University of California
Santa Barbara

Authoring and Experiencing Virtual 3D Environments

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Media Arts and Technology

by

Ehsan Sayyad

Committee in charge:

Professor Tobias Höllerer, Chair
Professor Pradeep Sen
Assistant Professor Jennifer Jacobs
Professor Marko Peljhan

March 2023

The Dissertation of Ehsan Sayyad is approved.

Professor Pradeep Sen

Assistant Professor Jennifer Jacobs

Professor Marko Peljhan

Professor Tobias Höllerer, Committee Chair

January 2023

Authoring and Experiencing Virtual 3D Environments

by

Ehsan Sayyad

Dedicated To

My beloved mother, whose love and support have been a constant source of inspiration and strength throughout my life.

Acknowledgements

First, I would like to thank my advisor, Prof. Tobias Höllerer, for his unwavering support, guidance, and encouragement throughout my Ph.D years. His insights and expertise have been invaluable for my research and career. He has consistently motivated me to follow my passion while keeping me grounded in the purpose and direction of my work. In addition to being an exceptional academic advisor, he is also a caring and compassionate mentor who always takes an interest in the personal lives of his students.

I want to thank my committee member Prof. Pradeep Sen, for his insightful encouragement and guidance throughout my studies and for constantly pushing me to strive for excellence. His knowledge and expertise have been instrumental in helping me to deepen my technical understanding.

I would also like to thank my committee members Prof. Jennifer Jacobs and Prof. Marko Peljhan, for their valuable expertise and guidance. Professor Jacobs has helped me to deepen my understanding of the complexities of design, while Professor Peljhan has provided me with an inspiring vision of what is possible in the field of Media Arts and Technology. I am also grateful to my collaborator, Prof. Misha Sra, for generously sharing her expertise in the field of Mixed Reality locomotion.

I am deeply grateful to all of my labmates at FourEyes Lab, who have been exceptional mentors and peers throughout my time with the lab. Special thanks go to Dr. Donghao Ren, Dr. Benjamin Nuernberger, and Dr. Domagoj Baričević, who provided guidance and support when I first joined the lab. I am also grateful to Yi Ding, Alex Rich, Noah Stier, You-Jin Kim, Brandon Huynh, CY Xu, Radha Kumaran, Sherry Chen and Daniel Lohn who have been wonderful colleagues and have contributed to my growth and learning in countless ways. Thank you all for your invaluable support and friendship.

I'm very thankful to the wonderful Media Arts and Technology community, whose

friendship and support have been a constant source of warmth and inspiration. I am grateful for the opportunity to be a part of such a talented and supportive group of individuals.

I am grateful for my friends Hojjat Aghakhani, Farnaz Kaboudvand, Farnood Merrikh Bayat, Sepideh Taheri, Fatemeh Jalali, Mehran Hoonejani, Salva Salmani Rezaie, and Kaveh Ahadi. They have been like a second family to me during my time abroad.

I want to thank the following funding agencies that supported the works in this dissertation: the Office of Naval Research under grant N00014-19-1-2553 and N00014-16-1-3002 and National Science Foundation (NSF) under grant 1911230.

Thank you to my fiancé, Sarah Payne, for her loving support and encouragement throughout my studies. She has always been there for me, providing me with the strength and inspiration I needed to succeed.

Curriculum Vitæ

Ehsan Sayyad

Education

2015 - 2023	Ph.D. in Media Arts and Technology, University of California, Santa Barbara
2012 - 2015	M.F.A. in Industrial Design, University of Tehran
2007 - 2012	B.S. in Chemistry, Sharif University of Technology

Research Experience

2016 - 2022	Graduate Student Researcher, Four Eyes Lab, University of California, Santa Barbara
Summer 2017	Software Research Engineer Intern, LogMeIn, Goleta, California
Summer 2018	Software Research Engineer Intern, LogMeIn, Goleta, California

Publications

1. Ehsan Sayyad, Pradeep Sen, and Tobias Höllerer. Panotrace: Interactive 3D modeling of surround-view panoramic images in virtual reality. *In Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology,*pages 1–10, 2017
2. Ehsan Sayyad, Misha Sra, and Tobias Höllerer. Walking and teleportation in wide-area virtual reality experiences. *In 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 608–617. IEEE, 2020
3. Pranav Acharya, Daniel Lohn, Vivian Ross, Maya Ha, Alexander Rich, Ehsan Sayyad, and Tobias Höllerer. *Using synthetic data generation to probe multi-view stereo networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision,* pages 1583–1591, 2021
4. You-Jin Kim, Radha Kumaran, Ehsan Sayyad, Anne Milner, Tom Bullock, Barry Giesbrecht, and Tobias Höllerer. Investigating search among physical and virtual objects under different lighting conditions. *IEEE Transactions on Visualization and Computer Graphics,* 28(11):3788–3798, 2022 vii

Abstract

Authoring and Experiencing Virtual 3D Environments

by

Ehsan Sayyad

The growing popularity of Virtual Reality and Augmented Reality has led to an increase in demand for 3D content. However, traditional methods for creating 3D environments are often expensive and complicated, making it difficult for non-expert users to produce their own content. This thesis addresses this issue through research focusing on two main areas: **(1) the study of 3D user experiences**, which helps create engaging and immersive experiences that meet users' needs and preferences and **(2) increasing accessibility for content generation**, which divides into creating *approachable 3D user interfaces* and *intelligent creative tools.* Our research aims to understand how users perceive and interact with immersive environments and, subsequently, to develop tools that enable users to create 3D content with ease and accessibility, even with limited experience or resources, with the help of generative models in machine learning and intuitive UI. Our project PanoTrace introduces a 3D modeling platform in VR for creating 3D scenes from 2D panoramas, providing a first example of our idea of novel *approachable 3D user interfaces.* Our wide-area VR walking study on the other hand falls entirely within the domain of studying 3D user experiences. It investigates the impact of natural walking versus teleportation on presence and user preference in VR experiences. We introduce the content-aware semantic editing and inpainting system (CASEIn) as an example of an *intelligent creative tool.* CASEIn generates high-quality results for guided image inpainting and semantic image synthesis using machine learning. Our projects DeepDive and Faded focus on accessibility for content generation by linking approachable UI design with machine learning. Faded is a memory reconstruction system that relies on our inpainting model, CASEIn, to extrapolate existing sparse information, including images and the user's memory of a space, into a cohesive 3D experience. In summary, this thesis demonstrates the feasibility and importance of approachable 3D UIs, the use of intelligent creative tools, and 3D user experience studies, all in service of enabling users to create 3D content in a more accessible way. Our research also provides insights into factors contributing to immersion, user preference, and viewing comfort in these settings.

Contents

Chapter 1

Introduction

1.1 Motivation

For most of our lives, humans have tried to make virtual worlds by creating stories. The human brain is a very powerful and creative content generator. It creates experiences based on what it has experienced before. Dreaming is the ultimate example of living in a virtual world without noticing it [1]. Let's call the experiences that the brain creates *soft virtual experiences*. The details are fuzzy, and they fade away. Moreover, the process relies heavily on the observer and their past. A novel that transpired into a virtual world through our brain might be very different from what someone else would create in their mind. That is something any reader experiences in the form of disappointment when a director creates a movie from their favorite novel. When we can simply close our eyes and use our imagination to create experiences and travel through time, why did we even start to solidify our imaginations and memories as art? The answer lies in sharing. These imaginations are entirely subjective and rely on personal experiences, which makes it challenging to explain unseen and new concepts through words. We also tend to abstract away the details. This also makes these creations often hard to remember. That is why we have been trying to capture reality in various forms from the earliest time in our history; to preserve and share them. We can see this urge to capture and preserve from the earliest paintings and relief sculptures to the invention of photography and the modern camera. With every invention in this domain came a broad library of story-telling abilities. Perspective, for example, was not discovered until the 1400s [2].

After the invention of the first motion film camera, it took a while for cinematic techniques to emerge. It is also worth noting that these media were not initially accessible to everyday users. Slowly, with the help of technology and mass production, people got access to them and started capturing their lives. Interactive 3D has been going through the same phase. We currently have access to some immersive 3D experiences through the work of large studios and game developers. However, personalized 3D experiences are yet to become a reality. With the apparent rise of AR and VR technologies, this is a future that will attract more focus and attention.

In recent years there have been significant advances in consumer augmented reality (AR) and virtual reality (VR) technologies. These advances have made it possible for people to experience 3D content in a more realistic and immersive way. The availability of affordable headsets such as Oculus Quest that can be used in both tethered and standalone modes has helped the viability of virtual reality. AR devices, such as the Microsoft HoloLens, MagicLeap, and Snap Spectacles, allow users to view 3D content in the real world, overlaid on their surroundings. These advances make it possible for people to experience 3D content in new and exciting ways.

Many prominent social media platforms currently focus on augmented reality technologies that are accessible through smartphones, mainly utilizing face tracking and video editing. Spark AR Studio by Meta [3], Lens Studio by Snap [4], and Effect House by Tiktok [5] are all in-house game engines that focus on creating augmented reality interfaces. These platforms focus mainly on creating social media effects and filters, preparing their user base for a potential mass release of AR devices.

1.1.1 Content Drives and Follows Technology

These advances create a new age for the consumption of 3D content. 3D content can be used for a variety of purposes, including entertainment, education, training, and product visualization. So far, most of the consumption of 3D content has been through offline rendering and bringing the 3D media into the 2D world. Real-time rendering is limited to applications in the entertainment and game industry. The restrictions mostly come from the fact that 3D content is still very expensive to create, and therefore, casual and day-to-day 3D content either does not exist or lacks quality.

Creative content has always been an area open to skilled users. From drawing to graphics design, you need to know basic design principles and how to use different software packages to produce high-quality content. The need for specialized knowledge gets even more emphasized when working in 3D, where software is usually more expensive and the learning curve to produce good quality content is even steeper.

However, the story is a little different in capturing everyday moments. Casual content in 2D, such as recording short video content with mobile phones, is an area of media creation that has disrupted the content scene [6]. Nowadays, the most popular type of content is consumer-quality, "honest" video recordings that are produced on the phone and posted as Tiktoks[1] or Reels[2].

The question is, what would be the equivalent of these contents for 3D?

[1] `https://www.tiktok.com/`.

[2] Reels are a type of video content on Instagram that were introduced to compete with TikTok. `https://about.instagram.com/blog/announcements/introducing-instagram-reels-announcement`.

1.1.2 Creative 3D content

Creative 3D content has been one of the first use cases of digital 3D content generation, mainly used for CGI and 3D video games. From the late 80s, when 3D shading and rendering had entered the entertainment industry, computer graphics software was being developed for artists. Since then, hundreds of applications would focus on enabling artists to convert concepts into 3D models. This rapid growth has led to the emergence of multiple categories of 3D content software, including but not limited to CAD, polygon modeling, sculpting, procedural modeling, and texturing. Each category usually focuses on solving specific problems well while still being capable at other tasks. Currently, Maya, 3ds Max, Cinema 4D, Fusion360, AutoCAD, Houdini, Zbrush, Blender, and many other software packages, are very popular and robust solutions to produce 3D content. However, they are all fairly complicated tools and have steep learning curves.

