3) to eliminate the need for intermediate memory.

3.1 DAG Explore LOD Generation

A typical instance of a multiresolution in a scene will have most of its area filled with lower resolution clusters. Selecting a low resolution cluster will instantly invalidate the many higher resolution clusters that descend from it. Testing these clusters would be wasted time, so we want to avoid exploring the entire DAG. This method will traverse the DAG recursively, starting from the roots. As we are searching for leafs of a dicut subset, this must encounter all clusters that should be drawn.

Traversing the DAG requires care. It is not a tree, so there are clusters that share children, but to traverse the DAG efficiently we should not explore the same cluster twice. Our DAG is, however, shaped similarly to a tree; it is formed of pairs of clusters that share identical children; we say clusters in such a pair are **spouses**. We can then view the DAG as a tree by only regarding the children of one cluster in each of these pairs, illustrated in Figure 6. In Alg. 1, we only explore the children of the spouse with the smaller index.

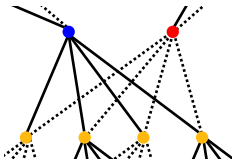


Figure 6: A group of clusters whose parents are in separate groups. Solid lines represent the edges traversed to explore the DAG as a tree.

Algorithm 1 DAG Explore, breadth first search

```
queue ← root-nodes
draw-buffer ← []
draw-count, head ← 0
tail ← |root-nodes|
while queue not empty do
    i ← queue[head]
    head ← head + 1
    cluster_i ← clusters[i]
    if err(i) < τ then                    ▷ Cluster should be drawn
        draw-buffer[draw-count] ← cluster
        draw-count ← draw-count + 1
    else if i < cluster_i → spouse then   ▷ DAG as tree
        for c in cluster_i → children do
            queue[tail] ← c
            tail ← tail + 1
        end for
    end if
end while
```

3.1.1 Multiqueue

Algorithm 1 does not appear GPU-friendly, as it leverages a single shared queue. We must allow multiple invocations synchronised access to the queue to maintain parallelism. Atomic buffer operations are too slow for this use case.

Our solution relies on subgroup[4] arithmetic (introduced in Vulkan 1.1) to synchronise queue access. In Vulkan terminology, a subgroup is a set of invocations executing in lockstep on the GPU. Subgroups will always be part of the same workgroup, a set of invocations with shared memory, but a workgroup may maintain multiple subgroups. Invocations in a subgroup may be active or inactive depending on factors such as dynamic branching, and an inactive invocation will not contribute to subgroup arithmetic results.

Subgroup arithmetic allows invocations to communicate via reductive operations, with each invocation submitting data. The most straightforward subgroup operation we use is `subgroupAdd(1)`, which will return the

[4] the Vulkan term; in AMD these are waves, in Nvidia, warps.

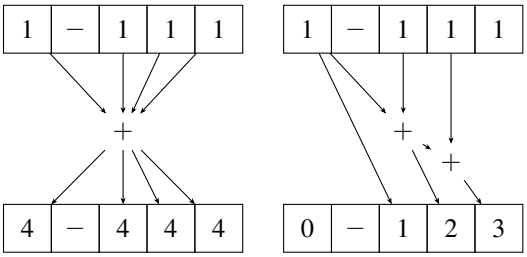


Figure 7: The subgroup operations `subgroupAdd(1)` (left) and `subgroupExclusiveAdd(1)` (right). Each block is a invocation within a single subgroup, with reduction operations linking inputs to results.

number of active invocations, as shown in Fig. 7 (left).

To implement DAG Explore, we limit the size of our workgroups to ensure they only contain one subgroup, which is generally 32 or 64 invocations, ensuring we can rely on subgroup arithmetic. This is the main source of difference from Nanite; persistent threads share work between workgroups, which relies on undefined GPU forward progression scheduling behaviour [7], while ours is limited to a single workgroup per instance. This will limit our latency of rendering a scene to processing the largest mesh, however, we assume scenes are highly populated and so parallelizable.

