Fakultät für Mathematik und Informatik
Institut für Informatik
Mensch-Computer-Interaktion

Masterarbeit

zur Erlangung des Grades
Master of Science (M.Sc.)
im Studiengang Informatik
an der Universität Würzburg

Experimental Investigation of Haptic and Pseudo-Haptic Feedback on 3D Media in Virtual Reality

vorgelegt von
Vanessa Pfeiffer
Matrikelnummer: 2253163

am September 23, 2024

Prüfer:
Prof. Dr. Sebastian von Mammen, Lehrstuhl Informatik IX
Prof. Dr. Birgit Lugrin, Lehrstuhl Informatik V

Betreuer:
Daniel Pohl, CEO immerVR GmbH

Zusammenfassung

Virtual Reality (VR) ermöglicht es den Nutzern, Bilder auf eine immersivere Weise zu erleben, insbesondere mit immersiven Formaten wie Panoramen (180° and 360°) und stereoskopische Bilder. Die derzeitigen Anwendungen ermöglichen die visuelle Erkundung dieser immersiven Inhalte, aber es fehlt an haptischen und interaktiven Elementen, die die Immersion erheblich verstärken könnten. In dieser Arbeit wird vorgeschlagen, haptisches und pseudohaptisches Feedback in die VR-Bildbetrachtung zu integrieren, um das Gefühl der Präsenz, ein Schlüsselindikator für die Immersion in VR, zu verstärken. Zu diesem Zweck werden aus panorma Bildern mit Hilfe von maschinellen Lernen Tiefenkarten generiert und diese zu 3D-Darstellungen der Bildes verarbeitet. Dieses 3D Mesh ermöglicht es dem Benutzer, mit einer virtuellen Hand mit dem Bild zu interagieren und haptisches und pseudohaptisches Feedback auf der Grundlage der berührten Oberfläche zu erfahren. Das haptische Feedback wird in Form von Vibrationen des Controllers implementiert, das pseudohaptische Feedback in Form von Tönen und visuellen Effekten. In einer Studie werden die Auswirkungen der 3D-Darstellung und der Feedbacktechniken auf das Präsenzgefühl untersucht. Die Ergebnisse zeigen, dass das kombinierte haptische und pseudohaptische Feedback das Gefühl der Präsenz beim Betrachten von VR-Bildern deutlich verbessern. Während das haptische Feedback allein nur eine begrenzte Wirkung hat, steigert seine Kombination mit Tönen und visuellem Feedback das Gefühl der Präsenz erheblich. Die Studie zeigt auch, dass Bilder, die in der 3D-Mesh-Darstellung präsentiert werden, ein stärkeres Gefühl der Präsenz vermitteln als monoskopische Formate, aber stereoskopische Formate sind nach wie vor am immersivsten.

Abstract

Virtual Reality (VR) allows users to experience images in a more immersive way, especially in immersive formats such as spherical (180° and 360°) and stereoscopic images. Current applications allow for visually exploring this immersive content but lack tactile and interactive elements that could significantly amplify immersion. This thesis proposes integrating haptic and pseudo-haptic feedback into the VR image-viewing experience to increase the sense of presence, a key indicator of immersion in VR. To this end, depth maps are generated from spherical images using machine learning models and processed into a 3D mesh representation of the image. Six degrees of freedom (DoF) tracked controllers, represented inside the virtual environment as hands, can collide with the surfaces of the image mesh and trigger haptic and pseudo-haptic feedback. Haptic feedback is implemented as controller vibrations, and pseudo-haptic feedback as audio cues and visual effects. A study is used to evaluate the impact of the 3D mesh and feedback techniques on the sense of presence. The results demonstrate that the combined haptic and pseudo-haptic feedback significantly enhance the feeling of presence in VR image viewing. While haptic feedback alone has a limited impact, its combination with audio cues and visual feedback substantially elevates the sense of presence. The study also shows that images presented in the 3D mesh representation offer a higher sense of presence than monoscopic images, but stereoscopic formats remain the most immersive.

Contents

List of Figures

List of Tables

1 Introduction

The Virtual Reality (VR) and Augmented Reality(AR) headset market is estimated to grow 35.6 % each year until 2032 [1]. The user base of these devices is showing a growing interest in diverse and more immersive VR applications, including services they already use on their smartphones but re-imagined for VR. This includes image viewing, which offers a unique opportunity to enhance the experience of exploring visual content. Stereo images deliver a convincing 3D effect when viewed through a VR headset. Panoramas immerse the user by wrapping around the viewpoint. And spherical (180° or 360°) images envelop the viewer in a virtual sphere and provide an unparalleled sense of presence. However, current VR image-viewing experiences remain primarily visual. Some integrate static effects, like background audio or particle systems, but lack interactive elements. Dynamic visual, audio, and haptic feedback could further amplify immersion by enabling viewers to explore media interactively.

This work investigates adding those interactive elements to VR image-viewing for spherical images. An existing VR image-viewing application is extended to allow the viewer to explore the image with a virtual hand and receive haptic and pseudo-haptic feedback relative to the surface they touch. To enable this interaction at the correct distance for the objects in the image, a 3D mesh representation is generated. The system uses machine learning depth estimation to create a depthmap of the image and processes it into a 3D virtual environment. In combination with a material map and surface properties, the appropriate vibration, audio cue, and visual effect are played in response to the user touching the image. Figure 1.1 visualizes the combination of pre-processing and real-time calculations that form this approach. A user study is conducted to determine the influence of the 3D mesh and the feedback techniques on the sense of presence.

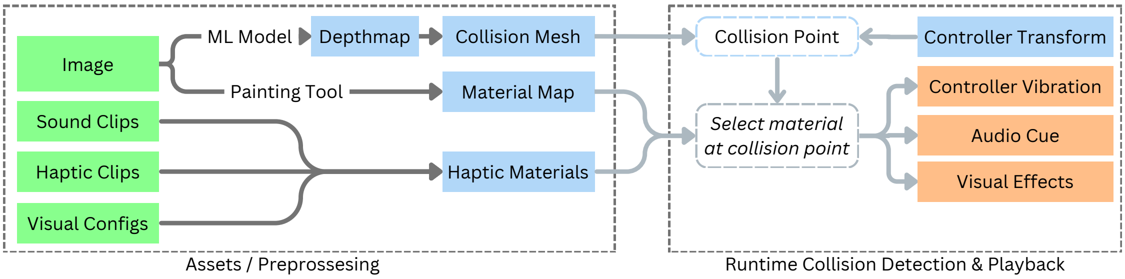


Figure 1.1. Using an image as input, a depth map is derived to create a collision mesh. A material map partitions the image into haptic materials. They contain sound and haptic clips correlating to the surface they represent. When a VR controller triggers a collision, the respective haptic material is selected to generate controller vibration, play back the material's associated sound clip, and display visual effects on the image.

The contributions of this thesis are the following:

- It introduces a process to enable exploring images with your hands in VR through haptic and pseudo-haptic feedback.
- Errors caused by the distortion of the equirectangular projection in state-of-the-art spherical monocular depthmap estimation models are visualized.
- A technique is presented for creating 3D representations from spherical images while working around this equirectangular projection error.
- The influence of haptic and pseudo-haptic feedback techniques on the sense of presence in the context of spherical images is measured.
- 3D mesh representation is shown to create more presence than the monoscopic images but less presence than the stereoscopic images.

The remainder of this work is organized as follows. Chapter 2 presents background information and summarizes related work for image viewing in VR, haptic and pseudo-haptic feedback, and different depthmap estimation methods for spherical images. Chapter 3 describes the methodology for the implementation and user study. After comparing different depthmap generation models, the process of constructing the 3D mesh representation is described. Then, the implementation of each feedback technique and the collision detection is laid out. Lastly, the study configurations for the haptic and pseudo-haptic feedback evaluation and the 3D representation investigation are described. Chapter 4 presents and analyzes the study results. Finally, chapter 5 concludes this thesis and discusses potential future works.

2 Background and Related Work

This chapter gives relevant background information by presenting current image-viewing applications for VR, introducing the concept of presence, and discussing haptic and pseudo-haptic feedback methods. Then, approaches for known depthmap generation, including techniques for spherical images, are presented.

2.1 Image Viewing in VR

The current landscape of VR image-viewing applications showcases a variety of approaches to enhance the user experience. For instance, the 'VR Photo Slideshow' [2] app in the Meta Horizon Store prioritizes customization, allowing users to control the number and placement of virtual screens within their environment. In contrast, the Steam platform offers viewers like 'VR Photo Viewer' [3] and 'Witoo VR photo viewer' [4], which emphasize compatibility with diverse image formats, including 360° panoramas and 3D photos. The 'Virtual Home Theater VR Video Player' [5] app on Steam further expands the scope of VR image viewing by simulating a cinematic experience for both 2D and 3D movies, complete with customizable audio settings. The app 'immerGallery' [6] for Meta Quest devices is made by immerVR [7], a German Virtual Reality startup around immersive media. It supports the most extensive range of image types, from panoramas to spherical images in mono and stereo formats. Furthermore, it enhances the viewing experience with particle effect systems, background music, and voice-over.

Despite these advancements, the current state of VR image-viewing apps reveals a considerable opportunity for innovation. The integration of sophisticated haptic feedback mechanisms, advanced 3D representations, and interactive features remains largely unexplored.

2.2 Presence

Presence is a key concept used to determine the effectiveness of a VR experience. In general, presence is described as feeling present inside a computer-based environment [8], [9]. In literature, presence is subdivided into two categories. Social presence refers to the sensation of being included in a group of virtual agents [10]. Spacial presence refers to the feeling of being physically located in the virtual environment. Dinh *et al.* [11] show that adding tactile, olfactory, and auditory cues can increase the sense of presence. A concept often used in conjunction with presence but not as clearly defined is immersion. Some researchers, like Slater *et al.* [12], define immersion as a quantifiable technological aspect of a system to which it is able to generate an illusion of reality. Others, like Witmer *et al.* [13], describe it as the experience of being included in an environment and forgetting the real world around oneself. Slater *et al.* [12] have shown that increased immersion does not necessarily lead to an increased presence.

Questionnaires can be used to measure the presence inside virtual environments. The most prominent presence questionnaires are created by Slater *et al.* [14] and Witmer *et al.* [13]. The Igroup Presence Questionnaire (IPQ) [15] subdivides its prompts into multiple presence measurement categories. It measures general presence, involvement, experienced realism, and spatial presence. Next to phrasing their own prompts, the igroup integrates prompts from multiple verified presence questionnaires into the IPQ. This includes prompts from the aforementioned works as well as prompts from Hendrix [16] and Carlin *et al.* [17].

2.3 Haptic Feedback in VR

In Virtual Reality, haptic feedback is used to inform users about tactile sensations when interacting with a 3D environment. This feedback deepens the immersion and realism the user experiences in virtual experiences.

2.3.1 Haptic Feedback in Current Commercial VR Systems

With the basis of VR controllers still using inertia-based vibration motors, the haptic feedback in commercial consumer VR hardware has not changed much since the introduction of the Oculus Touch controllers for the Oculus Rift CV1. It uses Linear Resonant Actuators (LRA) [18] to provide the vibration feedback. These can simulate different interactions by varying the strength and amplitude of vibrations. The controllers have a latency of 33 ms when responding to vibration changes. This causes a lack of nuance to convey more detailed feedback. To combat this, Oculus introduced buffered feedback, which allows for faster response times and more nuanced vibration waveforms. Later VR systems also improved their haptic feedback by increasing the detail the vibration waveform could represent. Newer haptic motors can achieve lower latency and a higher range of frequencies. The Valve Index Controllers do this by using high-definition LRAs [19]. This allows them to imitate a broader range of sensations. They also add individual finger tracking as a feature that could be used to add pseudo-haptic feedback to applications.

2.3.2 Haptic Feedback in VR for Media Content

Haptic feedback for media is a relatively unexplored area. Most research in haptic feedback for VR has been done for 3D objects. At this time, there is only one project that adds haptic feedback to photos and videos in VR. Touchly [20] is an application for the Meta Quest and SteamVR devices that provides haptic feedback based on a simulated hand collision with video content. It uses machine learning to create a depthmap of the media, then uses that map for the hand collision and vibration feedback. Detailed information about the application's internal workings is not available as it is a closed-source project, and the developers have not published any papers on the subject.

2.4 Pseudo-Haptic Feedback

Pseudo-haptic feedback refers to a sensory illusion in which users perceive haptic sensations without the use of direct physical forces or actuators. The illusion is created by manipulating the user's sensory perceptions through additional visual and auditory cues. This is possible thanks to the principle of sensory integration, the concept that the brain combines multiple sensory inputs to the objective perception. By feeding the brain certain inputs, the illusion of texture, weight, resistance, or motion can be created [21] [22].

Pseudo-haptic feedback can be created using various methods targeting different sensory inputs [23]. The most common techniques involve:

- Deformation: The touched object is deformed to match the reaction the user would expect with a real object. This can simulate a feeling of pressure and weight and give the user the feeling they are manipulating a real object. Sato *et al.* [24] have shown that this can create the sensation of softness.
- Translation and simulated inertia and momentum: When the user touches the object, it receives an appropriate amount of force. The force coupled with a physics engine with inertia and momentum replicates the movement the user would expect with a real object. It supports the user's feeling of force and weight.
- Texture changes: Using visual patterns around or behind the area the user interacts with can simulate the feel of different textures.
- Auditory cues: Adding auditory cues to visual feedback can positively influence the haptic illusion. Different materials create different sounds when interacted with. Synchronizing the user's interaction with fitting audio cues can suggest information about the surface, such as texture, hardness, and other qualities. By using 3D audio techniques, this effect can be strengthened. According to Hosoi *et al.* [25], auditory cues can also create pseudo-haptic sensations of the environment, such as wind. Kaneko *et al.* [26] have shown that changing the delay, frequency, and loudness of auditory feedback can even influence the heaviness sensation a user experiences. This is supported by Canadas-Quesada *et al.* [27], who show that people associate shorter auditory feedback with a more stiff haptic perception.
- Electrical muscle stimulation (EMS): A more novel approach to pseudo-haptic feedback is using EMS devices placed on the muscle of the arm. Kim *et al.* [28] have used this to simulate the muscle effort when lifting an object. This has been shown by Rietzler *et al.* [29] to successfully induce weight perception when combined with other pseudo-haptic feedback types.
- Virtual hand: The rubber hand experiment by Botvinick *et al.* [30] has shown that the visualizations of limbs can elicit a sense of ownership. By matching the sensations felt by the body with the observations of a fake limb, the brain can be tricked into believing the limb belongs to the body. Similarly, a virtual hand that follows the movement of a VR controller can elicit a sense of ownership.

In this context, a more realistic hand can evoke a stronger sense of ownership, but realism has no influence on the sense of agency [31].