More casual 3D content generation tools have been introduced in recent years. Dreams by Media Molecule [7] is an excellent example of an approachable 3D modeling software, which is a game engine released as a video game for PlayStation. Virtual reality has specifically been a platform for these approachable UIs. Tilt Brush and Poly by Google, Gravity Sketch [8], Adobe Medium [9], and Substance 3D Modeler[10] are all 3D modeling applications that use VR very intuitively at different levels of professional ambition. In the mobile world, Nomad sculpt has a very intuitive tablet interface letting users create complex sculptures using a stylus.

These applications and associated user interfaces primarily focus on the final visual product rather than the technical aspects of future usability in interactive scenarios. In creative scenarios, they fall under concept-art creation tools, and the resulting 3D model is usually not optimized to be used directly in production. However, additional advances in 3D content creation tools have allowed artists to quickly convert 3D concepts to usable assets. To name a few examples, tools that automatically optimize the topology of 3D meshes such as instaLOD [11] or 3DCoat [12], and technologies such as Nanite [13] for Unreal engine that let modern game engines handle hundreds of millions of polygons in a scene.

1.1.3 Capturing 3D content

Capturing 3D information from 2D observation dates back to 1867 and the introduction of photogrammetry for archaeological and geographical purposes [14]. With advances in computing, computational methods such as structure-from-motion (SfM) [15] technique rapidly grew. Nowadays, there are many ways to capture 3D scenes, including lidar scanners, depth sensors, multiview rigs and modeling techniques that use only monoscopic cameras for example by employing neural representations [16]. Just like many other fields, each method comes with specific pros and cons. While advanced lidar devices are remarkably fast and accurate, they usually come with a very high price tag. SFM and photogrammetry require expensive software, lots of computation, and lots of data. Recently, the push for accessible augmented and virtual reality has provided better and cheaper systems that enable end users to capture real environments more efficiently. Most high-end smartphones come equipped with depth sensors capable of scanning indoor environments. This goes hand in hand with the growing demand for creating a virtual replica of the real world or, as the industry calls it, the "Digital Twin [17]." Although companies like Apple and Google have constantly been scanning and recreating streets in large cities, practical augmented reality would not be feasible without the ability to scan and store all environments.

1.1.4 From 2D to 3D

There is a lot of content originally created in a 2D format that cannot be experienced in an immersive way in 3D, such as family photos and videos. This is a problem because it means that people cannot take advantage of the benefits of 3D experiences, such as immersiveness and interactivity. One way to address this problem is to create new 3D versions of existing 2D content. This can be done using specialized 3D modeling software and leveraging the expertise of people familiar with these tools. However, it would be very resource intensive and require great effort.

Creating new 3D versions of existing 2D content is a significant challenge. Most 3D content creation software packages are designed to create content from scratch. Although there have been attempts at creating image-based modeling tools, the process usually has a steep learning curve. One way to address this problem is to explore the creation of innovative yet simplified 3D modeling software specifically designed for this use case. In doing so, it is essential to study the main elements that contribute to a more immersive experience and to try to incorporate these elements using simplified and intelligent interfaces. A simplified interface is a key to making such a platform more accessible to everyone. This would open the door to a wide variety of content that would otherwise be considered obsolete.

Simplified interfaces should focus on natural interaction design. Natural interaction is important because it allows people to use their common knowledge of the world and interactive metaphors while creating content [18]. For example, sculpting using virtual clay could be second nature, whereas creating a low-poly 3D model in Maya would be unlike anything a user generally does on a daily basis. In addition, simplified interfaces should remove clutter and unnecessary features. This will help people focus on the task and not be distracted by irrelevant information. Virtual and augmented reality are great modalities to implement these natural interfaces in since the interactions in these modalities are more aligned with our experience with the real world.

1.1.5 Approachable user interfaces

Performing complicated tasks usually requires sophisticated tools. Due to its traditional use cases, 3D content generation usually falls under enterprise standards and needs, which will require particular tooling. For example, 3D models generated for a video game require a specific level of optimization that reflects a variable level of detail in model geometry and topology. They might need to be compatible with an engine-specific shading model or a specialized kind of rigging. All these complications are in the common toolset of a professional creator when approaching 3D content. However, these requirements leave casual creators in the dust.

Optimizing and working with hardware limitations is an art that turns obsolete with advances in hardware and computing power. The history of video games is a graveyard of optimization techniques such as bit-manipulation, dithering, and fixed-point math approximations that are just not labor-efficient to do in most cases. It would be a waste of talent to focus on every single CPU clock cycle in assembly, and these tasks are usually handed to a sufficiently efficient compiler since computing power is growing exponentially. Most video games from the early 1990s would take just a few megabytes of data while creating the same level of complexity with modern game engines will require hundreds of megabytes. On the other hand, a video game that once required a complete team of professionals to make could now be made by a single person using a game engine.

We argue that with the ever-growing need for 3D content, these user interfaces need to go through the same metamorphosis that creative desktop tools went through when being released for tablets and smartphones. However, in this case, we have a two-sided requirement. Not only do XR interfaces come with their specific upsides and downsides, but with a new use case definition and more powerful platforms, we would have a vastly different set of content quality requirements. For example, we will not need to worry about a highly optimized riggable model when capturing 3D memories using a smartphone if the goal is to only capture a frozen moment in time.

We define *Approachable User Interfaces* as interfaces that are easy to use and allow users to accomplish complicated tasks by removing clutter and focusing on critical elements. These interfaces should rely on familiar affordances and focus on maximizing a tool's effectiveness with minimal effort. Approachable interfaces should be second nature to users, require minimum training and be fun to use. They also need to focus on the essential goal a user is trying to accomplish by hiding clutter and separating out extra tools for advanced users. These interfaces can use algorithmic automation and artificial intelligence to perform the tedious tasks necessary to achieve a final goal while keeping the user in control of the application flow.

1.1.6 Why user behavior matters

While designing Approachable User Interfaces, it is crucial to study user experiences. Doing so helps make informed decisions about what features to keep in the interface and what to focus on to achieve the desired goals. Our primary goal in this body of work is to create 3D authoring tools that focus on approachable UI. To this end, we partially dedicate our efforts to studying user experiences in 3D environments. Throughout multiple projects, we investigate the effect of stereoscopic depth, degrees of freedom for viewing (3DoF vs. 6DoF), locomotion techniques, and lighting on immersiveness, task performance, and XR side effects like simulation sickness.

1.1.7 AI for simplifying complex tasks

AI has been playing a big role in automation throughout the industry. In robotics, aviation, and autonomous vehicles, AI has been proven an effective and practical tool. Content generation specifically has also benefited from AI and automated workflow. Traditionally, the role of AI and automation in content generation has been focused on specialized tools. Tools such as the wand selection in Adobe Photoshop or Z-remesher in Zbrush, are examples of these partially automated algorithms that help content creators avoid extremely repetitive tasks. However, there has been a serious limitation in using traditional AI in more holistic content creation. This story, however, has begun to change with the recent advances in deep learning. Problems that used to be only possible to be solved by humans are now being tackled by AI, and in some cases, results are more accurate and efficient.

1.1.8 AI for content generation

3D reconstruction in computer vision and graphics is recreating geometry and light in a virtual environment as it was in the real world. Algorithms such as Structure from Motion (SfM) and instruments such as lidar scanners and depth cameras perform the reconstruction task by gathering reliable information about an environment and converging to a single correct solution. In cases of missing data or incoherent inputs, these systems simply won't produce anything useful. However, if the goal of 3D reconstruction is to recreate the experience for a human user, that change of focus allows an algorithm to converge on a plausible output by "hallucinating" the results based on prior observations.

The field of generative models has constantly been growing alongside other areas of machine learning. Generative adversarial networks (GANs) started with lots of limitations, and today, models like BigGAN [19] or StyleGAN2 [20] can create high-resolution images that are indistinguishable from real images. Diffusion models [21], are the most recent architecture for generative models and have been proven even more effective in metrics such as FID [22] score.

More recently, CLIP-based, text-to-image generation systems like DALL.E [23] Imagen [24] or Stable Diffusion [25] have made a huge impact and are already being used in commercial applications like Midjourney. Continuous research in that domain has shown very effective results in text-to-video [26] and text-to-3D [27] as well. It is clear that the future of content generation will heavily rely on machine learning models.

1.2 Contributions

In this dissertation, through multiple projects, we investigate various ways novel approachable user interfaces could be utilized for generating and experiencing 3D content. Throughout the thesis we design and measure 3D user interfaces and study various elements that affect user performance while experiencing 3D content in AR and VR.

In Chapter 2, we introduce PanoTrace, a 3D image-based modeling tool in VR that is designed with natural 3D interactions. We propose a novel modeling tool that allows users to easily add 3D geometry to panorama images, which can significantly increase the immersion a viewer experiences when viewing the panorama in virtual reality. Our results indicate that our modeled scenes produce a significantly higher self-rated sense of immersion than a basic dome geometry for the panorama when viewed in VR with head orientation and position tracking.

In Chapter 3, we investigate the effects of two main forms of locomotion techniques in virtual reality on immersion, simulation sickness, and sense of direction through one of the first wide-area studies in virtual reality involving real walking. We introduce a new method of mixed reality self-redirection that lets users explore a large environment