Algorithm 2 Synchronised Queue Pop

```
int idx = gl_LocalInvocationID.x;
int cluster = queue[head + idx];
head += subgroupAdd(1);
```

Algorithm 2 uses subgroup arithmetic to synchronize invocations each taking a cluster from the queue. To pop a unique item for each invocation, we can offset the queue head pointer by each of their **local invocation ID**s, their indices starting from 0 within the workgroup. This, however, leaves us with conflicting information about the true head of the queue across invocations. The queue may contain a number of clusters fewer than the size of our subgroup, which will result in some number of invocations being inactive. We solve this by incrementing the head pointer by the number of active invocations using subgroup arithmetic, ensuring the data is synchronized.

The more complex operation is appending children to the queue. We do not know how many children each node has, so we cannot simply offset by our local invocation ID when pushing. We know the number of items each invocation will add onto the queue, so we can allocate blocks in the queue upfront. To allocate blocks, we can use a more advanced subgroup command, `subgroupExclusiveAdd` (see Fig. 7 (right)), which will perform an exclusive addition across all active threads *in one call*. The return value for this will be an allocated index for each invocation to write to, used in Algorithm 3. We synchronize the queue afterwards by taking

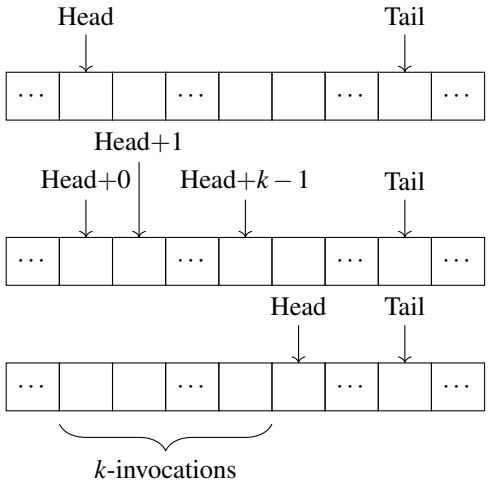


Figure 8: The positions of head and tail indices through Algorithm 2. After adding idx, each head points to a unique cell, then adding `subgroupAdd(1)` (= k), the head returns to being synced.

`subgroupMax(tail)`.

Algorithm 3 Synchronized Queue Push with Subgroup Arithmetic

```
cluster_t c = clusters[i];
int children = c.max_child_index
             - c.min_child_index + 1;
tail += subgroupExclusiveAdd(children);
for (int child_i = c.min_child_index;
         child_i <= c.max_child_index;
         child_i++) {
    queue[tail] = child_i;
    tail += 1;
}
tail = subgroupMax(tail);
```

3.1.2 Emitting clusters

Clusters with sufficiently low error must be emitted to be drawn. We push them to a *draw buffer* similarly to Alg. 3, but with a maximum of 1 cluster pushed. In our implementation, the draw lists for DAG Explore has enough space for the worst case (full resolution) of every instance. As future work, this size could likely be optimised, as we do not expect the worst-case full draw for every instance.

A task shader will then then emit mesh shaders for each cluster in the draw buffer. The number of clusters DAG Explore has emitted at this point is only recorded on the GPU, but must be communicated to our draw call to render, as invoking a task for each item in the buffer when many are empty would be wasteful. Indirect Dispatch is a common technique to allow some parameters of commands to be supplied by the contents of a buffer in the GPU, which mesh shaders support. This saves possible wasted bandwidth and latency sending the same counter

back and forth from the GPU. After filling our draw buffer, we set the *group count* parameter of `DrawMeshTasks` in our indirect buffer to the exact number of clusters to draw.

Extending this to drawing many instances is not complex. Using atomics, we can assign each instance's emitted clusters a space in a shared draw buffer. Each instance's draw call can then be instructed to read clusters from that range. `DrawMeshTasks` also supports the **multidraw** extension, which allows us to store the indirect arguments for multiple draw calls in a single buffer. A single buffer for draw data means we can additionally invoke the compute stages of all instances in a scene in a single command.