- Control-display (C/D) ratio manipulation: This technique involves creating an offset between the user's hand (control) and its virtual representation (display) when interacting (lifting, moving) with virtual objects [22], [32], [33]. Through this offset, the user experiences a feeling of weight. It has been shown by Moosavi *et al.* [22] to be effective at creating object lifting behavior similar to real object lifting. The perceived weight through C/D ratio manipulation depends on the size of the virtual objects. Larger objects require a lower C/D ratio for users to perceive any weight, while for smaller objects, a higher C/D ratio is enough [34].
- Shaking-Finger Effect: A similar technique to C/D ratio manipulation is the shaking-finger effect as described by Sato *et al.* [24]. This involves applying micro-movements to the fingers of a virtual hand while it is moving over a surface. The movements of the fingers can give the impression of touching an uneven surface. It gives a sense of roughness. Another technique described by Sato *et al.* is the increasing speed effect. Here, the speed of the virtual hand increases while the object is being touched.

2.5 Depthmap Generation

For this project, depthmaps are used to provide haptic feedback. Depthmaps are textures where each pixel holds a depth value, often an 8-bit or 16-bit value. In our use case, each pixel represents the depth of the corresponding pixel on the RGB image. The creation of depthmaps is a well-researched area. Depth estimation used to be most commonly done by matching stereo image pairs, but with the improvements in machine learning, recent techniques can estimate depth reasonably well, even from monoscopic images. Depth estimation for 180° and 360° is less researched. Some approaches project the spherical images into multiple planar images, run a standard depth estimation on each, and combine them into a spherical depthmap. The following sections will discuss the different methods of depth estimation.

2.5.1 Depth Estimation using Machine Learning

In recent years, many works have used machine learning to estimate depthmaps from single 2D input images. Eigen *et al.* [35] use a two-component architecture that combines global and local views to predict the depth. They use two convolutional neural networks (CNNs, [36], [37]), one to predict the global depth and a second to refine the depth locally. The second network takes the first output and the original image as input to refine the depth around object boundaries and wall edges. One year later, Eigen *et al.* [38] make several improvements to their architecture. Most prominently, they add more convolutional layers to each CNN and add a third CNN. This increases the output resolution and outperforms existing methods on the vast majority of benchmarks.

While Laina *et al.* [39] also use a CNN to predict depth, their approach uses a single CNN that follows residual learning. Thanks to being fully convolutional, they are able to greatly reduce the number of parameters and, thus, the number of training samples required. This approach is faster and more efficient than using multiple CNNs and even runs in real-time on videos.

One issue with these approaches is that they require a lot of labeled training data. To work around this, Garg *et al.* [40] present an approach using unsupervised learning. Instead of a depth ground truth, they use two images with a known camera movement between them, such as stereoscopic images. Using an Encoder-Decoder Architecture, a deep CNN predicts the depth of one image. Using the predicted depth and the known camera movement, the decoder generates a warped image. This is matched with the encoder input to construct a simple loss. They achieve a single-view depth estimation comparable to other state-of-the-art methods. Similarly, Godard *et al.* [41] train a CNN on reconstruction loss. Their approach differs in using bilinear sampling to achieve fully differentiable loss. It includes a left-right consistency check included in the network to improve result quality. Through this, they achieve a better-trained model, which beats the previous method in quality measurements.

Unsupervised learning based on stereo images has the drawback of "impose priors on the depth such as small depth gradient norms which may not be fully satisfied in the real environment" [42]. To take advantage of the easy data acquisition of unsupervised learning and the accuracy of supervised learning, they propose a semi-supervised approach. Their architecture uses a stereo image and depth data from LIDAR measurements as learning data. While LIDAR data is typically sparser than image resolution, combining it with the image alignment complements the LIDAR ground truth. They archive better performance metrics than both unsupervised methods across the board.

These deep learning methods can estimate accurate 2D depthmaps. However, when projected into a 3D mesh, the details around objects and edges are lacking. To improve this Li *et al.* [43] propose a two-streamed CNN. Instead of only estimating depth, their architecture splits the processing into two separate streams. One for the prediction of a depthmap and one for the prediction of a map of depth gradients. Both streams use two CNN layers and two fully connected layers. The output of both streams is fused together with the initial input to one highly detailed depthmap. Their use of a set loss over multiple images prevents overfitting and improves accuracy. With this, their architecture is competitive with other methods while getting more accurate and detailed results for 3D projections.

Liu *et al.* [44] propose a depth prediction approach using a Deep Convolutional Neural Field (DCNF) with Fully Convolutional Networks and Superpixel Pooling (FCSP). This technique brings an order of magnitude training speedup, which enables the use of deeper networks. Their results outperform previous methods, while their optimizations allow for larger input resolutions.

Instead of using a CNN, Islam *et al.* [45] are using a Generative Adversarial Network (GAN) to estimate depthmaps from single RGB images. Their architecture is composed of a fine-tuned generator and a global discriminator. The encoder takes the input image and depthmap and transforms them into their corresponding latent representations. Afterward, it translates each into a depthmap. The discriminator uses the fake and real depthmap to guide the generator in generating realistic outputs. The resulting model is able to estimate depth robustly in highly dynamic environments. Their results show that they match or slightly outperform other approaches.

Monocular depth estimation models not trained on 360° datasets produce suboptimal results for spherical images. However, training sets with ground truths for supervised learning are rare for omnidirectional media. To circumvent this challenge, Zioulis *et al.* [46] generate a new training data set with ground truth depth annotations using 3D datasets. They render 360° images together with ground truth depth maps from 3D environments of real-world captures and synthetic environments. With that dataset, they train two fully convolutional encoder-decoder networks. The first, UResNet, resembles other CNNS from literature, and the second, RectNet, was designed to handle better the distortions caused by the quirectangular format. Both networks beat models that were not trained on spherical images. RectNet makes more accurate predictions, but UResNet's predictions are smoother.

The biggest challenge in depth estimation for monocular 360° images is the distortion caused by the equirectangular projection. To counter this, Wang *et al.* [47] propose a network using two projections. In addition to the equirectangular image, the network uses a cube map projection as input. The equirectangular projection provides a wide field of view, giving access to all surrounding information, while the cube map projection provides smaller but non-distorted views. In their approach, they use spherical padding on the cube map to reduce boundary inconsistency on the cube faces. Each projection is fed into a separate model branch, and the features of each branch are shared after each layer using a bi-projection fusion procedure with learnable masks. This approach performs better than OmniDepth [46] in terms of qualitative results.

Similarly, Jiang *et al.* [48] utilize a fusion of a cube map projection branch and an equirectangular projection branch in their architecture. However, instead of using two branches for the encoding and decoding stage and bi-directionally feeding the features of each branch into the other, their approach uses two branches for the encoding but only one branch for the decoding stage. During the encoding stage, the features of both branches are fused, but the fused features are fed to the decoding stage. This reduces the complexity of the architecture and increases generalizability. UniFuse outperforms BiFuse [47] and OmiDepth [46] in terms of error metrics on four popular datasets.

Instead of designing and training specialized models for 360° images, Rey-Area *et al.* [49] propose an architecture that can utilize any state-of-the-art monocular depth estimator. In the 360MonoDepth framework, 360° images get projected into a set of overlapping perspective tangent images. Then, a state-of-the-art depth estimation model predicts depthmaps for each tangent image. After projecting the tangent depthmaps back into the equirectangular projection, a global optimization adjusts the scale and shift of the depthmaps to align them. Finally, the aligned depthmaps are merged into one 360° map using Poisson blending. Using MiDaS v2 [50], it matches the performance of UniFuse [48], but it can fail if the tangent maps are incorrect, e.g. for large plain walls. The advantage of this architecture is that it improves over time as it can easily be upgraded with any advancements in planar monocular depth estimation models.

Peng *et al.* [51] approach the depth estimation problem for spherical images similarly in that that their approach also projects the panoramic image into several perspective views, passes them into an existing monocular depth estimation method such as LeReS [52], and then stitches them back together. However, their approach uses an equirectangular reference depthmap generated using a panorama-based method such as UniFuse [48] to align the scale and shifts of the view's depth values. On top of that, they use a Laplacian-based Poisson blending to remove visible seams between the merged views. They are able to outperform other stitching-based panoramic depth estimation methods, such as 360MonoDepth [49] while being much faster.

2.5.2 Depth Estimation using Stereo Matching

Stereo matching is the process of finding pixels in the stereoscopic views that correspond to the same 3D point in the scene. This allows for the calculation of the disparity, the distance between the matched pixels in both views. Finally, the disparity, together with the camera lens specs, can be converted into a depth value using Equation 2.1 [53]. f_c is the focal length in pixels, which is the distance between the camera lens and the image sensor and varies depending on the camera model and resolution. ΔV is the distance between the two views in millimeters, and ΔP is the disparity, the distance between matched pixels. The equation returns the depth in millimeters of the object at the corresponding pixels.

$$\text{depth} = \text{f}_c \times \frac{\Delta V}{\Delta P} \tag{2.1}$$

Stereo-matching can be done using different computer vision or machine learning techniques. One of the simplest approaches is block-matching along horizontal lines. For this, it is assumed that the corresponding pixels are at the same vertical height in both views. Each horizontal scan line is checked for corresponding pixels. The correspondence is determined by comparing the area around the pixel. This method requires calibration to adjust for lens distortion and ensure alignment of the scan lines.

3 Methodology

For this thesis, the immersive media viewer 'immerGallery' [6] is being extended to support interactive haptic, auditory, and visual pseudo-haptic feedback. This chapter introduces the methodology for implementing this extension and the design of the study to evaluate its effectiveness. First, the creation of depthmaps and the construction of 3D representations of images is explained. Then, the implementation of each feedback mechanism and how collision detection triggers their playback are explained. Lastly, the user study design is laid out.

The depth estimation with machine learning, image pre-processing, and meta-file creation are implemented in Python 3.11.7, while the VR application is implemented using the Unity Engine 2022.3.4f1 [54] and the C# programming language.

3.1 Depthmap Generation and 3D Mesh Construction

To enable interaction with the objects inside an image at appropriate distances to the viewer, a 3D mesh is constructed using depthmaps. This section goes into detail about the machine learning models used to generate the depthmap, how it needs to be adjusted for spherical images, and how the final mesh is constructed and displayed using dynamic tessellation.

3.1.1 Depthmap Precprocessing

The textures, including depthmaps, are not included in the build of the application but are loaded from an external folder. Unity limitations force at runtime imported textures to use at most 8-bit data in each channel. To load 16-bit depthmaps, we encode the 16-bit data into the red and blue channels of an RGB texture. A Python script iterates over all depthmaps, converts their 16-bit data into a 24-bit RGB image, and saves it to a new file path. Afterward, it adds the new file paths to the corresponding image's meta-file.

3.1.2 Machine Learning Model Comparison

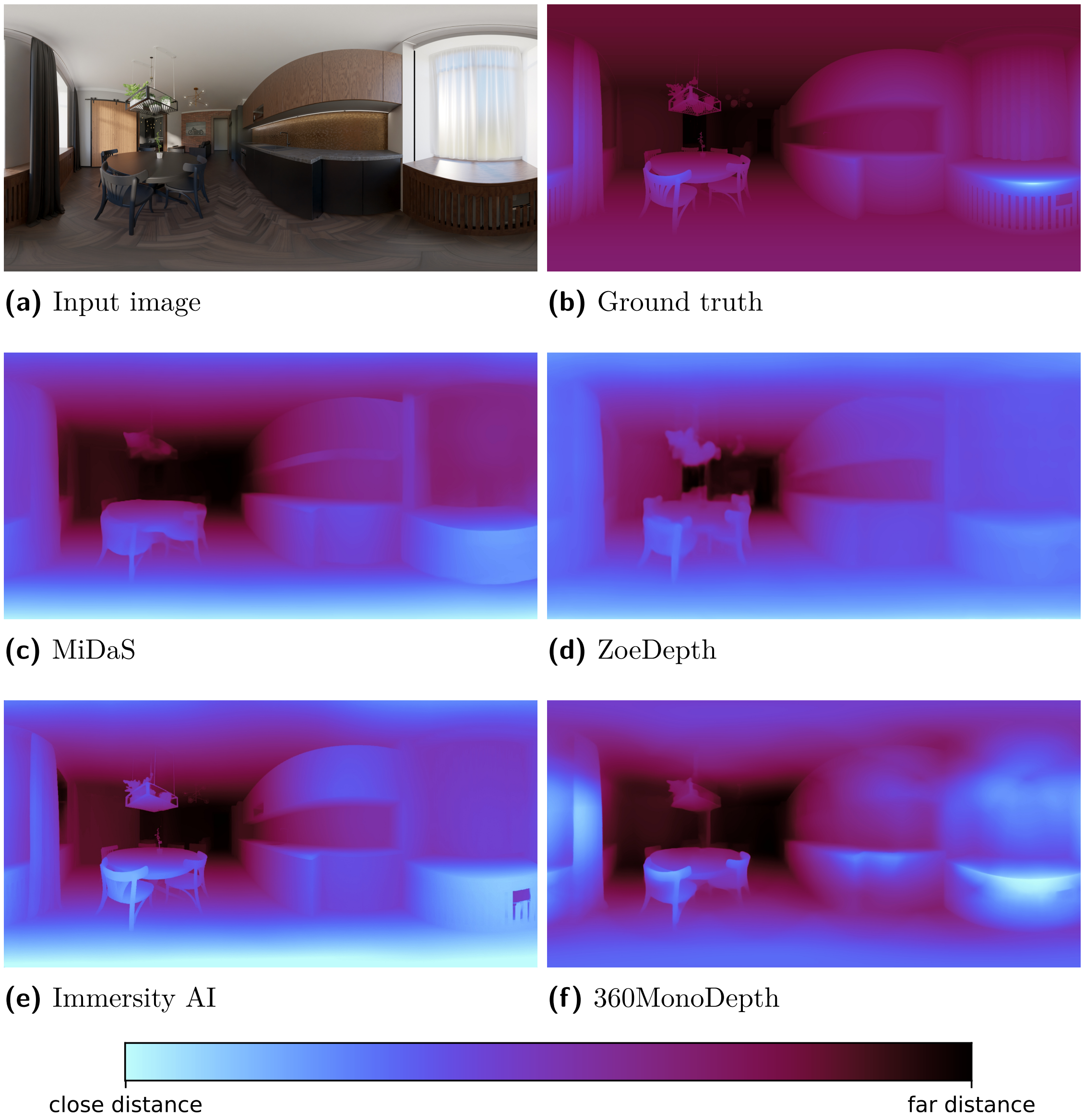
The selection of machine learning models for depthmap generation was guided by a multi-faceted evaluation that considered both the specific requirements of this research and the broader landscape of available techniques. The models chosen for in-depth comparison, MiDaS [50], ZoeDepth [55], 360MonoDepth [49], and Immersity AI [56], represent a diverse range of methodological approaches and performance characteristics.

Immersity AI was selected due to its promising performance in preliminary qualitative assessments, suggesting its potential for generating high-fidelity depthmaps suitable for creating immersive 3D representations. However, it is essential to acknowledge the limitations associated with its closed-source nature, which requires per-image payment, prevents reproducibility when the service changes its codebase and hinges on an active connection to its server.

MiDaS and ZoeDepth were included due to their open-source nature and efficiency, facilitating transparency and reproducibility in the research process. The installation using Python is easy and allows for automatizing the depthmap generation process. They take only a few seconds if executed on a GPU and produce outputs with reasonably accurate results. Additionally, their robust performance across various datasets indicates their potential for generalizability to the spherical image context.