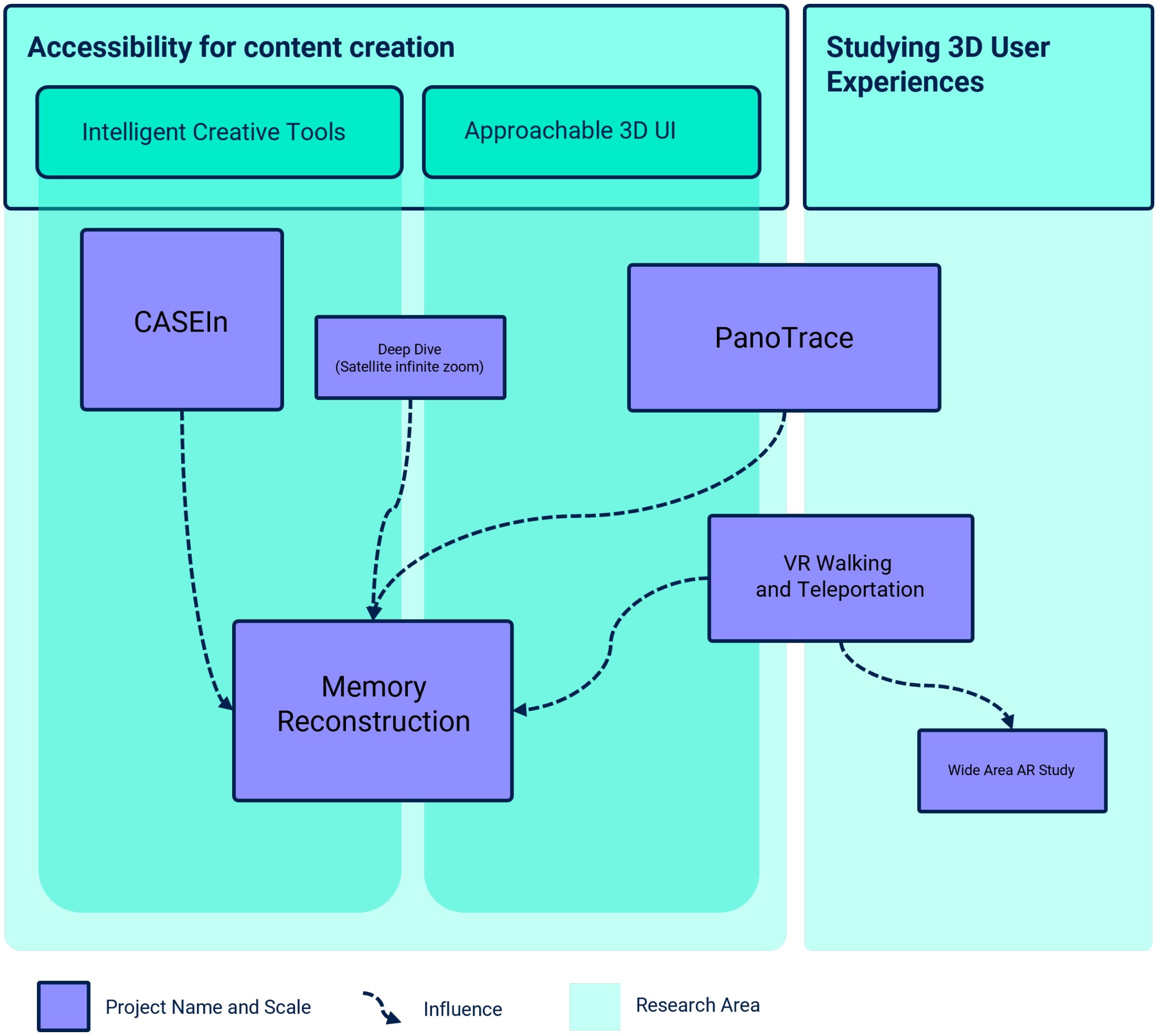


Figure 1.1: Overall map of contributions of this dissertation in relation to broader research areas.

using natural walking. Using the same research platform we developed for this project, we further investigate the effects of lighting conditions on task performance in wide-area augmented reality while walking.

In Chapter 4, we explore the space of machine learning (ML) for content generation. We discuss the existing generative methods in ML for 2D and 3D content. We introduce DeepDive, a GAN-based 3D application that connects a 3D VR interface into a GAN to let the user dive into hallucinated sequences of satellite images. We also introduce CASEIn, our state of art Content Aware Semantic Editing and Inpainting network. We investigate and compare the results and use cases of this architecture.

In Chapter 5, We introduce Faded, our memory reconstruction system that incorporates various ML models and approachable user interfaces to let the user create a 3D scene from limited existing 2D images of a place or memory. We demonstrate the elements, introduce the ML models and showcase results that were created using the system.

Our work in this dissertation encompasses two primary domains of studying 3D user experiences and accessibility for content generation. You can see Figure 1.1 for a detailed visualization of our contributions in these domains. Within the accessibility for content generation, we define two separate subdomains, designing intelligent creative tools and designing approachable user interfaces.

Our first work, PanoTrace, touches both of these areas by focusing on designing a minimal and intuitive 3D UI for 3D content generation and a study on user experience with different levels of 3D depth and detail. It provides insight into the effects of viewing degrees of freedom and stereoscopic depth in a panorama on the user's sense of immersion, self-reported presence, and simulator sickness. Our work in Chapter 3, labeled "VR Walking and Teleportation" in Figure 1.1, explores various forms of locomotion in 3D spaces and their effects on user understanding of a 3D space. The framework implemented in this work heavily influenced the "Wide Area AR Study [28]," which falls within the area of studying 3D user experiences.

While PanoTrace briefly investigates smart elements in a creative system, it leaves most of that domain unexplored. We extensively explore intelligent, creative tools in Chapter 4 by initially introducing DeepDive, our first attempt to link an intuitive VR interface with a machine learning algorithm.

Elements of DeepDive are later used in our project Faded, which we introduce in Chapter 5. Faded is more complicated in design and has more moving parts, with the central feature being the machine learning model CASEIn. CASEIn is a contribution solely in the area of deep generative models. Figure 1.1 shows where each work is getting inspiration and influence from other projects, with Faded being our latest work and therefore inheriting the ML model from CASEIn, Networking infrastructure from DeepDive, 3D UI design from PanoTrace and teleportation from VR Walking and Teleportation.

Our work impacts various types of visual media across multiple domains, including creation accessibility, experience accessibility, and immersiveness. In Figure 1.2, we show these areas in a 3D chart creating a perceptual map of the space of visual media.

Our works, Faded and PanoTrace, focus on moving from existing 2D data into a more immersive 3D form. By enabling users to add extra depth values and extrapolate an existing 2D image in a 3D environment, we positively move 2D panoramas and digital photos in the domain of creation accessibility and immersiveness.

Extrapolation of 3D scans and 2D images into a unified environment in our work Faded improves the creation accessibility of 3D scans. In our Walking and Teleportation in wide-area VR study, we identify the benefits of various methods of locomotion while exploring 3D environments, which helps improve the immersiveness of 3D scans.

Both PanoTrace and Faded also contribute to the creation accessibility of interactive 3D content by providing innovative 3D user interfaces for 3D modeling.

Overall, this dissertation tackles authoring and experiencing virtual 3D environments

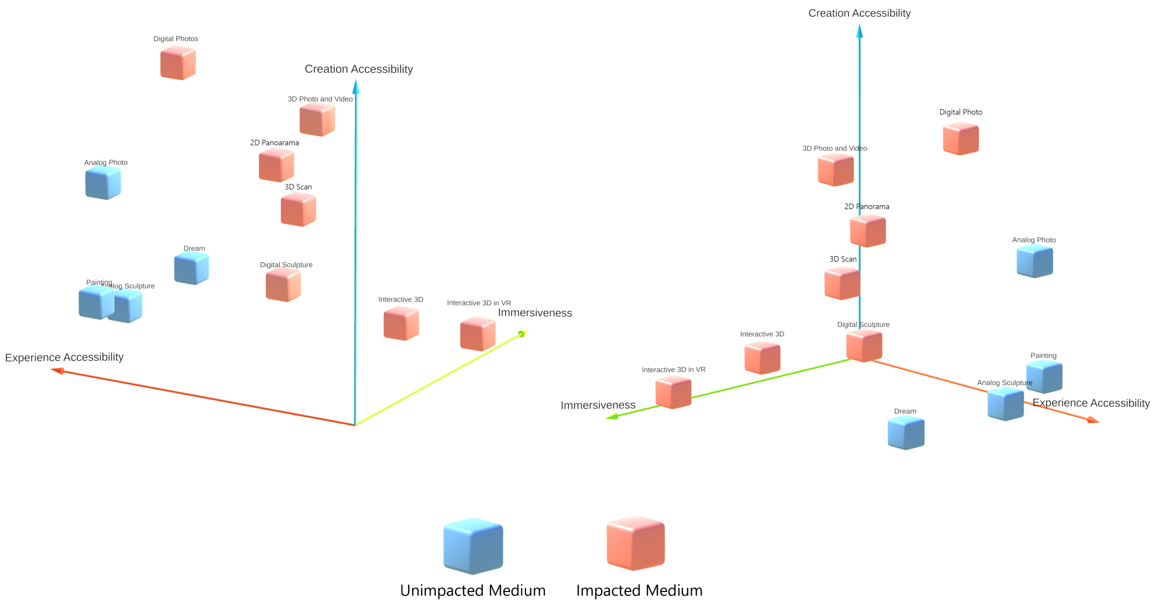


Figure 1.2: Perceptual map of different visual media color-coded based on whether they are affected by this thesis posed based on their relative location in three domains of creation accessibility, Experience accessibility, and immersiveness.

by studying user experiences, designing novel, approachable user interfaces, and building intelligent creative systems. Through various projects, it demonstrates answers to the ever-growing need to create 3D content by non-professional users. It provides insight into effective design decisions for these systems by studying user behavior in 3D experiences while creating these tools. It introduces novel user interfaces to generate 3D content. And lastly, it showcases the potential of machine learning and generative models in combination with approachable UI in empowering everyday users to create 3D content.

Chapter 2

PanoTrace: Interactive 3D Modeling of 360 Panoramic Images in Virtual Reality[1]

2.1 Introduction

Panoramic images are ubiquitous today. The desire to capture and share visual experiences has led to the development of many kinds of imaging techniques and devices for the acquisition of visual realities, and surround-scene panoramic images in particular, capture the experience of being "present" in a scene much better than just single points of view.

The capture of panoramas can either be done by taking multiple images with a standard, hand-held camera and then stitching them together with software techniques [30,

[1] The contents of this chapter have been previously published in VRST '17: Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology © 2017 ACM. Reprinted, with permission from Ehsan Sayyad, Tobias Höllerer, and Pradeep Sen, PanoTrace: interactive 3D modeling of surround-view panoramic images in virtual reality [29]