3.2 Task Select LOD Generation

DAG Explore does a lot of work and requires shared memory in picking which clusters to draw, queuing no further clusters once the boundary of the cut has been found. We present an alternative method that does not require shared memory and is simpler to implement.

The task shader allows us to integrate computations within the graphics pipeline. Integrated computation allows us to select view-dependent LOD without intermediate buffers, a method we will call *Task Select* LOD. This method invokes a task invocation on the GPU for every cluster and emits geometry to draw if the cluster has an error below the threshold, but parents with errors too great, with additional care to ensure the result has no holes.

3.2.1 Local Cut Selection

DAG Explore explores the DAG recursively, aiming to find the cut, made up of clusters whose error is just low enough to pass the threshold value τ. However, to enable the most parallelism, it is preferable to be able to test if a cluster should be included in the cut based on only itself and local neighbours. This is possible as our DAG has a single unique cut, as the screen-space error we compare decreases monotonically through clusters.

Algorithm 4 Local Cut

$\text{parent-err} \leftarrow \min(\text{err}(c_i \rightarrow \text{parent}_0),\ \text{err}(c_i \rightarrow \text{parent}_1))$
$\text{this-err} \leftarrow \min(\text{err}(i),\ \text{err}(c_i \rightarrow \text{spouse}))$
$\text{draw} \leftarrow (\text{this-err} \leq \tau) \wedge (\text{parent-err} > \tau)$

Each cluster's task invocation must make the decision of what fills its group's area; the group, or the two parents (if either), and draw the cluster if appropriate. This should be agreed by each cluster in the area implicitly, with no communication. This is done by assigning each cluster the error and bounding volume of the group as a whole, similarly to the bounding volume hierarchy of [2].

Algorithm 4 determines if a cluster is on the edge of the cut, and so should be drawn, based on the relation of its parent's errors to its own. The two parents are likely members of different groups, so likely have differing screen-space error. To compensate, Alg. 4 takes the minimum of the two. Comparing this against the error of this cluster would then leave a hole if $err(c_i \rightarrow \text{parent}_0) > \tau > err(c_i \rightarrow \text{parent}_1)$, as only one of the two parents would be drawn. To resolve this, Alg. 4 takes *this-error* to the minimum of the cluster's own error and that of the spouse. This fills the hole described above, as the previously missing parent will now draw based on its spouse's lower error.

Finally, some nodes in the DAG have no children or no parents, being the leaves and the roots. In these cases, we assume the error of the root's parent is ∞, and the error of the leaves children are $-\infty$. This ensures that for any finite value of τ, we will select a complete cut.

3.2.2 Task Shader Indirect Dispatch

A major issue with this approach would be wasted work in clusters that are too high resolution to be drawn. Such high-resolution clusters will make up the majority of most instances. For example, drawing a mesh at the first level of simplification in DAG Explore will, on average, only check half of a multiresolution's clusters, as the source mesh represents 50% of total triangles.

This method uses the same indirect dispatch draw call as §3.1.2. Task shaders can write to buffers just as compute shaders can, so, if we bind our indirect arguments buffer to the task shader, we allow ourselves to alter the number of tasks we invoke on the next dispatch.

We arrange our cluster buffer such that lower indices represent clusters at lower resolutions, meaning dispatching fewer tasks than there are clusters will cap the maximum resolution view that can be selected. Because we are able to control our dispatch count inside the shader, we can then cap this resolution dynamically depending on the current view of the instance. It is clear then that dispatching tasks for indices above the maximum that is selected is futile, so this maximum index is the value we wish to estimate, and set the indirect dispatch count to.

We say the *maximum requested index* for a cluster, based on the error values calculated in Alg. 4, is:

$$\text{max-idx}(i) = \begin{cases} \text{max-parent}(i) & \text{if parent-err} < \tau \\ \text{max-child}(i) & \text{if this-err} > \tau \\ i & \text{else} \end{cases} \quad (2)$$

Intuitively, bringing an instance closer to view yields a greater error for clusters, which may require replacing a cluster with its children, so we increase max-idx and the tasks invoked for the instance. Inversely, moving an instance away from view will reduce its tasks invoked. A cluster only views local data, so does not request drawing clusters beyond the scope of its parents or children, so we set our next dispatch count to the maximum requested indices of all clusters. This brings with it a single frame of latency to apply the requested index if it increases, so we need some small additional logic when selecting clusters in case the resolution we wish to draw at is not available. Simply, if our cluster has too high an error to draw, but our

children are out of range of current workgroups (and so are not being processed), we should draw ourselves anyway.