360MonoDepth was explicitly designed to handle the unique geometric properties of spherical images, addressing a key challenge in this research domain.

By incorporating these diverse models, the work aims to comprehensively evaluate depthmap generation techniques, ensuring a balanced assessment of general-purpose and specialized approaches. The ultimate goal is to identify the model that best aligns with the specific requirements of this research, balancing accuracy, efficiency, and applicability to spherical images while acknowledging the inherent trade-offs associated with each model's characteristics.


(a) Input image

(b) Ground truth

(c) MiDaS

(d) ZoeDepth

(e) Immersity AI

(f) 360MonoDepth

Figure 3.1. Example outputs of different machine learning depth estimation models. Each prompted with the sample computer-generated image. Black/dark = far distance, blue/bright = near distance. (a) input image. (b) ground truth. (c) MiDaS depth estimation. (d) ZoeDepth depth estimation. (e) Immersity AI depth estimation. (f) 360MonoDepth depth estimation. None of the depth estimations reach the detail of the ground truth model. ZoeDepth has more details than MiDaS but has a lower resolution output and is, as such, more blurry. Immersity AI has the sharpest edges and most detail on objects.

The models are objectively compared to each other in their ability to estimate depth for spherical 360° images using computer-generated images with accurate ground truth. The input images and depthmap ground truth are rendered using Blender [57] with the Cycles ray tracing engine. One indoor and one outdoor scene is rendered from multiple camera angles to create the testing set. This ensures that the test images were not in any of the training data sets of the models. Furthermore, it allows for comparison with an objectively correct ground truth. One possible drawback of this approach is that the test images might not be perfectly photo-realistic in terms of noise and complexity.

Each model is prompted using the generated testing set to generate depthmaps. Figure 3.1 shows an example of a testing input and the generated output. Figure 3.1a is the computer-generated input image. Figure 3.1b shows the computer-generated ground truth of the input. Figures 3.1c to 3.1f show the depth estimations by MiDas, ZoeDepth, Immersity AI, and 360MonoDepth. The ground truth and depthmap values are normalized between each image's minimum and maximum pixel values. This allows for a direct comparison between the different model outputs and the ground truth.

Table 3.1 shows the quantitative comparisons on the computer-generated dataset. It lists the absolute relative (AbsRel) error, the mean absolute error, the root mean square error (RSME), the log root mean squared (RSME-log), and the accuracy for each depth estimation model. Equation 3.1 lists the equations for the error and accuracy metrics. For error measurements, lower values are better; for accuracy measurements, higher values are better. Blue-highlighted text is the best value for the respective measurement, and bold text is the second-best. MiDaS has the lowest RSME and RSME-log and is second best regarding AbsRel error, $\delta < 1.25^2$, and $\delta < 1.25^3$ accuracy. 360MonoDepth has the lowest AbsRel error, the highest $\delta < 1.25^2$ and $\delta < 1.25^3$ accuracy, and comes second best regarding MAE and RMSE-log. Immersity AI has the lowest MAE, highest $\delta < 1.25$ accuracy, and comes second best in RSME and $\delta < 1.25^2$ accuracy. Overall, these three models are very close to each other in the results, each edging out the others by a margin in two or three measurements. Only ZoeDepth performs significantly worse, coming last in all but the $\delta < 1.25$ metric.

Table 3.1. Quantitative results on the computer-generated test dataset. Metrics are calculated using the Equations 3.1. Highlighting: **best**, **second best**.

Model	AbsRel	MAE	RMSE	RMSE-log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
MiDaS	**0.750**	0.214	0.260	0.174	0.234	**0.445**	**0.644**
ZoeDepth	0.988	0.244	0.295	0.202	**0.291**	0.372	0.468
360MonoDepth	0.720	**0.212**	0.264	**0.175**	0.233	0.510	0.704
Immersity AI	0.820	0.210	**0.262**	0.179	0.303	**0.445**	0.596

$$
\begin{aligned}
\text{Absolute relative error (AbsRel):} & \frac{1}{N}\sum_{i=1}^{N}\frac{|z_i - z_i^*|}{z_i^*} \\
\text{Mean absolute error (MAE):} & \frac{1}{N}\sum_{i=1}^{N}|z_i - z_i^*| \\
\text{RMSE:} & \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|z_i - z_i^*\|^2} \\
\text{RMSE (log):} & \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\log_{10} z + i - \log_{10} z_i^*\|^2} \\
\text{Accuracy } \delta < \tau\text{: \% of } z \text{ s.t.} & \delta = \max\left(\frac{z_i}{z_i^*}, \frac{z_i^*}{z_i}\right) < \tau
\end{aligned}
\tag{3.1}
$$

Figure 3.2 shows the cumulative distribution function (CDF) of the error between the estimated depth of each model and the ground truth. The x-axis represents the error values, and the y-axis is the percent of pixels with an error less than the curve value. The light blue line shows the CDF for the MidaS model, the dark blue line for the ZoeDepth model, the yellow line for the 360MonoDepth model, and the orange line for the Immersity AI model. 40 % of pixels of all models have an error of less than 0.15. After that, the errors of MiDaS, 360MonoDetph, and Immersity AI follow the same curve. 70 % of their pixels have an error of less than 0.3. ZoeDepth performs a little worse, having an error of more than 0.3 for 38 % of its pixels. These results show that the accuracy of the depth predictions is in a similar range.

Figure 3.3 visualizes the average error of each model as a heatmap. It shows where the error is most severe for each model in terms of UV coordinate. The color displays the error from dark blue to yellow in a range from 0 to 0.5. All models are most accurate in the center of the image. MiDaS has an error of up to 0.35 at the top and 0.5 at the bottom. Similarly, 360MonoDepth has an error of 0.4 at the top and up to 0.5 at the bottom. ZoeDepth is the opposite, having an error of 0.5 at the top

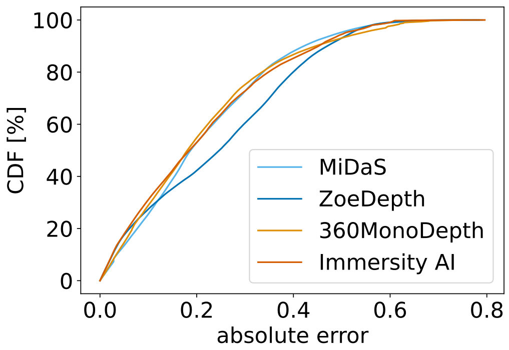


Figure 3.2. Cumulative distribution function (CDF) of depth estimation error for each model. Models were prompted with 12 computer-generated, equirectangular, 360° images from outside and inside scenes. All models have similar absolute errors. ZoeDepth is slightly worse.

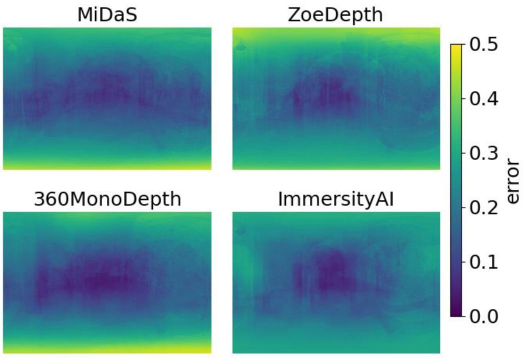


Figure 3.3. Average depth estimation error by UV position in the image. All models have similar error distribution: smallest error in the center, larger error towards left and right edges, and largest error towards the top and bottom edges. MiDaS and 360MonoDepth have their largest error at the bottom. ZoeDepth has its largest error at the top. Immersity AI has a similar error at the top and bottom.

and 0.4 at the bottom. It is also more severe towards the left and right edges of the image. Immersity AI has the least strong difference between the top and bottom edges and the middle. Both edges reach an error of up to 0.3. This shows how all models have problems estimating the depth in the polar region of the equirectangular projection. Even 360MonoDepth, the model specifically designed to account for the distortion of spherical images, cannot predict the correct depth. This problem with the polar regions can be seen even more clearly in Figure 3.4. It shows the average difference between a model's estimate and the ground truth based on the vertical position of the pixel. The x-axis represents the vertical height of the pixels in the image, and the x-axis shows the average error at that height. MiDaS is represented by the light blue line, ZoeDepth by the dark blue line, 360MonoDepth by the yellow line, and Immersity AI by the orange line. All models have the lowest error at 50 %, the vertical middle of the images, and significantly higher error towards the edges. In the center, MiDaS and 360MonoDepth have an average error of less than 0.12, while their error at the edges goes up to 0.5. ZoeDepth has a slightly lower error of 0.48 at the edges but a higher error of 0.16 in the center. Immsersity AI has a similar error at the center but achieves the smallest error at the edges with 0.34.

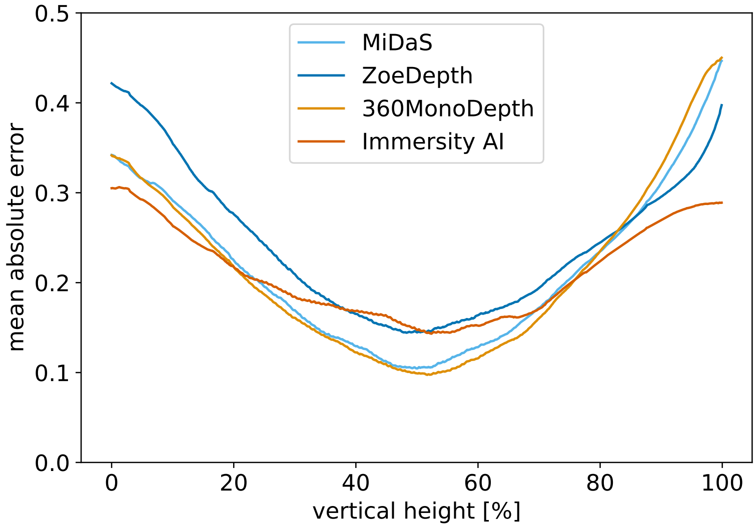
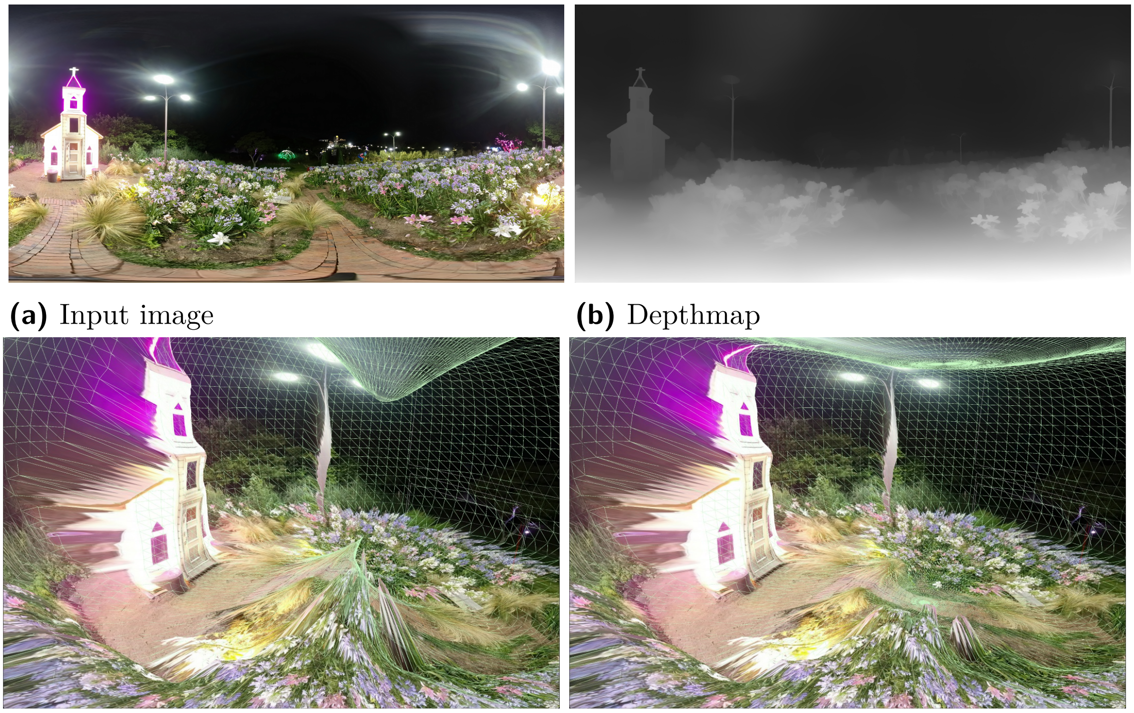


Figure 3.4. Average depth estimation error by vertical position in the image. Error peaks at the top and bottom edges of images. The smallest errors are in the center. Immersity AI has the smallest error of all models on the edges. MiDaS and 360MonoDepth have the smallest error in the middle.

This discrepancy at the polar regions of the depthmaps creates problems for the mesh generation and is addressed in Section 3.1.3. Regarding the non-polar regions, objective absolute depth correctness is not the most important metric. After generating 3D meshes using a prediction of each model, it is apparent that the subjective quality does not relate to the average corrective of the depthmap. Edge clarity, detail on objects, and relative correctness between objects are much more important than absolute correctness. A blurry edge around an object in the depthmap leads to long, broad gradients around them that become very apparent to the viewer. The relative depth between objects and inside objects is more important to a viewer's subjective perception than the correctness of the object's absolute distance. For example, the correct distance of background objects is not as important as the correctness of the details on the main foreground objects. This leads to the depth difference comparison with the ground truth not being a good measurement for the subjective quality of a depthmap for constructing the 3D geometry.

Subjective evaluation of a set of test images inside the VR application has shown that the depthmaps produced by Immersity AI consistently generate the best-looking 3D representations. Because of this, the user study uses these depthmaps.



(a) Input image

(b) Depthmap

(c) Initial mesh

(d) Rectified mesh

Figure 3.5. Errors in constructed mesh due to depthmap errors and correction through peak minimization. (a) input image. (b) depthmap for mesh generation. (c) constructed mesh without error adjustment has large peaks at the poles. (d) rectified mesh after error adjustment.

3.1.3 3D Mesh Construction for Spherical Images

Because of their design, most 360° cameras do not capture highly detailed information at the polar regions of images. This low-quality data, combined with the heavy distortion in those regions, leads the models to not estimate the depth accurately in the polar regions. Even the models designed explicitly for spherical images, like 360Monodepth, show this inaccuracy in a less extreme form. In the tested cases, this was always an underestimation of the actual distance. This causes the constructed meshes to protrude towards the origin position at the polar regions. The effect can be observed in Figure 3.5c, which was constructed from the input image seen in Figure 3.5a and the depthmap seen in Figure 3.5b. At the polar regions, the vertices in the middle of the constructed mesh protrude towards the geometry's center.

To remedy this, the mesh construction algorithm adjusts the depth in those regions. The construction starts with a simple, uniform sphere. First, the depth for each vertex is sampled from the depthmap at its respective texture coordinate. It is transformed to a distance value by multiplying with an image-specific multiplier. This modifier is calculated based on a reference value that has to be set in each image's meta-file. It is given by a UV coordinate and a goal distance value, which allows the program to calculate the modifier it needs to use to achieve the correct distance. Then, each vertex is positioned along its respective normal vector using the calculated distance. After the initial mesh is constructed, the system estimates the supposed height in both polar regions by averaging the height of the vertices along a circle around them. The depth values are adjusted using the estimated height to achieve that height more closely. In the edge region of the polar areas, the original depth and adjusted depth are lerped using a smoothstep function to create a smooth transition. Finally, the vertices are recalculated using the normals and the adjusted depth values. This has proven to rectify the problem and produce an improved mesh for all tested images. Figure 3.5d shows the algorithm output in the previous example case.