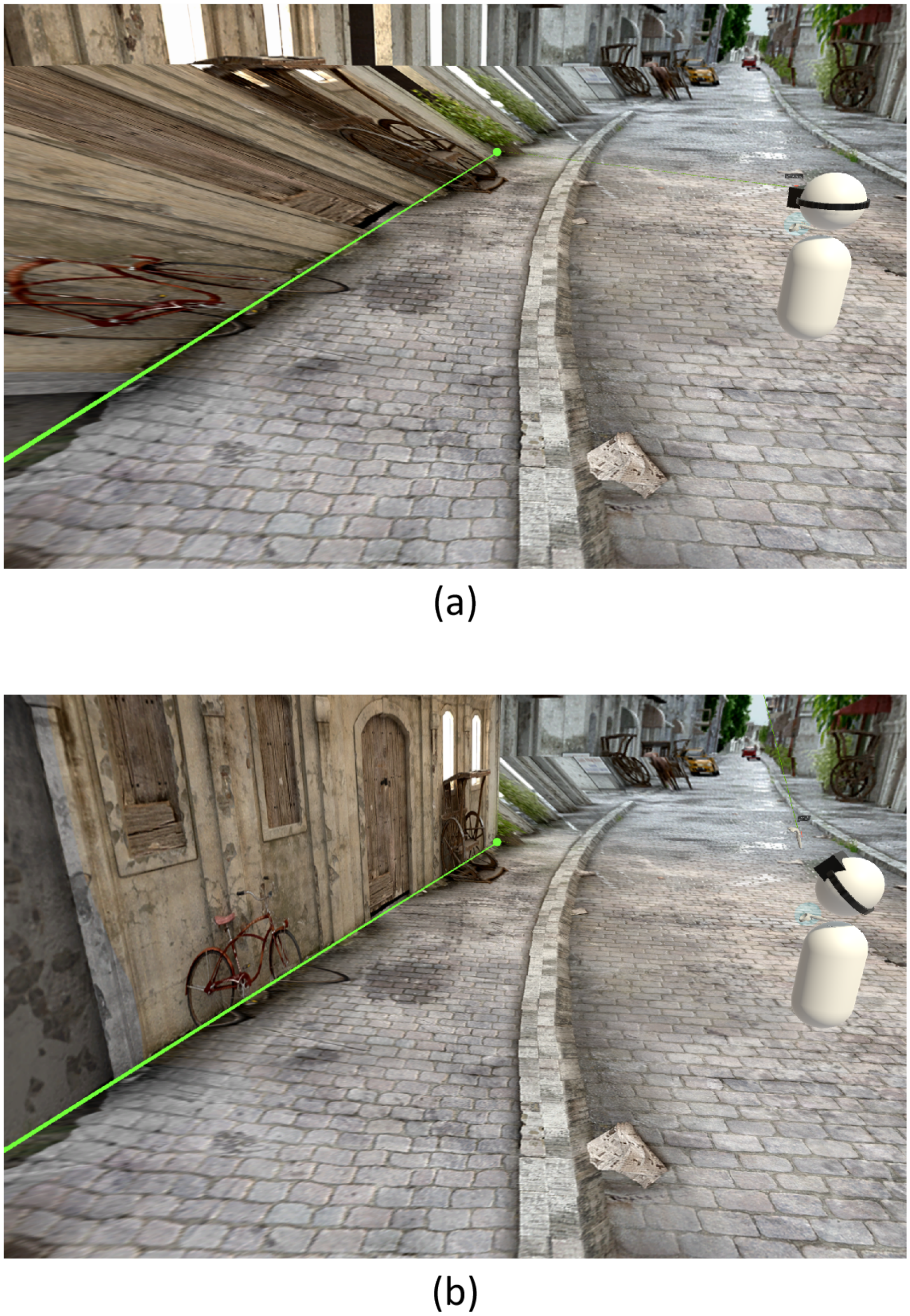


Figure 2.1: Example modeling action while experiencing a panorama in 6DoF-tracked VR. We see the user represented as an avatar from the side, a view that is only used for illustration purposes. The user is presented with a view corresponding to the avatar's head pose. The user has just placed a simple ground plane and now observes the panorama projected onto this plane. (a) User traces a wall line on the reference plane; (b) the User has extruded the wall.

31, 32], or directly with specialized panoramic cameras [33, 34, 35, 36, 37]. Furthermore, many tools and interfaces have been developed to explore and navigate image panoramas, such as QuickTime VR [38], YouTube Virtual Reality [39], Cardboard Camera VR [40], or Facebook 360 [41]. The sense of presence that panorama images provide along with the abundance of capture and visualization tools for them, has resulted in a plethora of online repositories for panoramic images produced by users all around the world.

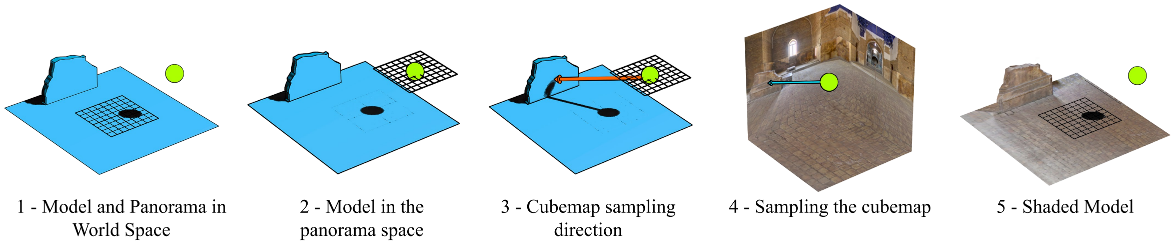

1 - Model and Panorama in World Space
2 - Model in the panorama space
3 - Cubemap sampling direction
4 - Sampling the cubemap
5 - Shaded Model

Figure 2.2: This image shows how the projection mapping works in the shader.

Recently, there has also been a flurry of renewed interest in immersive display technologies such as virtual reality (VR), thanks to the introduction of mass-market systems such as the Oculus Rift, HTC Vive, and Sony PlayStation VR. However, despite their full-surround nature and the sense of presence they provide, 2D panoramas have serious limitations when experienced in VR. 2D panoramic imagery simply lacks geometrical depth information necessary for viewing with binocular disparity (for stereo rendering) or for navigation of the scene freely in six degrees of freedom, orientation, and position (for motion parallax). When viewing a 2D panorama using VR, the user gets the feeling they are trapped, unable to move inside of a large "bubble" texture-mapped with the panorama.

This problem can obviously be addressed by augmenting the panorama with geometrical data. The idea of augmenting images with geometry to enable free viewing, or providing a sufficient number of alternative viewpoints to enable light field perception, goes back to the beginnings of image-based rendering [42, 43, 44] Later, researchers also explored the specific problem of generating geometry for panoramic images. For example, Oh et al. [45] showed how image-based modeling for panoramic images would result in an aesthetically pleasing mock-up of the environment. Unfortunately, many of the methods for augmenting panoramas with geometry usually need expensive or complicated setups at acquisition time [46, 47].

Other approaches require capturing panoramas at multiple locations in order to employ structure-from-motion techniques to reconstruct the environment [48].There are also ways to capture depth information by capturing multiple images and using optical flow [47, 49]. This often involves special-purpose and often expensive camera rigs and setups, and in the end, the geometry produced is often affected by noise. We are interested in the problem of generating depth information, or, even better, full 3D geometry, for 2D panoramas that have already been captured with simpler imaging solutions. There have been some great automatic approaches to generate 3D model information from a single image or a small set of images; Some have been successful in generating a 3D context for a single panorama using a well-trained Support Vector Machine [50]. This would suit specific kinds of environments but can be unsuccessful with a broader range of scenarios as you need to train the system using ground truth depth information and manual object annotation.

Finally, it is possible to manually model the scene using traditional 3D modeling tools. However, these tools are usually complex and require a skill set and a deep understanding of 3D environments. This makes it difficult for novice users to complete the modeling task.

We observe that VR itself could be a better environment for creating 3D models of 2D panoramas. After all, since VR is a natural medium for *viewing* immersive scenes, we hypothesize that it would also be the natural medium for *creating* immersive scene content. Therefore, in this work, we designed and implemented a complete toolset for modeling panorama geometry directly in VR. We use projection mapping as a way of tracing 3D geometry to make it easier for novice users to create complex 3D environments.

We designed a user study to examine the effect of augmenting panoramic images with geometrical models on a user's sense of immersion, realism, and discomfort. We compared the geometry created with our tool against other approaches by exposing users to panoramas rendered with various forms of underlying geometry including no geometry (infinitely-large sphere, which is what the default panorama would be), a simple hemispherical dome, our modeled scenes with the novel VR toolset, and the ground-truth geometry available from rendered scenes and Matterport [51] captures. Our results show that in cases with man-made environments, our modeled scenes had a significantly higher sense of immersion and realism than basic geometries such as the dome. They also tended to cause lower discomfort for the user in those cases.

The study results demonstrate that viewing a 2D panorama (with additional model geometry) using 6DoF (orientation and position) head tracking can significantly increase the feeling of immersion a viewer experiences. Using our toolset, novice users can model, in as little as 20 minutes, simple scene geometry that leads to a superior 6DoF viewing experience compared to projection onto a skydome and is about halfway as effective in terms of perceived immersion and scene realism as ground truth geometry models.

Previous work

Image based modeling Debevec [42] showed how by using a sparse set of images and a calibrated camera, some basic geometries could be constructed to match the image and generate the shape of buildings. Horry [52] presented a way to add depth to a single image based on indicating the vanishing points. Criminisi [53] later showed how by tracing parallel lines in an image, one could calculate the camera position and reconstruct the 3D model of the image. Zhang et al. [54] demonstrated a way to generate the 3D model of the scene from a single image based on a sparse set of user-specified constraints on the scene. Van den Hengel et al. [55] introduced a system to use images in a video as a reference to create complex 3D geometry while tracking camera movement. While these influential works all present methods to interactively create 3D geometry from 2D imagery, none of these operate specifically on surround-view panoramas, and none of these methods utilize interactive VR technologies for modeling.

3D modeling in VR There have been several early works regarding 3D modeling in VR [56, 57, 58, 59] and two-handed interaction in virtual environments [60, 61]. More recently, Jackson [62] generated a creative toolset that lets the user trace curves from images to create 3D objects. There is also a number of recently commercially available 3D modeling tools for VR [63, 64]. However, to our knowledge, no one has yet addressed the problem of modeling geometry in existing 2D image panoramas using VR tools.

Contributions

We posit that a 3D modeling interface in VR can be a very effective option to manually augment 2D panoramas with depth information. In order to create a useful toolset for novice users, we implemented some novel interactions and features (such as our 3D texture snapping and some 3D interactions as parts of our bi-manual transformation interface). We also performed a user study, and the analysis of the results revealed valuable information regarding the effect of depth information on a user's sense of immersion, realism, and discomfort.

2.2 System Overview

In this project, we have developed a complete interactive system to model panoramas in virtual reality. In the subsections that follow, we shall describe the different components of our system.

2.2.1 Architecture

The system is implemented using the Unity game engine, selected for its flexibility and general adoption. We have used SteamVR and the "OpenCV for Unity" plugins to control the HTC Vive device and use OpenCV functionality in the Unity environment. The program has been implemented in C# based on the existing Unity classes and data structures.

Data structure

We used equirectangular Panorama image files as the input. These files are being converted to OpenGL/ DirectX cubemap textures in Unity. The geometry is also being handled by Unity's Mesh class, which contains Vertex, Normal, UV, and Triangle arrays. In our rendering system, UV and Normal arrays are not being used.