Once we have determined a cluster should be drawn, the task shader code is identical to the previous method; see §3.1.2.

3.3 Cluster Culling

An engine based around instances and LOD chains may utilize *instance culling* to save time in rasterisation. This can be improved; instance culling has some of the same flaws as LOD chains, being based on arbitrary-sized objects. If we instead focus on culling clusters, we end up doing work on much more uniformly sized items[5], which results in finer-grained culling [5]. This technique was used in industry before cluster based multiresolutions, as the GPU friendliness of clusters makes them ideal for GPU-driven rendering systems.

A simple culling technique we apply is frustum culling, not drawing a cluster if it is outside the camera frustum. The frustum can be represented by six planes, which we extract from the model-view-projection matrix as described by [4]. These planes will then exist in object-space, and, from error calculations, each cluster contains a bounding sphere in object-space. We then cull clusters if their bounds are on the negative side of any plane.

This requires testing every cluster that may be drawn; DAG explore can be optimised further. The DAG is a nested bounding hierarchy, so a bound of a cluster contains the bounds of all children. A successful cull check on a cluster's bound would then rule out the entire hierarchy of clusters descending from it. DAG Explore can then stop exploration early if a cluster can be culled, as we then know no child may be rendered, culling as early as possible. At the coarsest grain, a successful cull on the root cluster is equivalent to instance culling.

4 Evaluation

We evaluate on two benchmarks on a GTX 1660 and r5 3600. The first has almost optimal conditions for LOD chains (*LOD efficient*), and the next demonstrates their primary limitation (*LOD deficient*).

Our LOD efficient benchmark moves a camera back from the scene origin, revealing a large 2D grid of instances. The benchmark uses the Stanford Dragon model [11], which contains 1 million source triangles, meaning our scene of 1000 instances contains 1 billion triangles. These instances will occupy a narrow slice of depth on the screen, so are suitable for traditional LOD rendering. Frame times are plotted in Figure 9.

[5]Uniformly sized in screen-space if done after LOD generation, making this take time proportional to target error.

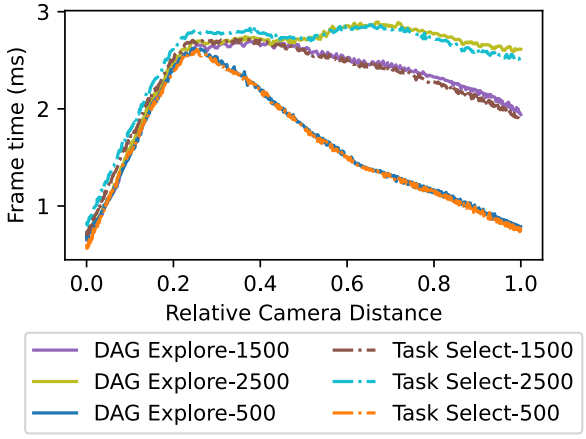


Figure 9: LOD efficient: Results combining benchmarks for 500 up to 2500 instances in the scene.

Instance Count	500	1000	1500	2000	2500
LOD Chain	2.58	3.49	3.85	4.01	4.10
Task Select	1.41	1.88	2.09	2.21	2.27
DAG Explore	1.39	1.89	2.13	2.27	2.37
% Change	1.42	-0.53	-1.93	-2.79	-4.40

Table 1: LOD efficient: Mean frame times (ms) for the different methods across instance counts. Our CPU based LOD chain uses an error function equivalent to the multiresolutions.

Task select achieves very similar performance across the board to DAG Explore while saving intermediate memory. For context, the full resolution 2000 instance scene takes a median of 472ms to render. The 2500 instance scene uses 38.82MB of GPU memory to store the draw buffer for DAG Explore, but just 29.31KB of intermediate memory for Task Select's indirect dispatch parameter buffer.