Figure 3.6 shows the CDF of the absolute adjustment made to the depth values during mesh construction. 80 % of values are adjusted by less than 0.2 and 33 % of values are not adjusted at all. At most the depth is adjusted by 0.64 and on average by 0.16.

3.1.4 3D Mesh Construction for Planar Images

While this thesis focuses on evaluating spherical images, the implemented solution also supports planar images. Mesh construction for planar images uses a different approach than the construction for spherical images. The process starts with a flat plane. First, the virtual camera position, the position in the application space where the camera would be placed to take the image, is calculated. The Field of View (FOV) of the camera that took the image is needed and must be defined in the image's meta-file. Equation 3.2 calculates the distance of the image plane origin position to the virtual camera. The scale is the width of the image plane, and the FOV is the provided FOV in degrees. Extrapolating the plane origin along the plane normal vector by the equation result gives the virtual camera position. Then, a normal direction vector from the virtual camera is calculated for each vertex,

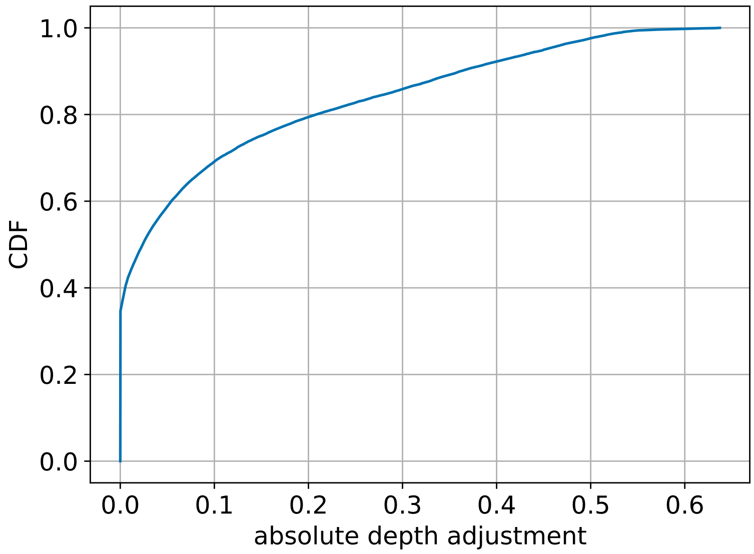


Figure 3.6. Cumulative distribution function (CDF) of the depth adjustment during mesh construction. The difference of depth values from before pole error correction to after.

which is multiplied with the same modifier as in the spherical process to get the final vertex position.

$$\text{cameraDistance} = \frac{0.5}{\text{scale}} \times \tan\left(\frac{\text{FOV}}{180} \times \pi\right) \tag{3.2}$$

In addition to the mesh construction, the planar process supports manual translation and scale values in the meta-file. This is used to position the planar image more optimally to improve the appearance of a virtual environment.

3.1.5 Dynamic Runtime Tessellation

While the collider mesh is constructed only once after an image is loaded, the visible mesh is built inside a shader from scratch for each frame. This enables the use of dynamic tessellation during the construction process, enabling a higher fidelity output while keeping to performance limits. First, the vertex shader calculates the raw depth value using the depthmap. Afterward, the hull shader uses this value to determine how strong the gradient of the edges of a triangle is. This gradient is used to determine the tessellation factor. At the same time, the hull shader clips any triangles outside of the view frustum. This prevents not rendered geometry from being tessellated and improves the performance. Finally, after the tessellation stage uses that factor to subdivide the geometry, the domain shader executes the same construction logic as the collider mesh generation.

Runtime tessellation is also used to create smoother geometry for visual effects such as the sine wave. The tessellation factor takes into account the area around the vertex, adjusting visual effects and increasing the fidelity of the mesh.

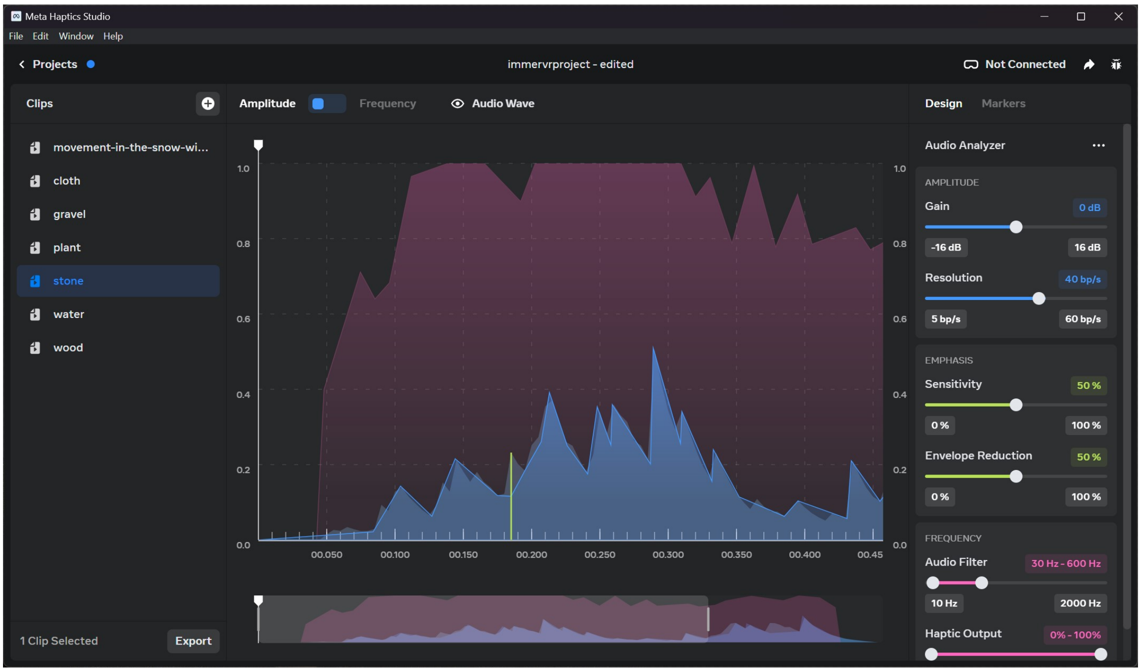
3.2 Feedback Mechanisms

This section introduces the implementation of the haptic and pseudo-haptic feedback mechanisms and how they are driven to influence the user experience.

Each image's meta-file defines different haptic materials. A haptic material defines the haptic and pseudo-haptic response that should be played for specific image parts. Each haptic material represents a combination of vibrohaptic, auditory, and visual feedback mechanisms. A material map is used to define which areas of the image are represented by which haptic material. The map uses specific color codes to determine regions for haptic material. The utilized color codes are taken from a list of 20 distinct colors [58] so they are easily distinguishable during the creation process. Currently, the material map has to be painted manually using any image editing program. However, it would be possible to partially or fully automate the process using modern image segmentation machine learning models. Virtual hands represent the user's controllers inside the virtual environment. They allow the user to see where they are moving their hands and interact with the image without taking them out of the experience. Furthermore, the user is able to send the virtual hands toward far away objects by holding down the grip button on the VR controller. This allows them to explore the surfaces of the entire image, even parts far out of reach.

3.2.1 Vibrohaptic Feedback

Haptic sequences are saved in *.haptic* files. They are defined by their file path for each haptic material. Their data is given in the JSON format and is loaded and deserialized as such. These haptic sequences have a 'frequency over time' curve and an 'amplitude over time' curve. They are created to match the profile of the



Gain 0 dB (-16 dB – 16 dB)

Resolution 40 bp/s (5 bp/s – 60 bp/s)

Sensitivity 50 % (0 % – 100 %)

Envelope Reduction 50 % (0 % – 100 %)

Audio Filter 30 Hz - 600 Hz (10 Hz – 2000 Hz)

Haptic Output 0% - 100%

Figure 3.7. Meta Haptic Studio [59], a tool for haptic sequence creation. In the center are frequency and amplitude curves, and on the right side are sequence adjustment settings.

audio file for the same haptic material and are played back synchronized with their respective audio to create a cohesive experience. To create these sequences, the Meta Haptic Studio [59] app is used. Figure 3.7 shows the app's interface. On the left side, audio files are displayed. In the center, the frequency and amplitude curves can be seen matched to the respective audio wave. The right-hand side tools allow for the adjustment of the output curves.

3.2.2 Auditory Feedback

Audio files are defined by a path inside the image's meta-file for each haptic material. They are loaded using UnityWebRequest and DownloadHandlerAudioClip. They are played back using spatialized audio sources on each hand, meaning an audio cue will be perceived as originating at the position the user is interacting with.

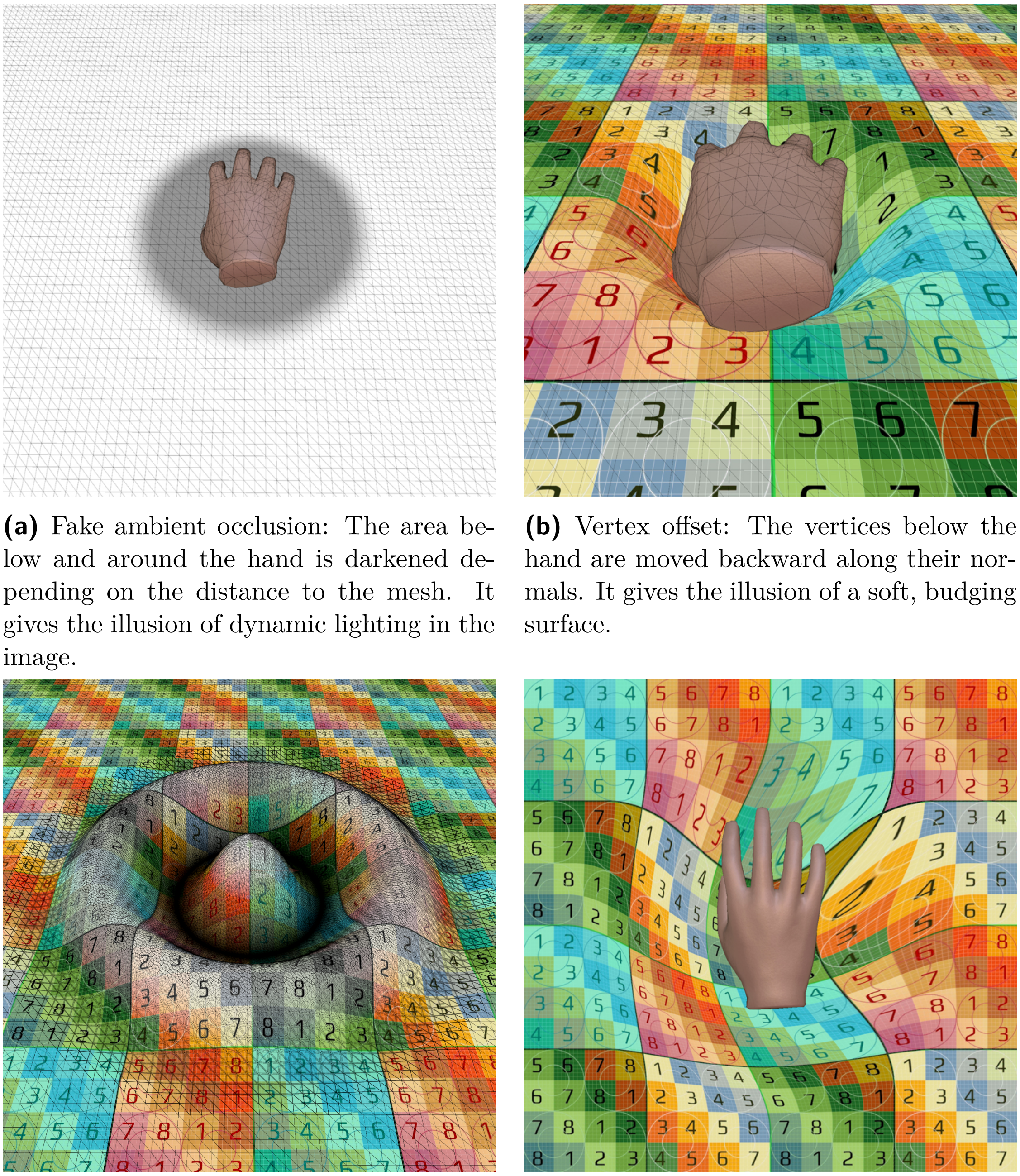
3.2.3 Visual Pseudo-Haptic Feedback

In addition to the pseudo haptics provided by audio cues, visual effects are added around the points the user touches. Fake ambient occlusion, vertex offset, UV distortion, and a ripple wave effect are implemented by extending the image shader.

The right-hand and left-hand positions are passed to two variables in the image shader. They are used to calculate the distances of each fragment to the hands. The shader uses that distance to darken the image around the area the hand hovers over or touches. This creates a fake ambient occlusion effect and the illusion that the hand casts a shadow onto the mesh. Figure 3.8a shows the ambient occlusion around the virtual hand on a white mesh. In addition to the hand positions, the hand origin to mesh distance, the UV coordinate, and the normal of the mesh contact point are collected for each hand via a ray cast and passed to the shader. They are used in combination to drive the rest of the visual effects. If the hand script detects a collision, the settings for the visual effects of the correlating haptic material are passed to the shader variables for the correct hand. Soft materials are visualized using a vertex offset along the normals as seen in Figure 3.8b. Depending on how 'soft' the haptic material is configured, the mesh will bulge inwards if the hand touches it. Another effect is the visual distortion around the touched area, as shown in Figure 3.8d. The UV texture coordinates are warped around the hand to mimic the effect of scrunching the surface, which can give the perception of moving around cloth. Lastly, a wave effect, shown in Figure 3.8c is implemented. It shows a sine wave around the contact point, defined by frequency, amplitude, distance, and speed. The wave is displayed using a vertex offset and coloring the high and low points of the wave with colors defined in the haptic material. For example, the sine wave can mimic water by adding a vertex offset, then darkening the through, and whitening the crest.

3.2.4 Collision Detection and (Pseudo-)Haptic Feedback Playback

The image's collider is realized as a concave mesh collider. In the context of the Unity engine, these collider types offer a reduced feature set. To work around their limitations, the collision detection uses a multi-faceted approach. Sphere colliders around the controller are used to determine a preliminary collision guess, and a ray cast is used to refine the position and retrieve the UV information from that point.



(a) Fake ambient occlusion: The area below and around the hand is darkened depending on the distance to the mesh. It gives the illusion of dynamic lighting in the image.

(b) Vertex offset: The vertices below the hand are moved backward along their normals. It gives the illusion of a soft, budging surface.