Rendering

All the model geometries are rendered using a cubemap projection shader. A Projector object sends the panorama's transformation matrix $T \times R \times S$ to the shader. In the vertex stage, we multiply each vertex's world position by the panorama matrix to find the relative vertex position. In the fragment stage, we look up each fragment's color by sampling the cubemap using the normalized relative fragment position (see Figure 2.2). Shading all the geometries with existing panorama data causes images to duplicate on

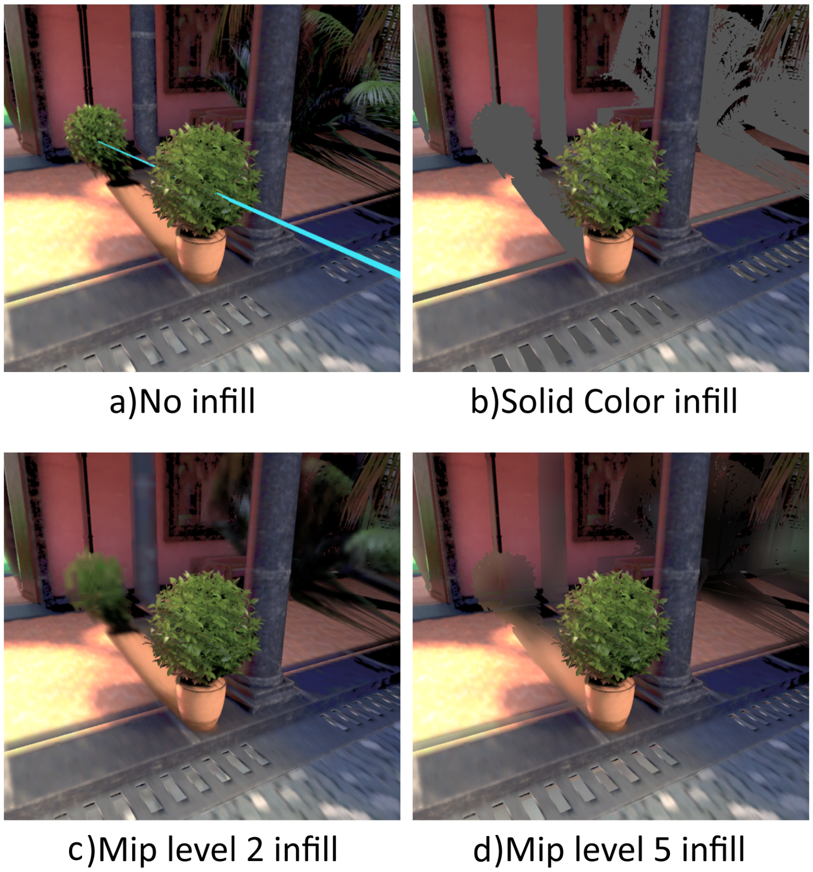


Figure 2.3: Texture inpainting: The blue ray indicates the direction from the projection center. Note how the colors are being repeated on all the surfaces.

the occluded surfaces (see Figure 2.3). This may cause confusion for the user. In order to diminish this effect, we implemented a shadow map algorithm that treats the projector as an omnidirectional light source and avoids shading the occluded fragments. We experimented with different ways of in-painting, i.e., replacing the occluded pixels, and provided a MIP mapping level control, giving the user an adjustment slider to choose a level that fits the scene.

User interface

The user input is handled by a SteamVR plugin. We used an HTC Vive with six-degree-of-freedom, room-scale tracking. It consists of a head mounted display and two controllers. A layer of user input handling was implemented on top of the raw input from SteamVR. The aim of the UI design was to create a 3D modeling interface for novice users. We focused on avoiding complexity in order to flatten the learning curve for the toolset. We developed a dynamic Pie menu system (see Figure 3.3) to benefit from the touch interface of the Vive controllers. We were able to select from up to 10 radial menu items effectively. However, for our final design, we limited the menu items to 5. Each menu item could trigger an action or lead to a sub-menu. Users can explore the menu system linearly (they can press the back button to return to the parent menu). Using the HTC Vive tracking functionality, users can move controllers and walk in the space. We also implemented basic 3D user interactions in the form of aiming and selecting, grabbing, shaking, and bimanual interactions.

2.2.2 Modeling

In order to model a panorama that corresponds to the ground-truth depth, first the transformation of the panoramic camera should be known. Unlike normal photographs, panoramas are by default considered to be equivalent to 6 pinpoint cameras with 90° field of view. Therefore, there is no need to calibrate for lens distortion. However, if a panorama is not aligned correctly, the importer needs to know the quaternion representing the panoramic camera's orientation. Users can use existing techniques to realign panoramas before introducing them to the system. A panorama's height can be adjusted while inserting a reference plane. The rest of the modeling, like finding the location of the walls, will proceed based on the visual feedback on the reference plane (see Figure 2.1).

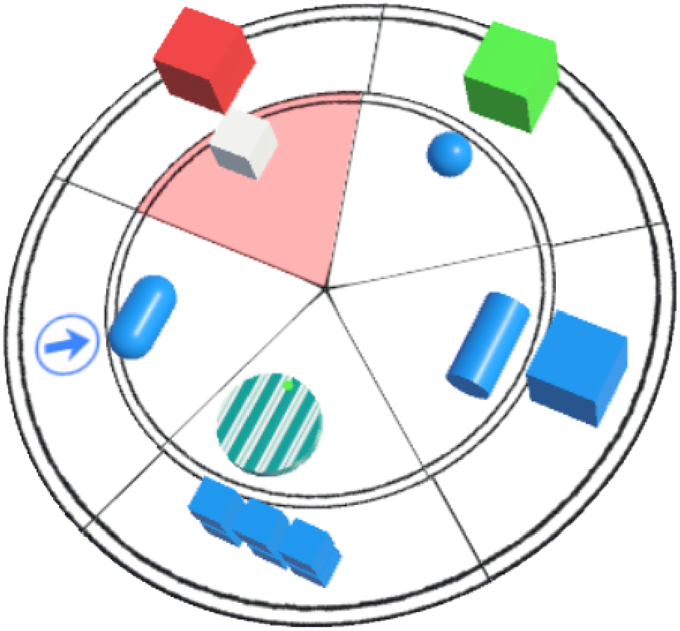


Figure 2.4: Our pie menu with some 3D geometry to showcase the functionality of each control. The menu appears above the touchpad on the controller.

Traditional modeling tools

3D brush tool: the brush tool draws geometry in a freehand style. This gives the user the ability to model more complex and organic shapes (see part (d) in Figure 2.5)

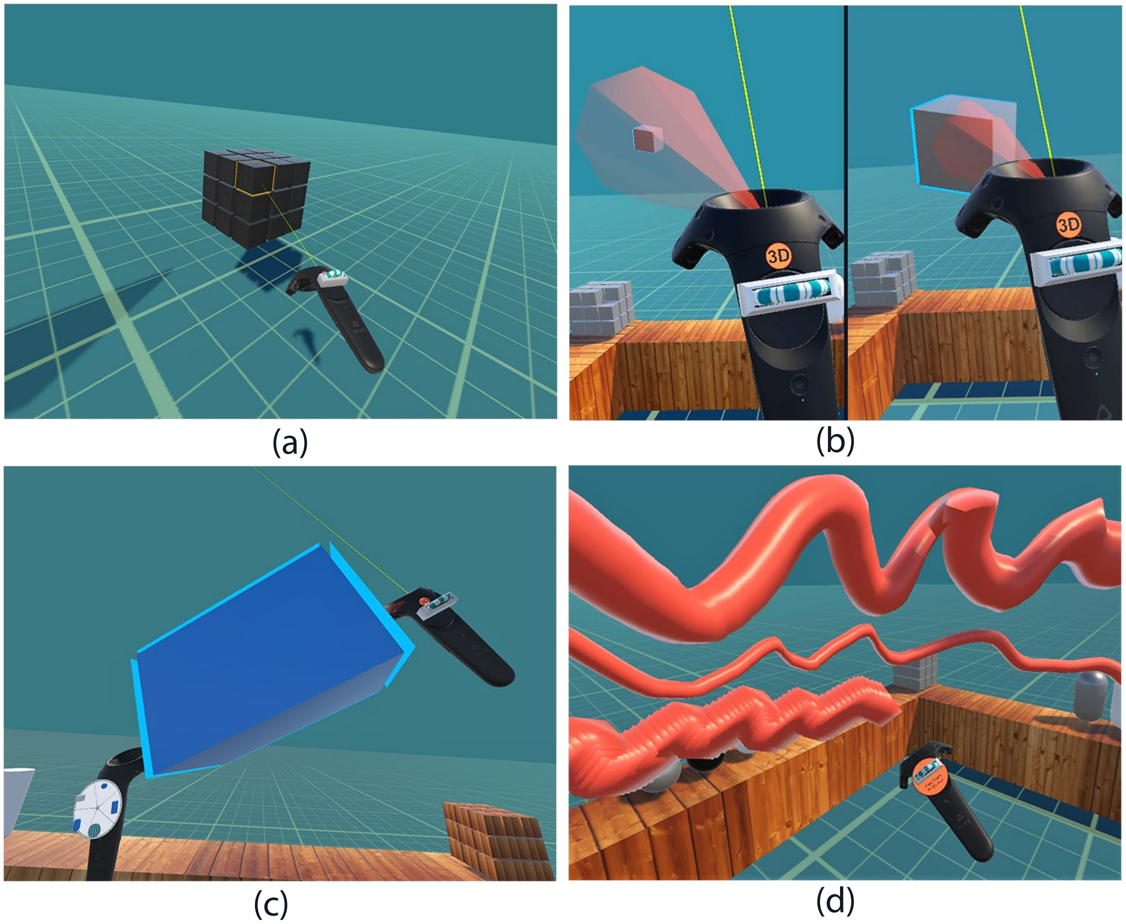
3D bimanual transformation tool: Users can point to and select objects in 3D. Objects can be moved and rotated with one hand. We apply the selecting controller's transformation to the object that we are displacing. Users can also move, rotate and scale the object using two controllers. We generate a transformation matrix that rotates, moves, and scales based on the position and rotation of the two controllers in each frame. This results in a widget-less direct manipulation tool that is easy to learn for novice users. A duplicate button is also placed on the controller to make a copy of the created shapes (see parts (a),(b),(c) in Figure 2.5)

Extrusion and vertex editing tool: The Extrusion tool works by creating a ground plane and extruding it to make walls. A vertex editing tool is provided to adjust the vertices to their correct locations. As the user points the tool to surface geometry, we cast a ray and find the contact point. The user can choose multiple points to create a polygon and extrude the polygon using the controller (see Figure 2.1)

Navigation: Our system offers different ways of navigation. The first one is the natural six-degree-of-freedom movement that the VR tracking provides. Users can walk and turn as long as they do not leave the Vive's tracking area. We also give the user the ability to fly around using a fly button. This is useful for users that do not experience motion sickness in VR. We also provide a World in Miniature [65] experience, which gives the user the ability to resize themselves to have better access to areas that are otherwise hard to reach or to change their precision by focusing on specific points.

Novel modeling tools

Panoramic image snapping: We implemented an image snapping algorithm [66] to help the user snap the pointer to points of interest on the panorama. We first create a Canny Edge cubemap for the panorama that we are modeling. Then, in each frame, we cast a ray to the geometry and find the contact point. Using the direction from the projection center to the contact point, we find the corresponding cubemap face and pixel. We apply the technique to find the snapping point. Based on the snapping point and the cubemap face, we regenerate a new ray as the snapping pointer ray. Canny edge detection could produce some undesired lines that do not represent change in geometry (such as shadow lines). However user can disable the snapping on the areas with large amount of false edge detection. (see Figure 3.9).