At relative camera distances of less than 0.4, the benchmark's screen is filled with instances, with varying amounts of culling. In these ranges, we can see in Figure 9 that task select is a bit slower per instance. This is due to its more fine-grained culling doing more work to cull an entire instance, while the variable-grained culler DAG Explore culls as early as possible while searching the DAG, and has an almost negligible slowdown per instance.

The cost per instance of a CPU cull check and draw

Method	GPU Time (ms)	Profiler samples (% + ms) Task	Mesh/Vert	Frag
Task Select	1.28	66%	29%	5%
		0.85	0.37	0.064
LOD Chain	9.15		97.5%	2.5%
			8.92	0.23

Table 2: LOD deficient: Frame time analysis of a single frame for the viewpoint from Fig. 2, Lucy, using NVIDIA Nsight. Fragment stages for both methods are identical.

call is clearly visible for our traditional LOD chain in Table 1, so we would expect improvements from a GPU implementation. In contrast, Table 2 shows a large efficiency gain from Task Select over the LOD chain that results from their limitations; the instance is both close and far from the camera, but the LOD chain renders at full resolution.

Our method optimises more effectively given more instances, as more instances give us more fine-grained control over the indirectly dispatched clusters. This is, however, slightly contrary to the original problem, the rendering of small numbers of massive meshes. We expect DAG explore to perform better in these cases. However, our method is still competitive due to its low reliance on memory bandwidth; massive multiresolutions would require more working space for the queue than is commonly available as workgroup shared memory.

5 Conclusions

Cluster-based rendering engines already give way to GPU-driven pipelines that excel at high-fidelity scenes. In such a pipeline, the practicality of multiresolutions is clear. They are generated automatically, sampled based on concrete metrics, and can be integrated into existing workflows.

The methods demonstrated in this paper are all limited by the rate of rasterisation, which is held roughly constant within a scene. This means multiresolutions are likely to fit into the frame budget of a high-fidelity renderer. This is a key goal of the method; a good error function should keep the screen-space triangle density roughly constant. In doing so, we grant complete flexibility on the source resolutions of any mesh in any scene, a major advantage for renderers targeting photorealism.

This renderer still relies on having enough VRAM to store a massive multiresolution, something that cannot be taken for granted within a large engine. As such, future work includes data streaming, which would load segments of the multiresolution into memory only on demand [7].

Nanite is a monolithic pipeline, making the methods used for generating and rendering multiresolutions hard to extract for general use. This paper has instead presented viable generic algorithms for rendering in this new paradigm. In future, we hope to see these help push multiresolutions as a standard tool in contemporary engines.

References

[1] P. Cignoni, F. Ganovelli, E. Gobbetti, F. Marton, F. Ponchio, and R. Scopigno. Batched multi triangulation. *VIS 05. IEEE Visualization, 2005.*, 2005.

[2] Paolo Cignoni, Fabio Ganovelli, Enrico Gobbetti, Fabio Marton, Federico Ponchio, and Roberto Scopigno. Bdam—batched dynamic adaptive meshes for high performance terrain visualization. In *Computer Graphics Forum*, volume 22, pages 505–514. Wiley Online Library, 2003.

[3] Paolo Cignoni, Fabio Ganovelli, Enrico Gobbetti, Fabio Marton, Federico Ponchio, and Roberto Scopigno. Adaptive tetrapuzzles: efficient out-of-core construction and visualization of gigantic multiresolution polygonal models. *ACM Transactions on Graphics (TOG)*, 23(3):796–803, 2004.

[4] Gil Gribb and Klaus Hartmann. Fast extraction of viewing frustum planes from the world-view-projection matrix. *Online document*, 2001.

[5] Ulrich Haar and Sebastian Aaltonen. Gpu-driven rendering pipelines. *SIGGRAPH*, 2015. URL `https://advances.realtimerendering.com/s2015/index.html`.

[6] Hugues Hoppe. Progressive meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96. Association for Computing Machinery, 1996.