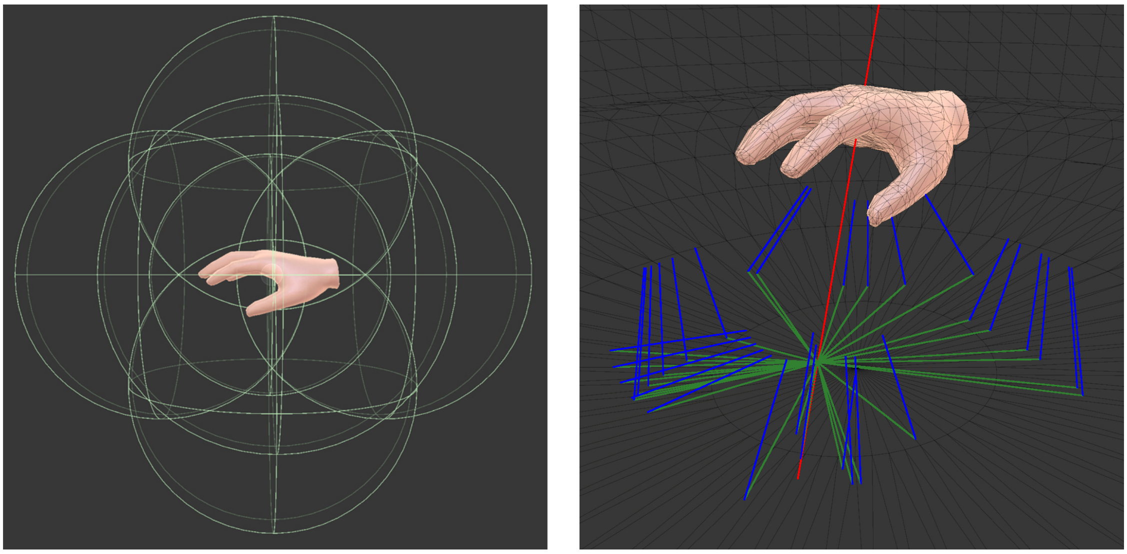
(c) Sine wave: Vertices are offset according to a sine wave. The top of the wave is brightened, and the bottom of the wave is darkened. The mesh is tessellated in the area of the wave to allow for smoother geometry. It enables effects such as water.

(d) UV coordinate distortion: The UVs around the hand position are distorted. This gives the illusion of a surface that is moved around and scrunched, for example, cloth.

Figure 3.8. Implemented visual feedback techniques.

Figure 3.9a shows the position of the sphere colliders around the controller. They only receive collision messages with the image collider if their origin point is on the front-facing side of the mesh face they are colliding with. With only a single sphere collider around the hand, the collision messages stop when the controller origin passes through the image mesh. To prevent this, multiple sphere colliders are placed around the controller, offset in each direction, to allow the collision detection to function even if the user partially or fully reaches through the image mesh.

During the update loop, all spheres' collision points and collision normals are weighted by their distance to the controller and averaged together. In the next step, a ray cast from the controller origin in the direction of the averaged collision point is made to get the controller's specific distance to the mesh and the UV coordinate at the collision point. This process is shown in Figure 3.9b. The positions where the green and blue lines meet are the collision points provided by the sphere colliders. Each



(a) Six sphere colliders positioned around the hand, offset into each axis direction. Sphere colliders can only register collisions with the 3D mesh if the sphere origin is on the front-facing side of the face it collides with. Multiple sphere colliders are used to enable collision detection, even if the hand moves behind the front faces.

(b) Sphere collision points with normals and the ray cast vector. Blue lines are collision point normals. Green lines are vectors from the collision point to the weighted average collision point. The red line is the ray cast vector. First, sphere colliders provided collision points around the closest mesh position. The averaged point gives a good estimation for the closest point. Then a ray cast provides a point on the mesh, distance, and UV coordinates.

Figure 3.9. Collision detection system with hand colliders and ray cast

blue line is the normal vector of a collision point, and the green lines are vectors from each collision point to the weighted average collision point. The red line is the final ray cast from the hand position toward the mean collision position.

If the user passes the controller too far through the image mesh, the collision point is replaced by a new ray cast from the head position to the controller position. During preliminary testing, this resulted in a better user experience than freezing the hand in place before it passes through the mesh or moving it outside using the collision point.

Finally, the UV coordinate of the ray cast is used to sample the material map and retrieve the haptic material at the interaction point. Then, the respective haptic sequence is passed to a haptic player instance, the audio file is set in the audio source, and the visual effects are passed to the image shader.

3.3 Study Design

A user study is conducted to determine the influence of the 3D representation and the haptic and pseudo-haptic feedback.

It is split into two parts. First, the influence of haptic and pseudo-haptic feedback on presence is researched; second, the influence of a 3D representation of images on presence. The hypotheses for the study are as follows.

- H1: The addition of haptic feedback (controller vibration) will significantly increase user presence in VR image viewing compared to no haptic feedback.
- H2: The combination of haptic feedback with visual pseudo-haptic feedback (visual distortions, waves, indenting) and audio feedback will further enhance presence compared to only haptic feedback.
- H3: A mesh 3D representation of the VR image increases the user presence compared to projecting the VR image onto a simple sphere.

Participants are shown a configuration with a specific image representation and no feedback or a combination of haptic/pseudo-haptic feedback enabled. After each configuration, a questionnaire is displayed and filled out inside the virtual environment. This is repeated multiple times for different configurations for each hypothesis. The answers to the prompts are collected on a 0 to 6 Likert scale with different meanings depending on the question. In the second part of the study, the Likert scale is offset and goes from -3 to +3. The study is conducted on a standalone Meta Quest 3 headset.

3.3.1 Influence of Haptic and Pseudo-Haptic Feedback on Presence

The first part of the study tests hypotheses H1 and H2. Participants are shown configurations that always enable the 3D mesh representation. The haptic, auditory, and visual pseudo-haptic feedback is enabled in the specific combinations listed in Table 3.2. After each image, the participant is asked to evaluate their experience using the prompts shown in Table 3.3 on a scale of 0-6.

	Haptic	Auditory	Visual
Control	/	/	/
Haptic	Yes	/	/
Haptic /w Audio	Yes	Yes	/
Full	Yes	Yes	Yes

Table 3.2. Overview of conditions for the first study part. Combinations of haptic and pseudo-haptic feedback.

	Prompt
G1	In the computer-generated world I had a sense of "being there".
SP1	Somehow I felt that the virtual world surrounded me.
SP2	I felt like I was just perceiving pictures.
SP3	I did not feel present in the virtual space.
SP4	I had a sense of acting in the virtual space, rather than operating something from outside.
SP5	I felt present in the virtual space.

Table 3.3. Prompts to determine presence. Igroup Presence Questionnaire [15]: G1, SP1 - SP5. G1 is a general presence prompt. SP are spacial presence prompts. Answers are given on a Likert scale of 0 to 6.

3.3.2 Influence of 3D Representation on Presence

The second part of the study tests hypothesis H3. Participants are shown one image with no haptic and no pseudo-haptic feedback enabled. Using the left and right VR controllers, they can change between two image representations. The sets of representations tested against each other are listed in Table 3.4. In each study condition, the participants view one image in two different representations and are asked to compare them in a questionnaire. The questionnaire uses the same prompts shown in Table 3.3, but on a scale of -3 to 3, with labels "more left-hand image", "similar", and "more right-hand image". Each condition is shown twice but with the inverted order regarding representations. The source images used for the comparisons are 180° stereo images with depthmaps. This enables them to be shown in any representation and thus be used for all comparisons.

	Representation 1		Representation 2
1	Mono	vs.	Stereo
2	Mono	vs.	3D Mesh
3	Stereo	vs.	3D Mesh

Table 3.4. Overview of conditions for second study part. Sets of image representations. Each set is a direct comparison between representation one and representation two.

Figure 3.10 shows how the two views of the stereo images and the depthmap are used to create each representation. Only the left view is rendered in both eyes to make the mono representation, while each view is rendered in the respective eye to make the stereo representation. For the 3D mesh representation, both eyes render the left view, but a mesh is constructed from the depthmap as described in Section 3.1.3 to display distances.

3.3.3 Study Controller

The study controller is a part of the program that runs the participants through the study. It iterates through the study configurations in a randomized order and prompts the participant with a questionnaire after each configuration. Each feedback mechanism can be globally enabled and disabled, and the desired image representation can be set to mono, stereo, or 3D mesh. The study controller instance uses this to control which image is displayed, which feedback mechanisms are used, and in what image representations.

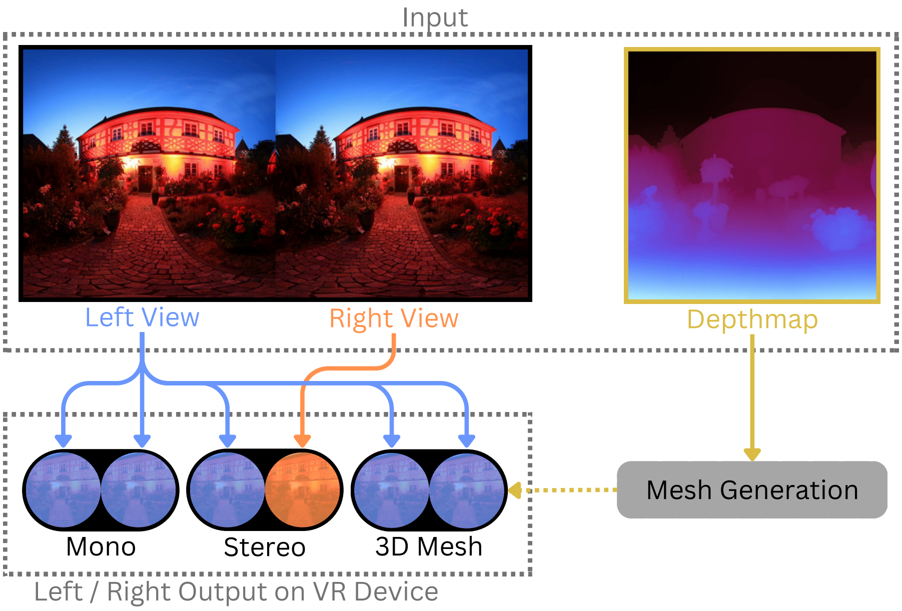


Figure 3.10. Usage of 180° stereo image views to construct mono, stereo, and 3D representation. The mono representation shows the left view in both eyes. The stereo representation shows the respective view of each eye. The 3D mesh representation shows the left view on both eyes but offsets the surface according to the depthmap to display distance.

4 Results and Evaluation

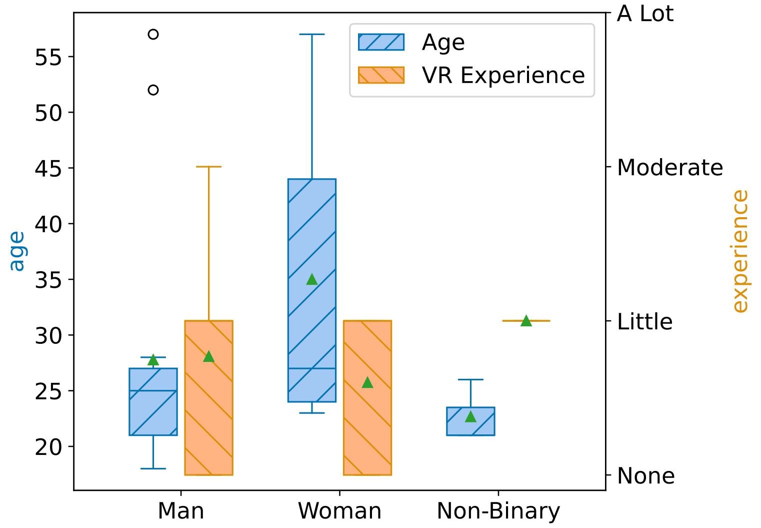


Figure 4.1. Age and VR experience distribution by gender of study participants. Average age of 29 years ± 12 std. 62 % men, 24 % women, 14 % non-binary. 33 % no VR experience, 57 % very little VR experience.

In this chapter, the study results are presented and analyzed. The study was run with the parameters given in Section 3.3. Before each participant's study execution, demographic information was collected using a questionnaire. After the experience, a second questionnaire was used to gather additional opinions. Figure 4.1 shows the demographic distribution of the participants. The left axis represents the participant's age in blue, and the right axis represents the participant's previous VR experience in orange. The x-axis splits the participants into three groups based on their reported gender identification. Each box shows the first quartile to the third quartile; the green triangles mark the mean values, the dark middle line the median, and the two handles extending at the top and bottom of each boxplot represent the farthest data point lying within 1.5 x the inter-quartile range. Circles represent outlier data points. The number of participants is 21, with 62 % of participants self identifying as men, 24 % as women, and 14 % as non-binary. Their mean age is

29 years with a standard deviation of 12. Of the participants, 33 % reported no previous VR experience, 57 % very little VR experience, and 10 % reported moderate VR experience.

4.1 Influence of Haptic and Pseudo-Haptic Feedback on Presence

Table 4.1. Average rating of each prompt per study condition. Prompts from Igroup Presence Questionnaire [15]; G1, SP1 - SP5. The control condition has no feedback enabled.

	Prompt	Control	Haptic	Haptic w/ Audio	Full
G1	In the computer-generated world I had a sense of "being there".	2.83	2.93	4.31	4.52
SP1	Somehow I felt that the virtual world surrounded me.	3.26	2.86	4.64	4.17
SP2	I felt like I was just perceiving pictures.	3.74	2.69	2.12	1.90
SP3	I did not feel present in the virtual space.	2.33	2.45	3.29	3.52
SP4	I had a sense of acting in the virtual space, rather than operating something from outside.	1.83	2.90	3.07	3.69
SP5	I felt present in the virtual space.	2.48	2.67	3.57	4.00

This section presents the results of the user study testing the influence of haptic, auditory, and visual feedback on the feeling of presence inside virtual environments constructed from pictures as described in Section 3.3.1.

Table 4.1 shows the averaged results of the presence questionnaire prompts for each study condition. The first column shows the prompts used and the remaining columns show the results averaged over all participants and condition instances. Each prompt has an answer range of zero to six. Figure 4.2 presents this data in a graphical format. The x-axis splits the different prompts, and the y-axis gives the rating of each prompt in the specific condition with its respective 95 % confidence interval. G1, SP4, and SP5 have the same order of condition ratings. Control has the lowest rating, closely followed by the Haptic condition. Haptic w/ Audio and Full have a significantly higher rating, the Full condition slightly edging out the

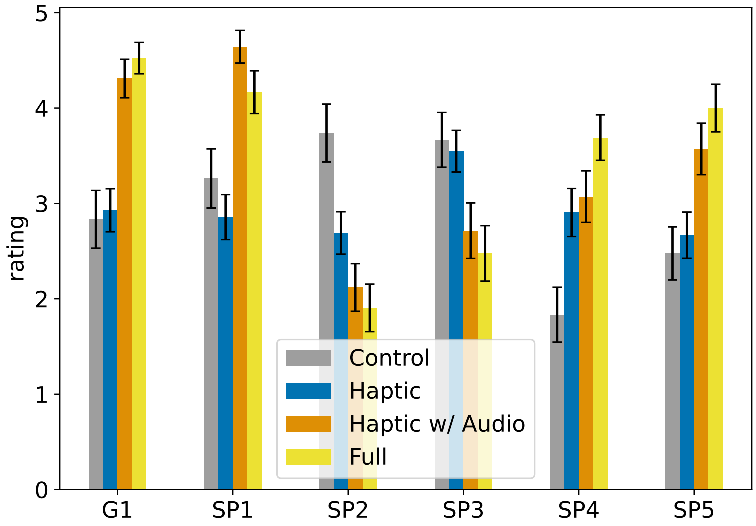


Figure 4.2. Average rating of each prompt per study condition. The answers to each questionnaire prompt are averaged over all participants. Visualization of Table 4.1. SP2 and SP3 are adverse prompts. Control and Haptic are rated significantly lower than Haptic w/ Audio and Full for all but the adverse prompts. Full is the best condition for five out of six prompts.