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

Figure 2.5: (a) Object selection with a laser pointer. (b) Bringing objects closer using a wheel. (c) Freeform bi-manual transformation of a cube. (d) Brush tool with tube and cube strokes with varying brush sizes.

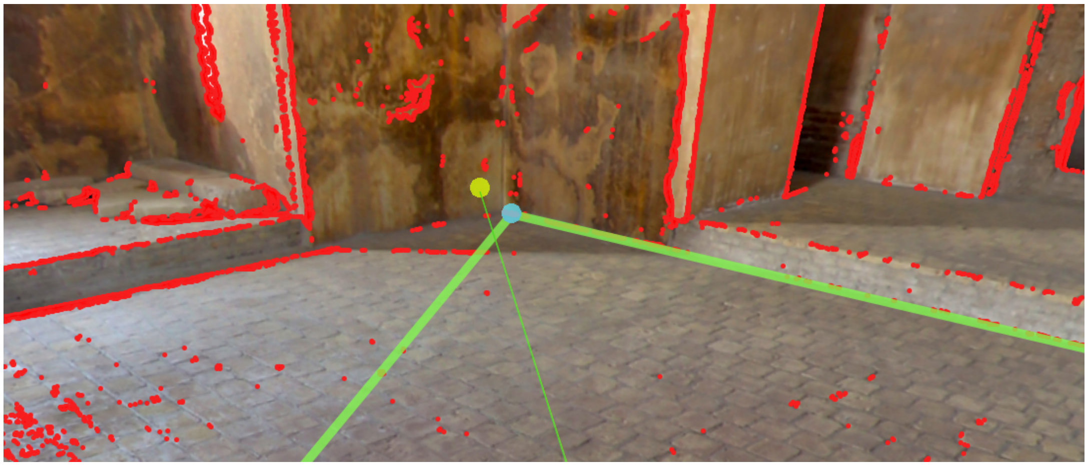
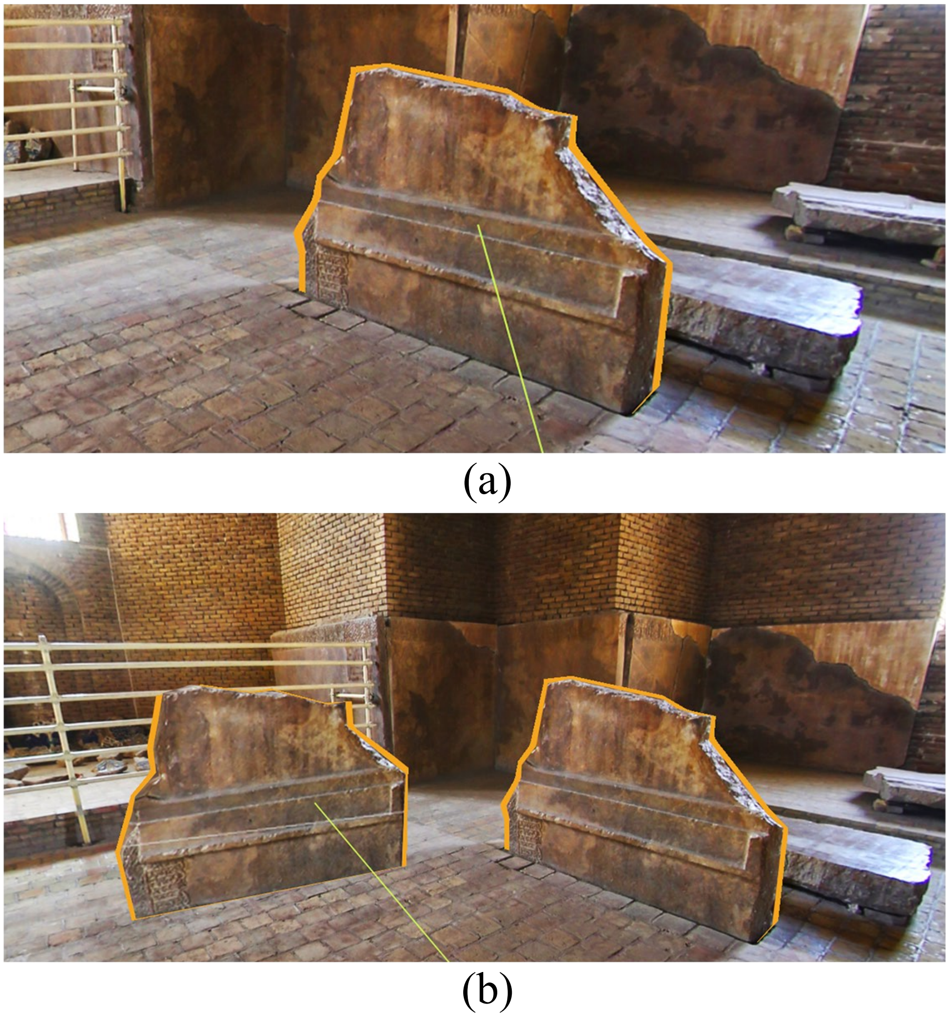


Figure 2.6: Red pixels showing the Canny edges. Laser pointer snaps to these pixels to help the user aim.

Texture baking and object duplication: A realtime-shader-based texture baking procedure was developed to support object duplication functionality. We generate an unwrapped UV set for the object and render it to a separate buffer using UV coordinates as vertex positions. Then we save the buffer as a texture and assign it to an unlit textured shader and render the object using that shader. Baked textures will be undistorted if they are projected on the correct corresponding 3D geometry. This makes it possible to extract textures for later use directly from the panorama (see Figure 3.12).

Depth information refinement: Depth data calculated from structure-from-motion or stereo-reconstruction techniques tend to be very noisy. We demonstrate a use case for our modeling tool as a refining and healing tool for existing depth information. This could also benefit cases of missing geometric information; We applied our tool to stereo depth-map panoramas from the Nokia OZO camera [37]. This high-end stereo real-time panoramic video camera specifically targets VR but it does not provide perfect stereo imagery: for example, it produces stereo imagery only for a field of view of $+/-130°$(h), $+/-65°$ (v). The rest of the surround field of regard does not have depth information. On



(a)

(b)

Figure 2.7: (a) 3D geometry of the object is created and the object is shaded using cubemap projection. (b) Pixels are baked onto a texture and object is duplicated.

these areas, and for noise within the existing depth map, missing depth can be replaced with user-generated 3D geometry or corrected with healing tools (see Figure 2.8).

2.3 Modeling Results

To test our system, we decided on a diverse set of representative panoramas, for which some sort of ground truth depth information was obtainable. Ground truth 3D

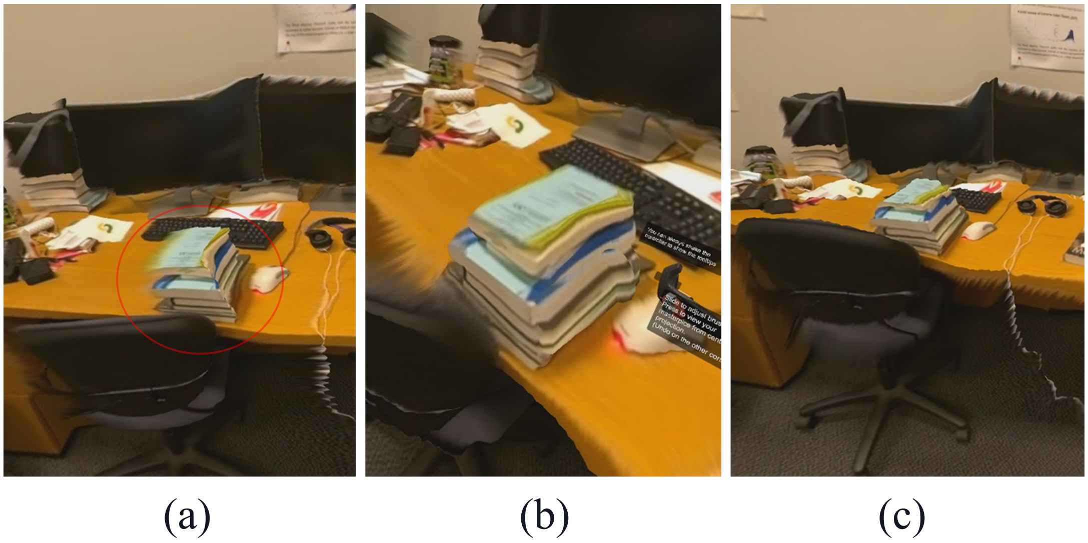

You can always shake the controller to show the tooltips
Slide to adjust brus
Press to view your masterpice from cent projection.
(Undo on the other con
(a)
(b)
(c)

Figure 2.8: (a) Error in depth information retrieved from the Nokia OZO panoramic camera is visible on the books. (b) User is refining the depth using the brush tool. (c) The result (just the stack of books was modified)

models of panoramic scenes are generally hard to come by, as most panorama capturing techniques will not provide noise-free depth information. Therefore, we decided to include computer-generated photorealistic-looking scenes, as those contain the actual ground-truth geometry implicitly. We created several surround environments from commercially available raytracing resources and rendered them using a bidirectional path tracer. We also decided to compare our method with scenes we captured with the Matterport 3D capturing system [51]. We selected 6 representative panoramas (3 synthetic outdoor scenes and 3 Matterport-modeled indoor scenes) for our test set. An expert user worked with PanoTrace on 2D panoramas for each of these scenes, for a maximum of 20 minutes to create PanoTrace scene models, not using any ground truth information, just the plain 2D panorama for each scene, and our system.