[7] Brian Karis, Rune Stubbe, and Graham Wihlida. A deep dive into nanite virtualized geometry. *SIGGRAPH*, 2021. URL `https://advances.realtimerendering.com/s2021/index.html`.

[8] Christoph Kubisch. Mesh shading for vulkan. *Khronos Blog*, 2022. URL `https://www.khronos.org/blog/mesh-shading-for-vulkan`.

[9] Amar Patel and Tex Riddell. D3d12 work graphs preview. *DirectX Developer Blog*, 2023.

[10] Federico Ponchio. Multiresolution structures for interactive visualization of very large 3d datasets. 2009.

[11] The Stanford 3D Scanning Repository. URL `http://graphics.stanford.edu/data/3Dscanrep/`.

[12] Sung-Eui Yoon, Brian Salomon, Russell Gayle, and Dinesh Manocha. Quick-vdr: Interactive view-dependent rendering of massive models. In *ACM SIGGRAPH 2004 Sketches*, page 22. 2004.

Posters

Automatic Mesh Generation for Realistic Human Avatars

Martin Halaj*
Dana Škorvánková†
Supervised by: Martin Madaras‡

Faculty of Mathematics, Physics and Informatics,
Comenius University
Bratislava, Slovakia

The demand for realistic human avatars has increased in various industries, including gaming, virtual and augmented reality, fashion, and healthcare. Therefore, it is crucial to accurately replicate anthropometric measurements on the body model of a human.

We create an accurate representation of the human body using anthropometric measurements. Our tool is able to take up to 16 input measurements of different parts of the body and generate shape parameters that are used to construct a body mesh using the SMPL model. To generate a training dataset, we used the SMPL Anthropometry tool developed by David Bojanić [1], which allowed us to measure the generated SMPL meshes. Our tool then takes a vector composed of measurements as input and produces 10 SMPL shape parameters that define the shape of the final body mesh. We researched the optimal model for deriving shape parameters and found that using linear or second-degree polynomial regression produces the most accurate results.

This approach was previously used in The Virtual Caliper [2]. Unlike our solution, it only uses up to 5 input measurements and is more resource-intensive. Another tool employing this approach is Meshcapade Me [1]. While Meshcapade Me offers better accuracy with up to 13 input measures, it is a paid commercial tool.

We created a dataset that includes images of standing humans. The dataset includes front and side images of humans with and without backgrounds, as well as joint locations in 3D space and meshes of corresponding avatars. We generated this dataset for 100,000 avatars, resulting in over half a million files. We adapted the SURREACT framework to create this dataset [3].

This dataset is designed to complement datasets composed of real data. Typically, datasets like this do not contain a large amount of data due to various reasons, such as the complexity of capturing numerous people or concerns regarding the privacy of individuals involved. In our work, we aim to extend these datasets with synthetic data for the purpose of training CNNs capable of extracting anthropometric measures from images.

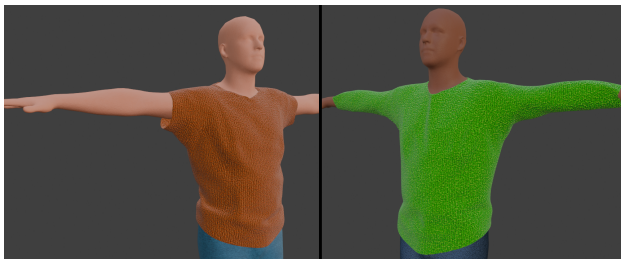


Figure 1: Example of clothed avatars.

Figure 1 shows two avatars wearing shirts generated by our tool we are developing. The tool adds procedurally generated skin to avatars and allows for adjusting skin tone to promote diversity in dataset. Additionally, it can generate some clothing items and add texture to enhance the model's realism.

In summary, our proposed framework addresses the challenge of automatic mesh generation based on provided anthropometric measurements and provides training data for CNNs focused on extracting measurements from images.