Haptic w/ Audio condition. SP2 and SP3 show the opposite order because they are negated prompts. Only SP1 has a slightly different result, with the Control condition being slightly higher rated than the Haptic condition and the Haptic w/ Audio condition being slightly higher than the Full condition.

Inverting the results of the negated prompt and averaging over the data of each condition produces a mean presence score. Figure 4.3 shows the presence scores for each condition with its respective 95 % error range. The x-axis shows condition labels for each bar, while the y-axis represents the mean presence score. It shows the same ordering as the graph of the individual prompts suggests: the Control condition having the lowest presence score, closely followed by the Haptic condition; Haptic w/ Audio condition and the Full condition being significantly higher, with the Full condition slightly outperforming the Haptic w/ Audio condition. Control and Haptic, as well as Haptic w/ Audio and Full, have overlapping error ranges.

Another visualization for the data is Figure 4.4. It shows the frequency of how the conditions ranked in terms of their presence score for each participant. The x-axis splits the ordered ranking, and the y-axis shows the frequency in which the condition takes that rank. Full is clearly the preferred condition, securing the first ranking

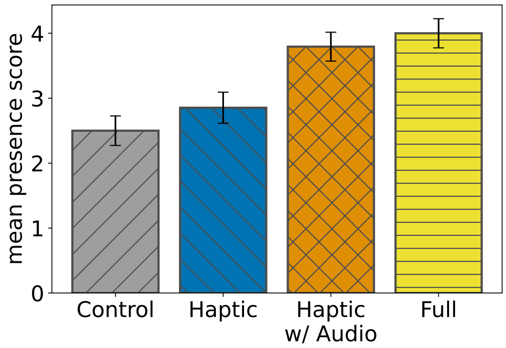


Figure 4.3. Mean presence scores with 95 % confidence intervals. The score is calculated using all prompts. Haptic w/ Audio and Full have a significantly higher presence than both Control and Haptic. Control has a slightly lower score than Haptic, and Full has a slightly higher score than Haptic w/ Audio.

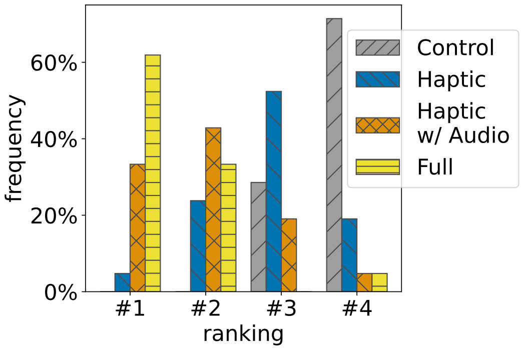


Figure 4.4. Frequency of ranking of conditions by mean presence score. Bars show the percentage of participants where the corresponding condition ranked at that index. Control ranked last for 71 % of participants. Haptic ranked third or last for 71 % of participants. Haptic w/ Audio ranked first for 32 % and second for 40 % of participants. Full ranked first for over 60 % of participants. The ranking shows a clear ordering of Full > Haptic w/ Audio > Haptic > Control.

across 62 % of the participants and the second ranking across 33 %. While not as dominant as Full, the Haptic w/ Audio condition garners a significant number of top rankings. For 33 % of participants it ranks first and for 43 % second. The Haptic condition ranks third for 52 % and last for 19 %. Control receives the lowest rankings. It occupies the last spot for 71 % of participants and the second to last spot for 29 %. This plot again shows a clear hierarchy of preference for the conditions. Full leading the ranking, followed by Haptic w/ Audio, then Haptic, and finally Control.

A one-way ANOVA test on the data of the four conditions has returned a p-value of less than 1 %, suggesting that the population means between at least two conditions are statistically significantly different.

A pairwise Tukey's honestly significant difference (HSD) test is used to determine which conditions have different presence score means from each other at a 95 % confidence level. HSD tests each combination of conditions against each other for the null hypothesis H0, which is that the conditions have the same mean. H0 is rejected if the p-value of the combination is below 5 %, suggesting with high confidence that

Table 4.2. Pairwise Tukey HSD test for statistical significance of condition pairs, reporting the mean difference between the respective two conditions. The right column shows the 95 % confidence interval. A star after mean difference signals a p-value lower than 5 %.

group1	group2	meandiff	CI
Control	Full	1.50*	[0.77 2.23]
Control	Haptic	0.35	[-0.38 1.08]
Control	Haptic w/ Audio	1.29*	[0.56 2.02]
Full	Haptic	-1.15*	[-1.88 -0.42]
Full	Haptic w/ Audio	-0.21	[-0.94 0.52]
Haptic	Haptic w/ Audio	0.94*	[0.21 1.67]

the combination of conditions has different means. Table 4.2 shows the results of the HSD, the combination of conditions on the left, followed by their mean sample difference, and the confidence interval. A star behind the mean difference marks H0 as rejected for this combination. Confidence interval is the range of values within which we are 95 % confident that the true population mean lies. The results show that the Haptic w/ Audio and Full conditions deliver a higher presence score than the Control or Haptic conditions to a 95 % confidence level. However, the null hypothesis was not rejected for the comparisons between Control and Haptic and between Haptic w/ Audio and Full. This does not mean that each combination has no difference in experienced presence, but our study samples do not allow us to draw statistically significant conclusions.

Considering this, the results suggest that auditory feedback can significantly elevate the feeling of presence. Furthermore, it hints that haptic feedback alone slightly increases presence compared to when no feedback mechanisms are used but is not able to confirm it to a significance level of 5 %. Similarly, it hints that adding visual feedback on top of haptics and auditory cues slightly enhances the feeling of presence but cannot confirm it to the confidence level of 95 %. An exact comparison between the feedback mechanisms alone cannot be made as the study conditions only used them in combinations and not individually.

Responses to the post-questionnaire support those findings. Participants were asked to select the feedback mechanism that most enhanced their experience and which, if any, most decreased their experience. Figure 4.5 shows the results of those questions. The x-axis splits into the different feedback mechanisms, and the y-axis shows the percentage of participants who selected each method. Visual feedback is the most

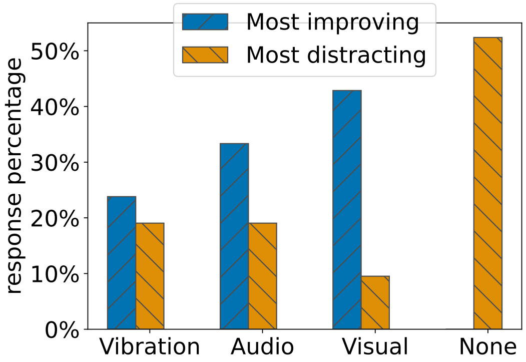


Figure 4.5. Best and worst voted feedback technique by category. Data was gathered from post-questionnaire questions for the feature most increasing and most decreasing the experience. 50 % did not find any feedback distracting. visual was voted most improving by the most participants, closely followed by audio and lastly by haptic feedback. Haptic and audio were voted to distract from the experience by 20 %.

popular technique, with 43 % selecting it as the best feedback and only 9.5 % thinking it decreased their experience. In the second place, audio feedback was selected as the best technique by 33 % of the participants and thought as decreasing to the experience by 19 %. Haptic feedback was the least popular feature, with 24 % voting it the best and 19 % believing it decreased their experience. 52 % of participants believed no feedback technique was decreasing their experience. The most common critique of haptic feedback is insufficient variance between different surfaces. Multiple participants describe the feeling of vibration as 'weird' or 'strange'. Auditory feedback has fewer negative comments. Some participants noted that more variance in response to the user's interaction and speed is required. They expect different sounds when hitting or scratching the surface and varying audio cues depending on the speed at which they move their hand along the surface. Multiple participants also reported that background sounds unrelated to their interaction may help increase their immersion. Examples they listed are ocean, nature, and crowd sounds. No participants noted any specific dislike toward visual feedback in their report. While a few participants stated that it was not very noticeable, most described it as 'nice', 'interesting', or 'cool'.

4.2 Influence of 3D Representation on Presence

This section presents the results of the user study testing the difference between mono, stereo, and 3D mesh image representation in terms of presence as described in Section 3.3.2.

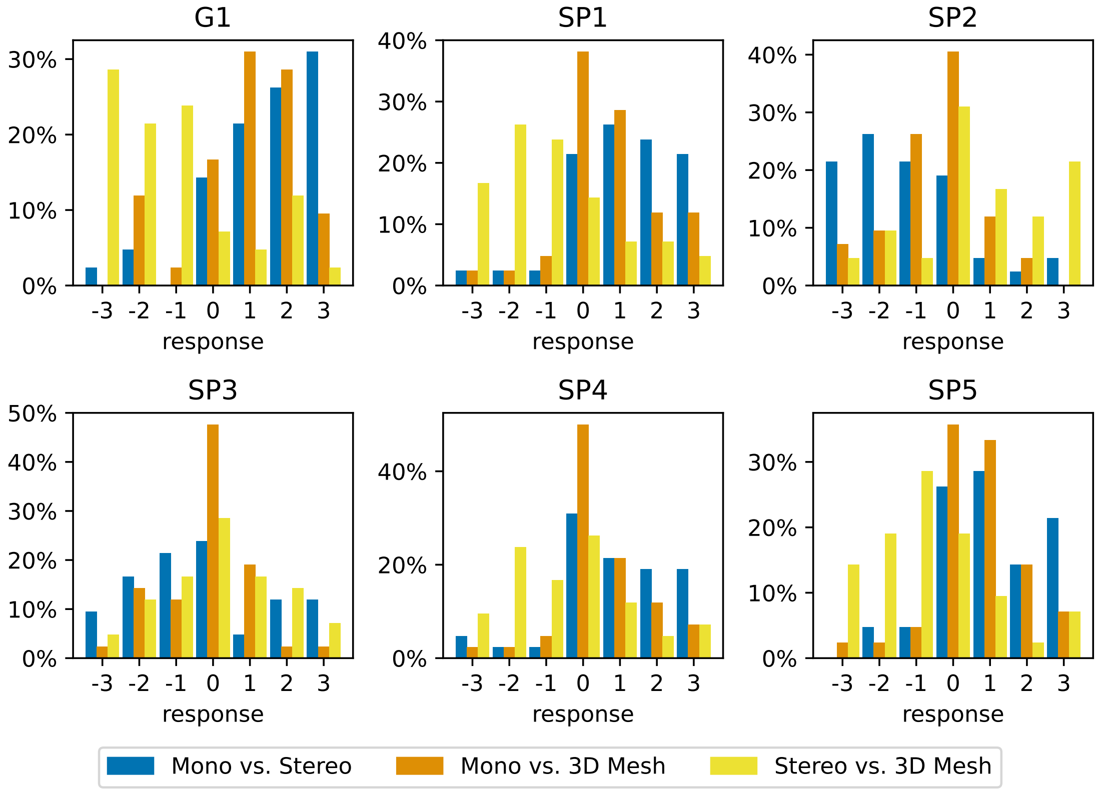


Figure 4.6. Response frequencies for each image representation comparison. The x-axis is the answer possibilities on a Likert scale of -3 to 3. The y-axis shows the percentage of participants that selected each answer to the prompts. SP2 and SP3 are adverse prompts.

Figure 4.6 shows the response frequencies for each presence questionnaire prompt with respect to the test condition. Each subfigure has the answer possibilities on a scale from negative three to positive three. Negative numbers mean leaning toward the comparison's first representation, and positive numbers mean leaning toward the second representation. The y-axis shows the percentage of participants that selected that answer. The colors represent the different study conditions of mono representation vs. stereo representation, mono representation vs. 3D mesh representation, and stereo representation vs. 3D mesh representation. The prompts are the same as in the first part of the study and are listed in Table 3.3. To interpret the results, it has to be noted that SP2 and SP3 are, again, adverse prompts. For the Mono vs. Stereo comparison, all graphs favor stereo representation. The distributions of answers to the positive prompts lean toward positive numbers, representing the stereo view. The distributions of the adverse prompts lean toward negative numbers, representing the mono view. Similarly, the distributions of answers for the Mono vs. 3D Mesh comparison lean toward the 3D mesh representation, however less pronounced. The most prominent answer for five out of six prompts is zero, representing a larger indifference between the two representations. For the Stereo vs. 3D Mesh comparison, the answer distributions to all prompts except SP3 show a considerable preference toward the mono representation. The distribution of answers to SP3 is more evenly spread between the two representations.

Figure 4.7 shows the percentage of participants favoring each image representation per prompt. Each subgraph represents one comparison where the x-axis shows the different prompts, and the bars represent the percentage of participants favoring one representation or being indifferent between both representations. The y-axis is the percentage of participants. Indifferent means the participants answered zero for that prompt. Favoring one representation means the participants answered the prompt in any amount favoring one representation. Blue represents a favoring of the mono representation, orange favoring of the stereo representation, yellow favoring of the 3D mesh representation, and gray represents being indifferent between two representations.

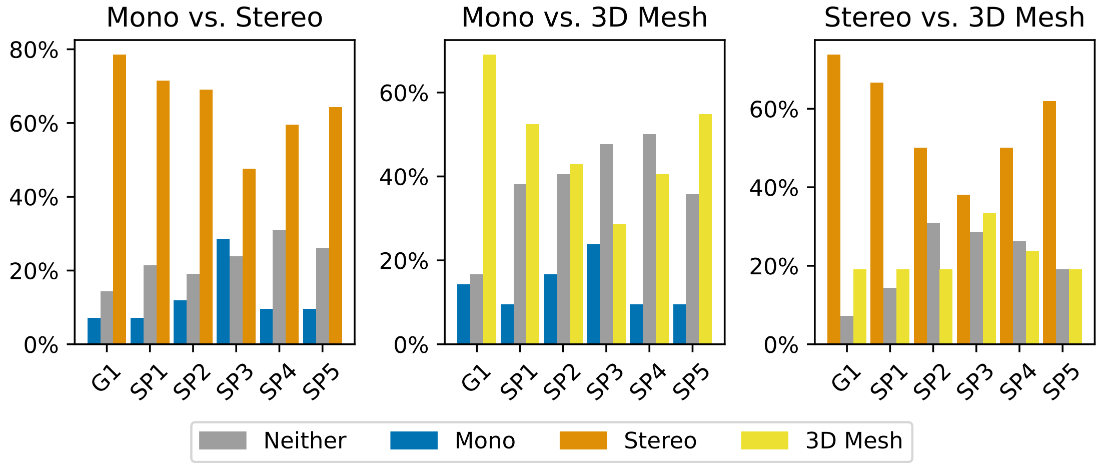


Figure 4.7. Favoring of one image representation over another for each prompt. One plot per comparison. The bars in each plot represent how many participants preferred each representation and how many were indifferent.