The Matterport capturing system is much more suitable for indoor scenes than for outdoor scenes and high-frequency depth information is challenging for it. We decided to dedicate the rendered scenes to outdoor environments and high-frequency details such

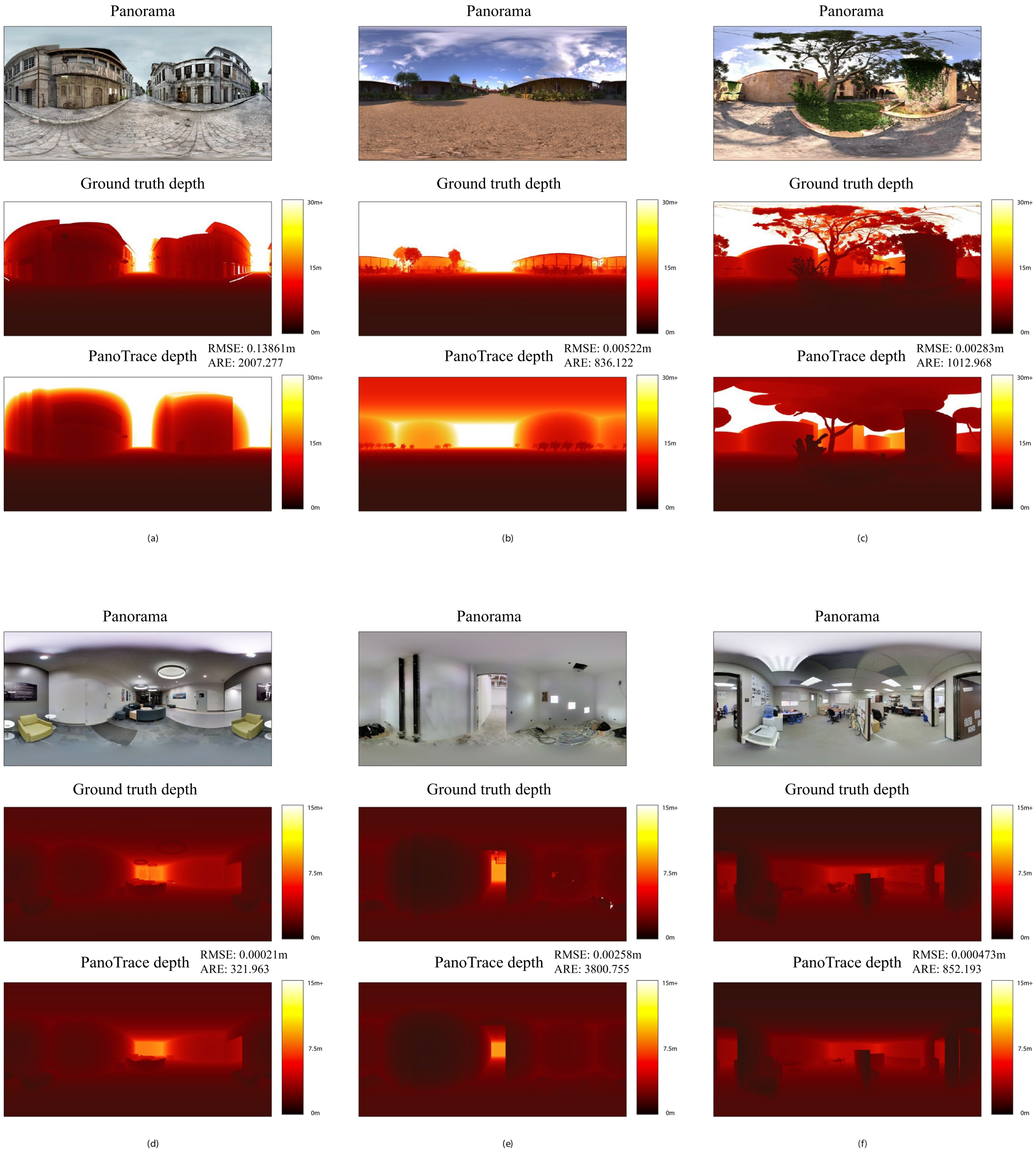


Figure 2.9: Panoramas along with the ground-truth depth map and the depth map resulting from the expert user's attempt to model the scene with PanoTrace. (a),(b) and (c) are synthetic rendered scenes and (d),(e), and (f) are real scenes captured using the Matterport system. RMSE and ARE errors are provided for each PanoTrace model

as plants and foliage. All 3 Matterport scenes cover indoor environments. Overall, this selection approach ensured the variability of test cases in our user study and lets us compare the success of our system in modeling different types of environments.

Figure 2.9 shows the results of the PanoTrace modeling efforts as equirectangular projections for each panorama, PanoTrace-created depth map, and ground truth depth map. It also lists two common depth image difference metrics [67] for each pair of PanoTrace and ground truth depth images: Absolute Relative Error [68] and RMSE [69]. The smaller these values, the more similar the PanoTrace panorama models were to ground truth. Several observations can be made here:

In terms of modeling error as compared to ground truth for each depth map stemming from the 20-minute PanoTrace session our reasonably experienced modeler spent on each panorama, the indoor Matterport scenes generally are better than the synthetic outdoor scenes (with the exception of panorama e), which is a panorama of a more confined space than the others, and the ARE metric biases against that to a certain extent - perceptually, the modeled depth and the ground truth are still very close there). The example of panorama c) demonstrates that natural geometry such as tree branches and foliage are difficult to model in a short amount of time, and panorama b) exemplifies a case where visually apparent differences do not factor too badly into the metrics because relevant geometry occurs generally at large distances.

2.4 User Study

A user study was conducted to compare user experiences with different geometrical representations of the same panoramas, including the results of our PanoTrace modeling system. Panoramas can be presented on VR headsets in different ways. In all cases, the user's head was orientation-tracked, so that he or she could look around naturally in the surround panorama. Head position tracking did only have an effect in our 6DoF conditions (see Study Setup below for conditions). When geometry (a simple skydome, our PanoTrace model, or a ground truth model) was present and 6DoF tracking mode was on, the user could move the head around, and even take a few steps, to perceive the scene with motion parallax. In that case, textures behind objects were simply duplicated (see Figure 2.3 a). We were interested in the question of what advantages a 3D panorama might have over a 2D panorama in terms of perceived realism, immersion, and viewing comfort, and how far a PanoTrace model with a modeling time limit of 20 minutes can get you towards a detailed 3D panorama. In this section, we discuss the details of this study.

2.4.1 Participants

We iteratively designed the study by first performing pilot studies with 3 users. Then, for the actual study, a total of 11 participants were recruited, ages 18 to 32 years old (average 23.2), 7 male and 4 female. Participants had either normal (7 users) or corrected vision (4 users). We did not have participants with colorblindness or stereoblindness, as determined by standard tests. Of the 11 users, 8 reported themselves as having “only tried out VR a few times,” 2 said they were not familiar with VR at all, and 1 user said they “frequently used VR.” Each user was compensated with $10 US for the 1.5 hours they spent on the study.

2.4.2 Study setup

To do the study, 6 panoramas were chosen, 3 of them synthetic and rendered using a path-tracer, and 3 of them captured with a Matterport 3D capturing system. Each panorama was presented with 7 different conditions (3DoF and 6DoF represent 3 and 6 degrees of freedom, respectively):

1. 2D: Plain 2D panorama without being projected to 3D geometry (equivalent to an infinitely large sphere, only viewed in 3DoF).
2. Dome 3DoF: panorama projected on a user-adjusted hemisphere geometry, viewed with 3DoF head orientation tracking.
3. Dome 6DoF: panorama projected on a user-adjusted hemisphere geometry, viewed with 6DoF head-tracking.
4. PanoTrace model 6DoF: panorama projected on the model that an expert user prepared in 20 minutes, viewed with 6DoF head-tracking.
5. PanoTrace model 3DoF: panorama projected on the model that an expert user prepared in 20 minutes, viewed with 3DoF head orientation tracking.
6. Ground truth with 6DoF: panorama projected on the ground truth geometry, viewed with 6DoF head-tracking.
7. Ground truth with 3DoF: panorama projected on the ground truth geometry, viewed with 3DoF head orientation tracking.

The user-study framework was developed using the Unity engine [70]. The study system presents the panoramas in random order on an HTC Vive VR head-mounted display (HMD). For each panorama, the user would experience 2 repetitions of each condition, so that we could check on the consistency of responses.

2.4.3 Task

The 7 conditions listed above with 2 repetitions each resulted in 14 test scenes for each panorama. Each user was shown scenes 1–14 in two cycles. In the first cycle, the user would be given an overview of all 14 scenes, and then in the second cycle, they would answer questions about each. Users could move on to the next scene by pushing a button on their controller. The scene number would show up on their controller. After experiencing each scene in the second cycle, users had to answer these 3 questions on a Likert scale:

- "How strongly did you feel like actually being in the scene?" (immersion)
- "How realistic was the experience of the scene?" (realism)
- "How much discomfort did you experience while viewing the scene?" (discomfort)

The questions would pop up on the display after the users chose to proceed from the scene.

2.4.4 Experimental design

We used a within-subjects design with 3 dependent variables (immersion, realism, discomfort) and 2 independent variables (6 panoramas, 7 conditions). We really considered four different conditions (baseline 2D, Skydome, Model from PanoTrace, and ground truth) and two different degrees of freedom for viewing (3DoF and 6DoF) but since it doesn't make sense to experience the plain 2D panorama in 6DoF, we simply enumerated the seven resulting conditions. The order of independent variables were defined randomly for each user.

Our hypotheses about the outcome of the study were as follows:

H1: Users will experience more immersion, more realism, and less discomfort in expert-user-modeled scenes (using our tool) compared to the less detailed dome scene.

H2: Users will not experience more immersion, more realism, and less discomfort in the ground truth model scenes compared to the expert-user-modeled scenes using our tool.

2.4.5 Procedure

Before the study could begin, each participant was tested for colorblindess using Ishiahara color plates, and for stereoblindness using a VR random dot stereogram. Then they filled out the pre-study questionnaire with their demographics and some background information. Next, the user was asked to stand in the tracking area while wearing the HMD. The test administrator explained how to operate the system and how many scenes are they going to explore, but they did not recommend any option. Also, users were not asked to pay attention to the differences between the scenes. They were told that they could have some limited movement in the tracking space (e.g., walking a few steps).

After finishing the study, users could choose to interact with the modeling system in the remaining time. Administrator explained how to toolset would work and how to use the tooltips to learn the interactions in our tool. Later, users could choose to model one of our modeled scenes from scratch.

2.4.6 User study results

After experiencing each scene, participants were asked to rate the experience based on the amount of immersion, realism and discomfort that they felt, on 7-point Likert scales. The average values are shown in Figure 2.9.

We performed a Wilcoxon signed-rank test on the dependent variables to check for any statistically significant difference between 3DoF and 6DoF. All three variables, Immersion, Realism, and Discomfort, showed to be significantly affected by DoF (see Ta-

Table 2.1: Wilcoxon signed-rank test on the dep. variables by DoF. Listed are the Likert question means, critical z, and p values. Scenes were perceived as more immersive, more realistic, and less discomfort-inducing with 6DoF viewing.

Dep. Variable	M 3DoF	M 6DoF	z	p
Immersion	4.39	4.81	5.532	<.0005
Realism	4.42	4.56	3.007	0.003
Discomfort	2.38	1.97	-6.929	<.0005

ble 2.1), with 6DoF viewing leading to higher reported immersion and realism, and lower discomfort levels.

For each dependent variable, a Friedman test was run to determine if there were differences in participant responses using any of the different 3D Model categories (2D, Dome, PanoTrace, and Ground Truth). Pairwise comparisons were performed with Bonferroni correction for multiple comparisons.

Main Findings

The main findings relating to the performance of our PanoTrace models in terms of perceived immersion, realism, and viewing comfort among the 6DoF viewing conditions can be summarized as follows:

- PanoTrace models and Ground Truth models provided a greater sense of immersion than Dome models when viewed with 6DoF head tracking.
- Both PanoTrace models and Dome models provided a lower sense of realism than Ground Truth models overall across all six panoramas when viewed with 6DoF head tracking.
- For indoor Matterport models, only Dome provided a lower sense of realism than Ground Truth.