Keywords: Human Body, Synthetic Data, SMPL

References

[1] David Bojanić. SMPL Anthropometry. https://github.com/DavidBoja/SMPL-Anthropometry, 2023. Last accessed 23 February 2024.

[2] Sergi Pujades, Betty Mohler, Anne Thaler, Joachim Tesch, Naureen Mahmood, Nikolas Hesse, Heinrich H Bülthoff, and Michael J Black. The virtual caliper: Rapid creation of metrically accurate avatars from 3d measurements. *IEEE transactions on visualization and computer graphics*, 25(5):1887–1897, 2019.

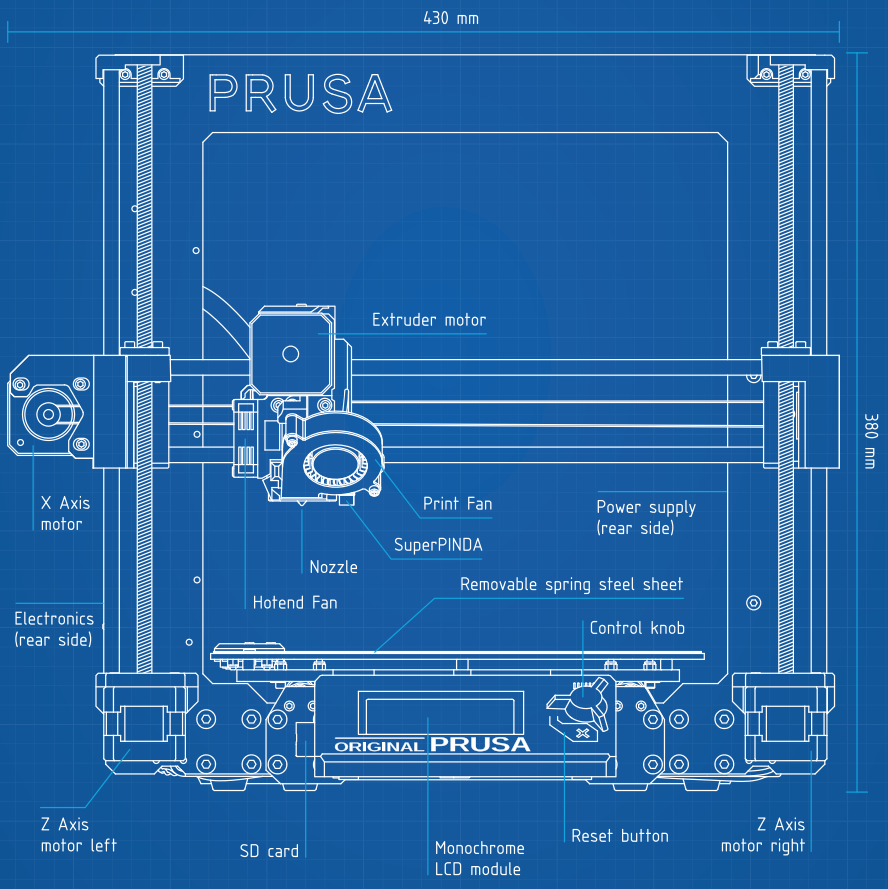
[3] Gül Varol, Ivan Laptev, Cordelia Schmid, and Andrew Zisserman. Synthetic humans for action recognition from unseen viewpoints. In *IJCV*, 2021.

*halaj21@uniba.sk
†dana.skorvankova@fmph.uniba.sk
‡martin.madaras@fmph.uniba.sk
[1] https://me.meshcapade.com/

Partners of CESCG 2024

CHANGE THE WORLD WITH 3D PRINTING

BE PART OF THE DEVELOPMENT OF THE WORLD'S BEST 3D PRINTERS!



PRUSA3D.COM

PRUSA RESEARCH
by JOSEF PRUSA

WHO ARE WE LOOKING FOR?

Working with us is a truly remarkable adventure. We develop, invent, and test... We have no limits on what is possible and what is not. We have only one particular goal: **to make the best 3D printers in the world!**

That's why we're looking for **skilled software and hardware developers** to join our development department. Interested?
Take a look at our development teams that could use a hand:

FIRMWARE

WE HAVE SEVERAL DEDICATED TEAMS. THESE TEAMS DEVELOP SPECIALIZED FIRMWARE FOR OUR FDM AND SLA PRINTERS.