In the Mono vs. Stereo comparison, most participants favor the stereo representation. On five prompts, more than 60 % of the participants favor the stereo representation, while less than 12 % prefer the mono representation. SP3 shows the same preference, although less pronounced. 47 % of participants prefer the stereo representation while 29 % prefer the mono representation. This observation of SP3 having less pronounced results is also true in the other comparisons. For all but SP3, over 40 % of the participants favor the 3D mesh representation over the mono representation. With 17 %, the percentage favoring the mono representation over the 3D mesh is almost as low as the percentage favoring the stereo representation over the mono representation. However, in the Mono vs. 3D Mesh comparison, more participants are indifferent to the type of representation. For five out of six prompts, 34 % of participants have no clear preference. In the Stereo vs. 3D Mesh comparison, more than 50 % of the participants favor the stereo representation for all prompts but SP3. The 3D mesh representation is favored by 33 % of participants.

Figure 4.8 displays the average responses to each prompt and the combined value for the different representation comparisons. The x-axis lists the prompts and the combined value. Colored bars represent the image representation comparisons. Their height shows the averaged response value with the 95 % confidence interval. The combined value is the average of the responses to all prompts and all study condition instances.

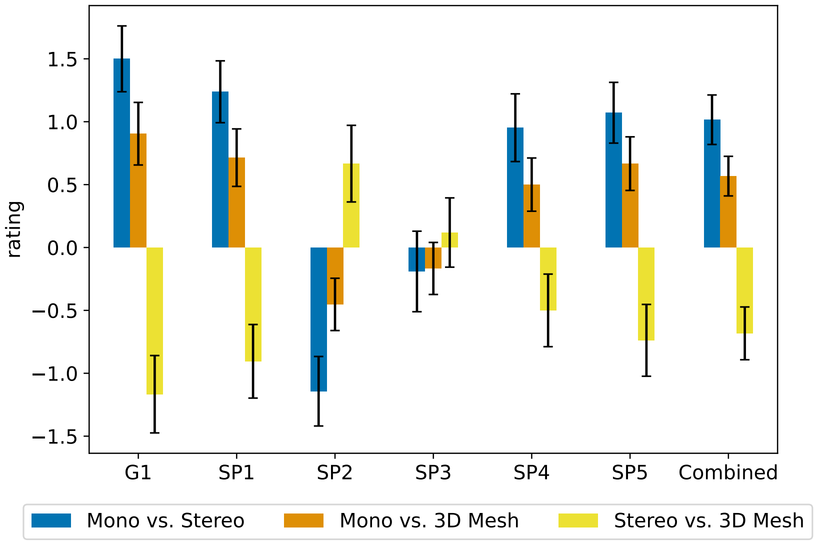


Figure 4.8. Average rating for prompts and combined rating for each image representation comparison. Answers are on a Likert scale between -3 and 3.

The Mono vs. Stereo comparison trends on average 1.0 points towards the stereo representation and always at least 0.15 points. The 3D mesh representation also rates higher than the mono representation. In the Mono vs. 3D Mesh comparison, the results trend on average 0.6 points toward the 3D mesh representation. Of the two, the stereo representation is rated better than the 3D mesh presentation. In the Stereo vs. 3D Mesh comparison, the results trend toward the stereo representation by, on average, 0.5 points. SP3, while following the same trend as the other prompts, has a much smaller rating in any of the comparisons. This is most likely because the prompt's negative wording combined with the answer labels' negative wording is confusing to participants.

A Wilcoxon Signed-Rank Test is conducted to check the results of the prompts and combined values for significance. The test's null hypothesis is that the distribution of responses is symmetric about zero. The alternative hypothesis for Mono vs. Stereo and Mono vs. 3D Mesh is that the distribution mean is greater than zero. The alternative hypothesis for Stereo vs. 3D Mesh is that the distribution mean is less than zero. This is because we expect the stereo and 3D mesh representation to beat the mono representation, but the 3D mesh representation to loose against the stereo representation.

Table 4.3 shows the sample mean of the prompts and the combined value for each comparison. A star behind the mean value signifies a p-value of less than 1 % on the Wilcoxon Signed-Rank Test. It means the image representation of the alternative hypothesis is preferred to a confidence level of 99 %. For the Mono vs. Stereo comparison, the results for all prompts, except SP3, show a statistically significant preference for the stereo representation. Similarly, for the Mono vs. 3D Mesh com-

Table 4.3. Means of responses to image representation prompts. The column labeled 'combined' is mean over responses to all prompts. A star signifies a p-value of less than 1 % on a Wilcoxon signed-rank test. It tests the null hypothesis that the response distribution is symmetric about zero against the respective alternative hypothesis ('greater than 0' for the conditions Mono vs. Stereo and Mono vs. 3D Mesh; 'less than 0' for Stereo vs. 3D Mesh).

	G1	SP1	SP2	SP3	SP4	SP5	Combined
Mono vs. Stereo	1.5*	1.24*	1.14*	0.19	0.95*	1.07*	1.02*
Mono vs. 3D Mesh	0.9*	0.71*	0.45*	0.17	0.5*	0.67*	0.57*
Stereo vs. 3D Mesh	-1.17*	-0.9*	-0.67	-0.12	-0.5	-0.74	-0.68*

parison, the results of the same prompts show a statistically significant preference for the 3D mesh representation. Only GP1, SP1, and the combined value show a statistically significant preference in the Stereo vs. 3D Mesh comparison. While the means of the other prompts also sway towards the stereo representation, only GP1, SP1, and combined do so to the confidence level of 99 %.

These results suggest that the stereo representation is able to create the highest feeling of presence out of the tested techniques. Furthermore, participants also felt a higher sense of presence for the 3D mesh representation than the mono representation, suggesting that the gained 3D effect outweighs the artifacts produced by the technique. This is also supported by statements of the participants in the post-questionnaire. 95 % selected that the 3D representation improved their experience. Some participants noted that while the stereo representation has the most convincing 3D effect, the lack of head translation freedom makes them feel dizzy. The most common criticisms of the 3D mesh representation are the edge artifacts and the inconsistent distance on far objects. For example, the too-close distance of the sky took them out of the experience.

4.3 Shortcomings on the Study

This section discusses the study's shortcomings and details improvements that could be made to the design.

The most prominent shortcoming is the setup of prompt SP3 in the second part of the study. SP3 from the igroup, "I did not feel present in the virtual space." is worded negatively, and the labels for the Likert scale in the A/B study were also worded negatively. On the lower end, the answer was "less for left-hand image", and on the higher end, "less for right-hand image". This can be interpreted in either direction, the lower end meaning "less presence" for the left-hand image or "feeling less not present" for the left-hand image. With the second interpretation, the results of the prompt should match the prompt SP5, "I felt present in the virtual space." and with the first interpretation, they should match the inverted results. Neither of those is the case, and instead, prompt four is the only prompt that has, on average, almost no preference in any study condition. Reports of the study participants also support this; multiple people exclaimed during their study execution or on their post-questionnaire that this prompt confused them. In the future, this problem can be avoided by rephrasing the scale labels to be more clear about which side means less presence for which image representation.

Another shortcoming of the study is its small sample size. With only 21 participants making statistically significant statements in respect to the demographics is not possible. Furthermore, the little diversity in terms of previous VR experience does not represent regular users of VR applications, the group most likely to use and benefit from the features investigated in the thesis.

Lastly, an improvement to the study execution would have been giving the participants examples of the different feedback methods before they went through the study conditions. The order of conditions was randomized, which led some participants to start with instances without any feedback mechanism. This made them unsure of what was being tested until a condition with feedback was played. This could have been avoided by prefixing the study with an example environment showcasing the different feedback methods being tested.

4.4 Runtime Performance Analysis

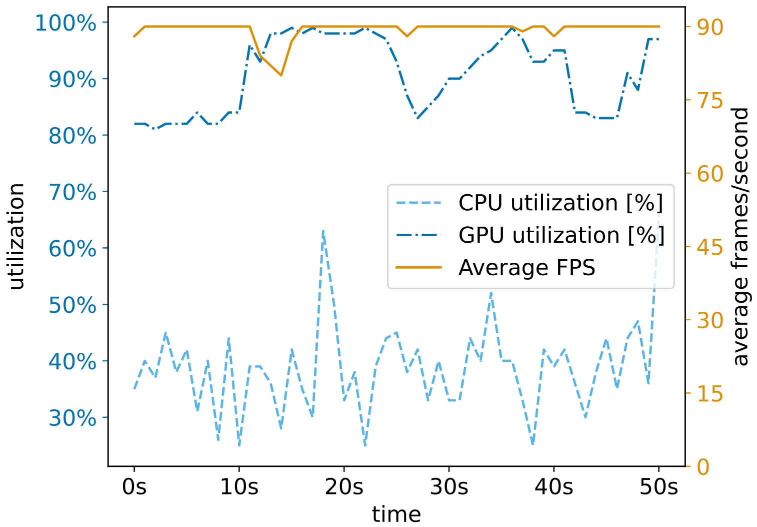


Figure 4.9. Performance of the implementation on the Meta Quest 3. Graphs show the CPU utilization, GPU utilization, and average FPS during 50 s period. The frame rate 10 % lows is 88 fps, the 1 % lows is 81 fps.

This section analyzes the performance of the implemented solution and suggests ways to improve it in the future. Using the OVR Metrics Tool [60], the performance at runtime execution on a standalone Meta Quest 3 is captured. The render resolution per eye is 2352 px by 2464 px after all render scale multipliers have been applied. Figure 4.9 shows the CPU utilization, GPU utilization, and average frames per second (FPS) during a 50 s time frame. The x-axis shows the time in seconds, the left y-axis shows the percentage utilization, and the right y-axis shows the frames per second. The light blue graph shows CPU utilization, the dark blue is GPU utilization, and the orange is the frame rate. The CPU utilization consistently stays below 60 % and the GPU utilization stays between 80 % to 100 %. The frame rate stays consistent between 88 fps and 90 fps, only occasionally dipping slightly below when changes in the rendered scene occur, such as at the 15 s timestamp. At that time, the camera frustum moved a more tessellated part into view, which can also be seen in the GPU performance increasing right after that point. Most likely, the dip was caused by the headset needing to adjust the GPU clock speed to match the higher performance requirement. This shows that the application is limited by GPU performance, mainly because of the high triangle count used on the 3D mesh representation to get clean edges around objects. The tessellation settings are tuned to stay just within the performance limits of Meta Quest 3 and still achieve a smooth frame rate. VR devices with less powerful graphics processing will not be able to uphold this frame rate.

To increase the performance of the rendering, the mesh generation is the most critical subsystem to optimize. In its current version, the construction is very simplistic, only taking into account the actual need for high detail during the runtime tessellation process. Tessellation is a very expensive technique, especially on standalone VR devices. Furthermore, even the technique used here, which tessellates dynamically based on gradients, creates more geometry than necessary. A smarter approach could be analyzing the needs of the mesh based on the depthmap during the initial mesh construction and placing the vertices along object edges and at the start and end of gradients.

A more straightforward approach to producing a higher-detail mesh without tessellation could be to utilize a destructive process. First, a high-detail mesh would be generated, and then, progressively, neighboring vertices would be merged together if they have a similar depth.

Finally, the mesh generation could be improved by taking more depth samples into account to represent object distances more accurately. Depth estimation models often do not estimate distant objects as being the long distances away that they should be. In the current iteration, one sample distance is used to calculate the depth multiplier and construct the mesh. This leads to some objects being the correct distance from the user but others being incorrect. This usually means the objects close to the viewer look correct, but the background, like the sky, is not far enough away. In the post-questionnaire of the study, multiple participants mentioned this, taking them slightly out of the experience. One solution to adjust this could be requiring multiple sample distances at different depths. Then, during the mesh construction process, the multiplier can be calculated dynamically for each depth value based on the sample distances provided.

5 Conclusion and Future Work

5.1 Conclusion

The increasing prevalence of consumer-friendly Virtual Reality (VR) devices has opened up exciting possibilities for enhancing how we experience digital media, particularly in the realm of image viewing. The immersive nature of VR provides a unique opportunity to elevate the experience of viewing spherical images, creating a sense of presence that traditional 2D displays cannot replicate. However, the current landscape of VR image-viewing applications lacks the tactile and interactive elements that could further deepen the sense of immersion. The research presented in this thesis aimed to address this gap by exploring the potential of haptic feedback, interactive audio cues, pseudo-haptic visual feedback, and 3D depth information to transform static spherical images into dynamic and interactive experiences.

To achieve this goal, depthmaps were generated from spherical images using existing machine learning depth estimation models. This thesis addressed the processing of depthmaps to correct for inaccuracies produced by the equirectangular projection and construct 3D meshes that allow for a more realistic and interactive representation of the scene. It also explored the design and implementation of haptic feedback, audio cues, and visual feedback into the generated 3D environment to enhance the user experience. Finally, a user study was conducted to evaluate the effectiveness of the 3D mesh representation and different feedback mechanisms on the sense of presence in VR image viewing.

The results of the study provide valuable insights into the effectiveness of the different feedback mechanisms and the 3D mesh representation. The findings suggest that haptic feedback alone slightly increases the sense of presence while adding audio cues significantly increases it. Furthermore, it indicates that a combination of haptic, audio, and visual feedback creates the highest feeling of presence, but not to a statistically significant degree over the combination of haptic and audio feedback.

Finally, it shows that of mono, stereo, and 3D mesh representation, the stereo representation delivers the highest feeling of presence, with the 3D mesh representation being preferred over the mono representation.

The research presented in this thesis shows the positive effects on the sense of presence when incorporating haptic and pseudo-haptic feedback in VR image viewing. Furthermore, it gives insight into what is needed to implement the feedback mechanisms and how the different feedback techniques affect the feeling of presence. Lastly, it shows how changing the image representation can have large effects on the amount of presence experienced. These insights can guide the development of more immersive and interactive VR applications, particularly those focused on image viewing.

5.2 Future Work

Setting up images to support the feedback methods and 3D mesh representation presented in this thesis is currently a manual process. In the future, machine learning could aid in automating the generation of material maps using image segmentation models. Furthermore, semantic segmentation models could provide descriptions of surfaces to configure the haptic materials automatically. Improvements to the mesh construction could be made using edge detection or a destructive process as described in Section 4.4. The construction could also be replaced with a machine-learning approach. There is also the potential of expanding the work to video content. For this, the depthmap and haptic map would need to be extended into the time dimension, which requires special care to keep temporal consistency between frames.

Additionally, the study's limitations, such as the small sample size, the short duration spent with stereo images, and the potential influence of prior VR experience, highlight areas for improvement in future research designs. A direct extension of this work could be to investigate the effects of the feedback mechanism individually or test if the 3D mesh representation with the feedback techniques delivers a higher presence than the stereo representation with no feedback. Another direction could be investigating how the combination of stereo representation and feedback mechanism could work. Aside from spherical images, this work also presented a process for creating 3D representations of planar images. However, the study did not investigate the effect on the sense of presence for these formats because of time limitations. The 3D mesh representation of planar images increases the FOV covered by the scene by stretching features, such as the floor towards the viewer. This could lead to a larger increase in presence between the mono and 3D mesh representations than for spherical images. Thus, another extension to this work could be investigating the 3D mesh representation, haptic, and pseudo-haptic feedback for other image formats.