C++ AND PYTHON

PRUSASLICER

WE DEVELOP OUR OWN SLICING SOFTWARE FOR PRINTING DATA PREPARATION AND WORKING WITH 3D OBJECTS.

C++

PRUSA CONNECT

WE ARE DEVELOPING A TOOL TO CONTROL THE ENTIRE 3D PRINTING ECOSYSTEM REMOTELY.

C++ AND PYTHON

WEB DEVELOPMENT

WE HAVE SEVERAL WEBSITES, ALL DEVELOPED IN-HOUSE.

PHP, PYTHON, DEVOPS (K8S), JAVASCRIPT (REACT, ANGULAR)

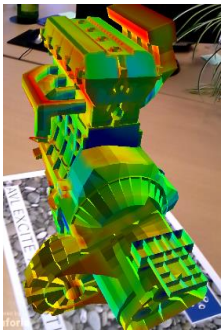
WHAT TOOLS DO WE USE?

We use **GitHub** for software development and internal projects, **JIRA** for progress tracking, and Confluence for documenting work. We don't send dozens of emails to each other, we communicate through **Slack**. Our company also offers a well-equipped lab and workshop with a large variety of tools: oscilloscopes, spectrometers, lasers, CNC, etc. You can also use revolutionary and exciting technologies such as robotic arms or a cybernetic dog from Boston Dynamics. It's up to you what you can do with them.

VISIT OUR WEBSITE FOR CURRENT OPEN JOB OPPORTUNITIES

WWW.PRUSA3D.COM

zentrum für
virtual reality und visualisierung
forschungs-gmbh



VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH

The VRVis Research Center is a joint venture in research and development for virtual reality and visualization. VRVis was founded in 2000 as part of the Austrian Kplus program to bridge the gap between academic research and commercial development as well as to supply the necessary transfer of knowledge between the academic community and industry. The competence center VRVis is funded by BMVIT, BMDW, the Vienna Business Agency, Styria and the Styrian Business Promotion Agency (SFG) within the scope of COMET – Competence Centers for Excellent Technologies. The program COMET is managed by FFG.
The company is located in Vienna, Austria. Today, around 70 researchers together with about 20 students do high-level applied and basic research in four different areas.

The Team

VRVis consists of internationally experienced researchers in the areas Visual Analytics, Complex Systems, Smart Worlds and Multiple Senses. Their outstanding experience and knowledge in these topics qualify them for the innovative research they are performing. The research areas are headed by key researchers who manage these areas, define goals and projects for this area, as well as conduct the defined research together with their staff. Most members of the research teams are young researchers, whose creativity and ingenuity is the key to the success. Beyond that VRVis has a friendly and inclusive company culture, which translates into great teamwork and –spirit, also outside of the office (e.g. our running teams).

Research Program

The scientific research program consists of the previously mentioned research areas in which thematically matching projects are conducted. Each research area realizes application projects on the one hand and basic research for these application projects on the other hand.

Working at VRVis

VRVis is always looking for students, junior and senior researchers who want to join the team. VRVis is offering regular positions as well as internships, diploma and PhD theses in cooperation with universities. For more detailed information or currently open positions visit our website at www.vrvis.at.

Selection of Partners

Scientific Partners:

- Vienna University of Technology
- Graz University of Technology
- University of Vienna

Industrial Partners:

- AVL List GmbH
- AGFA Healthcare GesmbH
- Austria Power Grid AG
- Geodata Ziviltechniker GmbH
- HILTI Corporation
- ÖBB-Infrastruktur AG
- RHI Feuerfest GmbH
- Zumtobel Lighting GmbH
- and many more

Currently, VRVis is again extending its industrial base with new partners from several new fields.

Additional Information and Contact

Please visit our website for detailed information about the research program or current projects at www.vrvis.at or contact us at office@vrvis.at or via phone +43 (1) 908 98 92.