Bibliography

[1] "AR and VR Headsets Market Size, Share, and Trends 2024 to 2034." (2024), [Online]. Available: https://www.precedenceresearch.com/ar-and-vr-headsets-market (visited on 09/05/2024) (cit. on p. 1).

[2] "VR Photo Slideshow." (2024), [Online]. Available: https://www.meta.com/en-gb/experiences/6637009056380469 (visited on 08/06/2024) (cit. on p. 3).

[3] "VR Photo Viewer." (2017), [Online]. Available: https://store.steampowered.com/app/531980/VR_Photo_Viewer (visited on 08/06/2024) (cit. on p. 3).

[4] "Witoo VR photo viewer." (2018), [Online]. Available: https://store.steampowered.com/app/970210/Witoo_VR_photo_viewer (visited on 08/06/2024) (cit. on p. 3).

[5] "Virtual Home Theater VR Video Player." (2019), [Online]. Available: https://store.steampowered.com/app/989060/Virtual_Home_Theater_VR_Video_Player (visited on 08/06/2024) (cit. on p. 3).

[6] "immerGallery." (2024), [Online]. Available: https://www.immervr.com/immerGallery.html (visited on 08/06/2024) (cit. on pp. 3, 13).

[7] "immerVR." (2024), [Online]. Available: https://www.immervr.com (visited on 08/06/2024) (cit. on p. 3).

[8] M. Slater and S. Wilbur, "A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 6, pp. 603–616, Dec. 1997. eprint: https://direct.mit.edu/pvar/article-pdf/6/6/603/1623151/pres.1997.6.6.603.pdf (cit. on p. 4).

[9] F. Biocca, "The Cyborg's Dilemma: Progressive Embodiment in Virtual Environments," *Journal of Computer-Mediated Communication*, vol. 3, no. 2, JCMC324, Sep. 1997 (cit. on p. 4).

[10] F. Biocca, C. Harms, and J. K. Burgoon, “Toward a More Robust Theory and Measure of Social Presence: Review and Suggested Criteria,” *Presence: Teleoperators and Virtual Environments*, vol. 12, no. 5, pp. 456–480, Oct. 2003. eprint: https://direct.mit.edu/pvar/article-pdf/12/5/456/1623957/105474603322761270.pdf (cit. on p. 4).

[11] H. Dinh, N. Walker, L. Hodges, C. Song, and A. Kobayashi, “Evaluating the importance of multi-sensory input on memory and the sense of presence in virtual environments,” in *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*, 1999, pp. 222–228 (cit. on p. 4).

[12] M. Slater, V. Linakis, M. Usoh, and R. Kooper, “Immersion, presence and performance in virtual environments: an experiment with tri-dimensional chess,” in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, ser. VRST ’96, Hong Kong: Association for Computing Machinery, 1996, pp. 163–172 (cit. on p. 4).

[13] B. G. Witmer and M. J. Singer, “Measuring Presence in Virtual Environments: A Presence Questionnaire,” *Presence: Teleoperators and Virtual Environments*, vol. 7, no. 3, pp. 225–240, Jun. 1998 (cit. on p. 4).

[14] M. Slater and M. Usoh, “Representations Systems, Perceptual Position, and Presence in Immersive Virtual Environments,” *Presence*, vol. 2, pp. 221–233, Jan. 1993 (cit. on p. 4).

[15] “Igroup Presence Questionnaire (IPQ).” (2024), [Online]. Available: https://www.igroup.org/pq/ipq/download.php (visited on 08/15/2024) (cit. on pp. 4, 29, 33).

[16] C. Hendrix, “Exploratory Studies on the Sense of Presence in Virtual Environments as a Function of Visual and Auditory Display Parameters,” 1994 (cit. on p. 4).

[17] A. S. Carlin, H. G. Hoffman, and S. Weghorst, “Virtual reality and tactile augmentation in the treatment of spider phobia: a case report,” *Behaviour Research and Therapy*, vol. 35, no. 2, pp. 153–158, Feb. 1997 (cit. on p. 4).

[18] “Oculus Touch Buffered Haptic Feedback.” (2016), [Online]. Available: https://www.roadtovr.com/oculus-touch-buffered-haptics-feedback-sdk-documentation (cit. on p. 5).

[19] "Index Controller Specs." (2019), [Online]. Available: https://www.valvesoftware.com/en/index/controllers (visited on 01/15/2024) (cit. on p. 5).

[20] "Touchly." (2023), [Online]. Available: https://touchly.app (visited on 01/15/2024) (cit. on p. 5).

[21] A. Lécuyer, "Simulating Haptic Feedback Using Vision: A Survey of Research and Applications of Pseudo-Haptic Feedback," *Presence: Teleoperators and Virtual Environments*, vol. 18, no. 1, pp. 39–53, Feb. 2009. eprint: https://direct.mit.edu/pvar/article-pdf/18/1/39/1624884/pres.18.1.39.pdf (cit. on p. 5).

[22] M. S. Moosavi, P. Raimbaud, C. Guillet, J. Plouzeau, and F. Merienne, "Weight perception analysis using pseudo-haptic feedback based on physical work evaluation," *Frontiers in Virtual Reality*, vol. 4, 2023 (cit. on pp. 5, 7).

[23] Y. Ujitoko and Y. Ban, "Survey of Pseudo-Haptics: Haptic Feedback Design and Application Proposals," *IEEE Transactions on Haptics*, vol. 14, no. 4, pp. 699–711, 2021 (cit. on p. 6).

[24] Y. Sato, T. Hiraki, N. Tanabe, H. Matsukura, D. Iwai, and K. Sato, "Modifying Texture Perception With Pseudo-Haptic Feedback for a Projected Virtual Hand Interface," *IEEE Access*, vol. 8, pp. 120 473–120 488, 2020 (cit. on pp. 6 sq.).

[25] J. Hosoi, Y. Ban, K. Ito, and S. Warisawa, "Pseudo-Wind Perception Induced by Cross-Modal Reproduction of Thermal, Vibrotactile, Visual, and Auditory Stimuli," *IEEE Access*, vol. 11, pp. 4781–4793, 2023 (cit. on p. 6).

[26] S. Kaneko, T. Yokosaka, H. Kajimoto, and T. Kawabe, "A Pseudo-Haptic Method Using Auditory Feedback: The Role of Delay, Frequency, and Loudness of Auditory Feedback in Response to a User's Button Click in Causing a Sensation of Heaviness," *IEEE Access*, vol. 10, pp. 50 008–50 022, 2022 (cit. on p. 6).

[27] F. Canadas-Quesada and A. Reyes-Lecuona, "Improvement of perceived stiffness using auditory stimuli in haptic virtual reality," in *MELECON 2006 - 2006 IEEE Mediterranean Electrotechnical Conference*, 2006, pp. 462–465 (cit. on p. 6).

[28] J. Kim, S. Kim, and J. Lee, “The Effect of Multisensory Pseudo-Haptic Feedback on Perception of Virtual Weight,” *IEEE Access*, vol. 10, pp. 5129–5140, 2022 (cit. on p. 6).

[29] M. Rietzler, G. Haas, T. Dreja, F. Geiselhart, and E. Rukzio, “Virtual Muscle Force: Communicating Kinesthetic Forces Through Pseudo-Haptic Feedback and Muscle Input,” in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, ser. UIST ’19, New Orleans, LA, USA: Association for Computing Machinery, 2019, pp. 913–922 (cit. on p. 6).

[30] M. Botvinick and J. Cohen, “Rubber hands ‘feel’ touch that eyes see,” *Nature*, vol. 391, no. 6669, pp. 756–756, Feb. 1998, Publisher: Nature Publishing Group (cit. on p. 6).

[31] F. Argelaguet, L. Hoyet, M. Trico, and A. Lecuyer, “The role of interaction in virtual embodiment: Effects of the virtual hand representation,” in *2016 IEEE Virtual Reality (VR)*, ISSN: 2375-5334, Mar. 2016, pp. 3–10 (cit. on p. 7).

[32] F. Argelaguet and C. Andujar, “A survey of 3D object selection techniques for virtual environments,” *Computers& Graphics*, vol. 37, no. 3, pp. 121–136, 2013 (cit. on p. 7).

[33] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa, “The go-go interaction technique: non-linear mapping for direct manipulation in VR,” in *Proceedings of the 9th annual ACM symposium on User interface software and technology*, 1996, pp. 79–80 (cit. on p. 7).

[34] J. Kim and J. Lee, “The Effect of the Virtual Object Size on Weight Perception Augmented with Pseudo-Haptic Feedback,” in *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2021, pp. 575–576 (cit. on p. 7).

[35] D. Eigen, C. Puhrsch, and R. Fergus, “Depth Map Prediction from a Single Image using a Multi-Scale Deep Network,” in *Advances in Neural Information Processing Systems*, vol. 27, Curran Associates, Inc., 2014 (cit. on p. 8).

[36] K. O’Shea and R. Nash, *An Introduction to Convolutional Neural Networks*, arXiv:1511.08458 [cs], Dec. 2015 (cit. on p. 8).

[37] Y. LeCun, B. Boser, J. S. Denker, *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989, Conference Name: Neural Computation (cit. on p. 8).

[38] D. Eigen and R. Fergus, "Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture," in *2015 IEEE International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Dec. 2015, pp. 2650–2658 (cit. on p. 8).

[39] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper Depth Prediction with Fully Convolutional Residual Networks," in *2016 Fourth International Conference on 3D Vision (3DV)*, Oct. 2016, pp. 239–248 (cit. on p. 8).

[40] R. Garg, V. K. B.G., G. Carneiro, and I. Reid, "Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2016, pp. 740–756 (cit. on p. 8).

[41] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 1063-6919, Jul. 2017, pp. 6602–6611 (cit. on p. 8).

[42] Y. Kuznietsov, J. Stückler, and B. Leibe, "Semi-Supervised Deep Learning for Monocular Depth Map Prediction," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 1063-6919, Jul. 2017, pp. 2215–2223 (cit. on p. 8).

[43] J. Li, R. Klein, and A. Yao, "A Two-Streamed Network for Estimating Fine-Scaled Depth Maps From Single RGB Images," 2017, pp. 3372–3380 (cit. on p. 9).

[44] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence (cit. on p. 9).

[45] N. U. Islam and J. Park, "Depth Estimation From a Single RGB Image Using Fine-Tuned Generative Adversarial Network," *IEEE Access*, vol. 9, pp. 32 781–32 794, 2021, Conference Name: IEEE Access (cit. on p. 9).

[46] N. Zioulis, A. Karakottas, D. Zarpalas, and P. Daras, "OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham: Springer International Publishing, 2018, pp. 453–471 (cit. on pp. 9 sq.).

[47] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "BiFuse: Monocular 360 Depth Estimation via Bi-Projection Fusion," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2020, pp. 459–468 (cit. on p. 10).

[48] H. Jiang, Z. Sheng, S. Zhu, Z. Dong, and R. Huang, "UniFuse: Unidirectional Fusion for 360° Panorama Depth Estimation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1519–1526, Apr. 2021, Conference Name: IEEE Robotics and Automation Letters (cit. on pp. 10 sq.).

[49] M. Rey-Area, M. Yuan, and C. Richardt, *360MonoDepth: High-Resolution 360° Monocular Depth Estimation*, arXiv:2111.15669 [cs], Mar. 2022 (cit. on pp. 10 sq., 14).

[50] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, *Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer*, 2020. arXiv: 1907.01341 [cs.CV] (cit. on pp. 11, 14).

[51] C.-H. Peng and J. Zhang, "High-Resolution Depth Estimation for 360deg Panoramas Through Perspective and Panoramic Depth Images Registration," 2023, pp. 3116–3125 (cit. on p. 11).

[52] W. Yin, J. Zhang, O. Wang, *et al.*, "Learning To Recover 3D Scene Shape From a Single Image," 2021, pp. 204–213 (cit. on p. 11).

[53] "Configuring Stereo Depth." (2024), [Online]. Available: https://docs.luxonis.com/hardware/platform/depth/configuring-stereo-depth (visited on 09/10/2024) (cit. on p. 11).

[54] "Unity 2022.3.4f1." (2024), [Online]. Available: https://unity.com/releases/editor/whats-new/2022.3.4 (visited on 08/15/2024) (cit. on p. 13).

[55] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, *ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth*, 2023. arXiv: 2302.12288 [cs.CV] (cit. on p. 14).

[56] "ImmersityAI." (2024), [Online]. Available: https://www.immersity.ai (visited on 08/13/2024) (cit. on p. 14).

[57] "Blender." (2024), [Online]. Available: https://www.blender.org (visited on 08/14/2024) (cit. on p. 16).

[58] "List of 20 Simple, Distinct Colors." (2024), [Online]. Available: https://sashamaps.net/docs/resources/20-colors (visited on 08/15/2024) (cit. on p. 23).

[59] "Meta Haptics Studio." (2024), [Online]. Available: https://developers.meta.com/horizon/resources/haptics-studio (visited on 09/19/2024) (cit. on p. 24).

[60] "OVR Metrics Tool." (2024), [Online]. Available: https://developers.meta.com/horizon/documentation/unity/ts-ovrmetricstool (visited on 09/19/2024) (cit. on p. 43).

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht.

Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Würzburg, September 23, 2024

Vanessa Pfeiffer

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Versicherung zur selbständigen Leistungserbringung

Titel der

Thema bereitgestellt von (Titel, Vorname, Nachname, Lehrstuhl):

Eingereicht durch (Vorname, Nachname, Matrikel):

Ich versichere, dass ich die vorstehende schriftliche Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die benutzte Literatur sowie sonstige Hilfsquellen sind vollständig angegeben. Wörtlich oder dem Sinne nach dem Schrifttum oder dem Internet entnommene Stellen sind unter Angabe der Quelle kenntlich gemacht.

Weitere Personen waren an der geistigen Leistung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich nicht die Hilfe eines Ghostwriters oder einer Ghostwriting-Agentur in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar Geld oder geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Arbeit stehen.

> Mit dem Prüfungsleiter bzw. der Prüfungsleiterin wurde abgestimmt, dass für die Erstellung der vorgelegten schriftlichen Arbeit Chatbots (insbesondere ChatGPT) bzw. allgemein solche Programme, die anstelle meiner Person die Aufgabenstellung der Prüfung bzw. Teile derselben bearbeiten könnten, entsprechend den Vorgaben der Prüfungsleiterin bzw. des Prüfungsleiters eingesetzt wurden. Die mittels Chatbots erstellten Passagen sind als solche gekennzeichnet.

Der Durchführung einer elektronischen Plagiatsprüfung stimme ich hiermit zu. Die eingereichte elektronische Fassung der Arbeit ist vollständig. Mir ist bewusst, dass nachträgliche Ergänzungen ausgeschlossen sind.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht. Ich bin mir bewusst, dass eine unwahre Erklärung zur Versicherung der selbstständigen Leistungserbringung rechtliche Folgen haben kann.

____
Ort, Datum, Unterschrift