Additionally, we observed the interesting effect among the 3DoF viewing conditions that the Dome condition resulted in lower viewing discomfort than the plain 2D panorama condition.

We categorize all detailed results as follows:

6DoF immersion

Immersion values were significantly different using different model categories, $\tilde{\chi}^2(2) = 31.311$, $p < .0005$. Post-hoc analysis revealed statistically significant differences in Immersion from Dome (Mean = 4.44) to PanoTrace (mean = 4.80) ($p = 0.027$) and Dome (mean = 4.44) ($p = .027$) to Ground Truth (mean = 5.19), but not between Ground Truth and PanoTrace models.

This supports both hypotheses H1 and H2.

6DoF realism

Realism values were significantly different using different model categories, $\tilde{\chi}^2(2) = 21.919$, $p < .0005$. Post-hoc analysis revealed statistically significant differences in Realism from Ground Truth (mean = 4.98) to Dome (mean = 4.24) ($p < 0.0005$) and from Ground Truth (mean = 4.98) ($p = .049$) to PanoTrace, but not between the Dome and PanoTrace models.

This supports hypothesis H1 but does not support hypothesis H2.

6DoF discomfort

Discomfort values were significantly different using different model categories, $\tilde{\chi}^2(2) = 31.311$, $p = .004$ according to the Friedman test, with discomfort highest for the Dome geometry. However, a Bonferroni post-hoc analysis did not reveal statistically significant differences in discomfort from different models.

Based on these results we decided to narrow down the data-set in one more step. We filtered out the panoramas that were computer-generated and focused on the indoor panoramas that were captured by Matterport.

6DoF - indoor - immersion

Immersion values were significantly different using different model categories, $\tilde{\chi}^2(2) = 17.495$, p ¡ .0005. Post-hoc analysis showed statistically significant differences in Immersion from Dome (mean = 4.09) to PanoTrace (mean = 4.83) (p = 0.027) and Dome (mean = 4.09) (p = .001) to Ground Truth (mean = 5.12), but not between the Ground Truth and PanoTrace models.

This supports both hypotheses H1 and H2.

6DoF - indoor - realism

Realism values were significantly different using different model categories, $\tilde{\chi}^2(2) = 13.624$, p = 0.001. Post-hoc analysis revealed statistically significant differences in Realism from Ground Truth (mean = 4.92) to Dome (mean = 3.91) (p = 0.006) but not between the Ground Truth and PanoTrace (mean = 4.50) models.

This supports hypothesis H2 but does not support hypothesis H1.

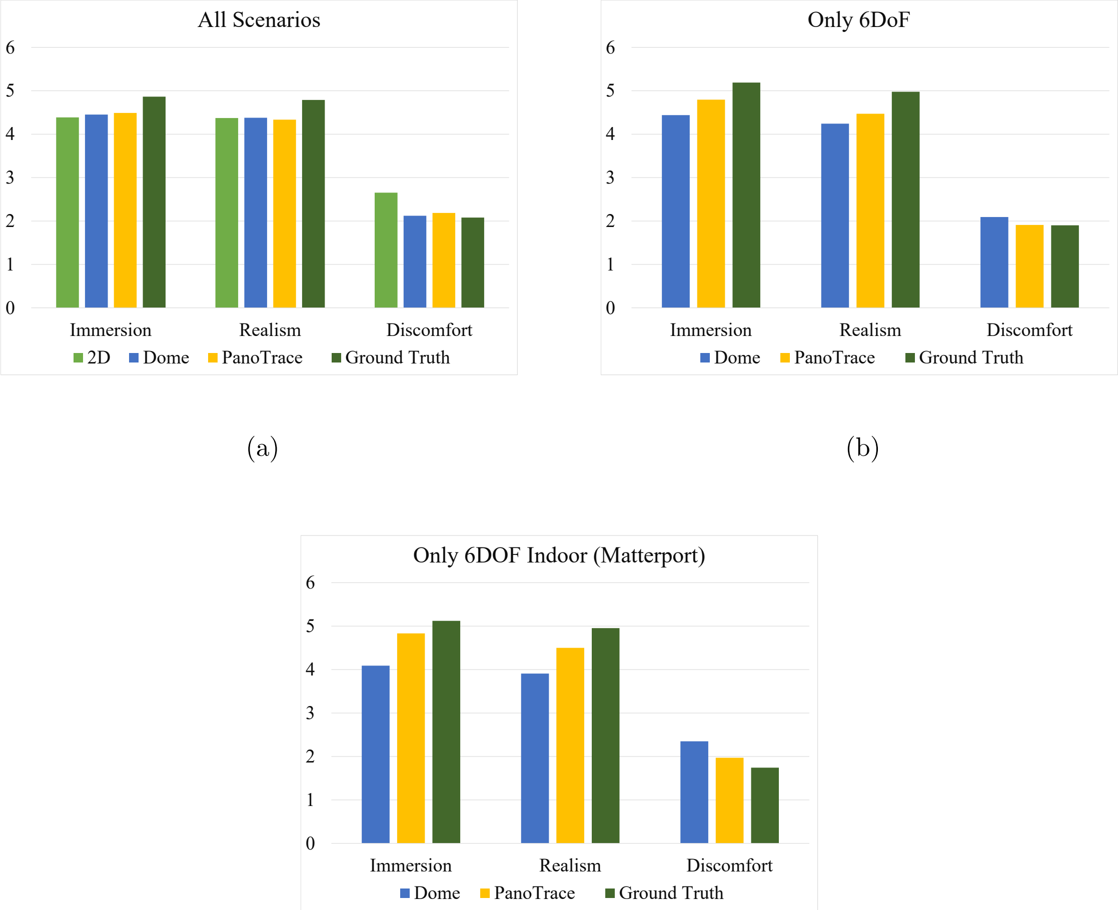
6DoF - indoor - discomfort

Immersion values were significantly different using different model categories, $\tilde{\chi}^2(2) = 16.775$, $p < .0005$. Post-hoc analysis showed statistically significant differences in Immersion from Ground Truth (mean = 1.74) to Dome (mean = 2.35) (p = 0.008) but not between the Ground Truth and PanoTrace (mean = 1.97) models.

This supports hypothesis H2 but does not support hypothesis H1.

3DoF immersion, realism and discomfort

We performed the Friedman test using the results from all 3DoF conditions. The test showed significant differences between the models for all 3 dependent variables. A set of Bonferroni post-hoc tests only showed a significant decrease (p = 0.037) of discomfort for dome (mean = 2.15) as compared to 2D (mean = 2.65).



(a)

(b)

(c)

Figure 2.9: User responses for Immersion, Realism, and Discomfort. Note that only (a) includes a plain 2D panorama condition. (b) and (c) are showcasing results under the 6DoF-head-motion condition, which is not meaningful for 2D panoramas

2.5 Discussion

We expected the result for our expert users's PanoTrace models to fall in between the ground truth and our very basic 3D model, the dome, in many of the scenarios. Across all datasets, we could not find a statistically significant difference between PanoTrace models and the dome. However, for 6DoF viewing, this was the case. The mean for all three dependent variables for our method was always in between the ground truth and the dome and there was significance for immersion between Dome and PanoTrace but not between PanoTrace and Ground Truth.

Most of the results that were aligned with our hypotheses were about the participants' sense of immersion. This may mean that even a modestly realistic environment can feel more immersive if it somehow provides a proper response to the user's movements.

In the 6DoF cases, the realism perceived from PanoTrace models was significantly less than the ground truth. This could be related to the fact that our tool is not really strong at modeling organic and very detailed geometry, such as trees and foliage. This was a motivation for us to also analyze the data regarding indoor Matterport scenes separately.

On the indoor Matterport scenes, our method is shown to be very effective. This could be due to three factors. First, Matterport scenes do not have many organic objects, so it was easier to for the expert user to create a representation of the scene. Second, Matterport models are not the actual ground truth, as the system has artifacts and is an approximation of the true geometry. Third, the Matterport scenes all include geometry close to the user. The skydome approach would not naturally do well with such scenes.

The very significant difference between 6DoF and 3DoF shows the importance of the movement for the users, since users were told they could move if they wished. Many of them did try to walk in the environments, and reported feeling dizzy when the panorama was 'moving with them' as happens with 3DoF viewing.

There appears to be a benefit of using simple dome geometry in the 3DoF viewing case in terms of discomfort, i.e., the Dome condition had lower discomfort than the plain 2D condition. This means that surround panoramic content in VR would likely benefit from this simple viewing adjustment.

2.6 Conclusion and Future Work

We designed and developed a system to model 2D panoramas interactively in virtual reality, featuring several novel interactions. We demonstrated the tool by modeling several scenes, both reconstructed physical spaces, and virtual computer graphics scenes, for which ground truth geometry was accessible. Then we designed a user study, in which participants experienced different versions of six surround-view panoramas and reported their sense of immersion, scene realism, and discomfort. An analysis of the collected data provided statistical evidence on the significance of our tool.

In the end our tool proves to be very effective in certain areas, such as modeling man-made indoor scenes (rooms and furniture) and still of relevance for complex natural outdoor scenes.

However there are a lot of expansions that would seem appropriate for this work. A more automated approach involving additional computer vision constraints would be a worthwhile extension of this system. One might utilize previous existing machine learning [71, 72] or computer vision approaches [73] to generate 3D geometry from a single image. Also, using interactive lighting control could be an effective way of increasing the interactivity, user engagement and immersion. Implementing a relighting algorithm for the projective texture mapping would also benefit this system.

Chapter 3

Walking and Teleportation in Wide-Area Virtual Reality Experiences[1]

Virtual reality (VR) has the potential to revolutionize the way we experience entertainment and interact with the world around us. However, in order for VR to reach its full potential, it is important to understand how people respond to and interact with different types of VR experiences.

Location-based VR, in particular, offers a unique opportunity for people to engage with VR in a way that is not possible with at-home VR. In location-based VR, people are able to physically move around a large space, which can provide a more immersive and interactive experience. At the same time, it is important to consider the trade-offs of natural walking versus artificial locomotion techniques, such as teleportation, in terms

[1] The contents of this chapter have been previously published in 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) © 2020 IEEE. Reprinted, with permission from Ehsan Sayyad, Tobias Höllerer, and Misha Sra, Walking and teleportation in wide-area virtual reality experiences [74]

of presence and user preference.

In this chapter, we show that by comparing natural walking to teleportation in a large physical space, we were able to understand the impact of these two different locomotion techniques on presence and user preference. Our results suggest that walking may be a preferred method of locomotion in VR and that teleportation may lead to increased simulator sickness, while also indicating a potential relationship between natural walking and increased presence.

Overall, this study has important implications for the design and development of location-based VR experiences. By understanding how people respond to different locomotion techniques, we can create VR experiences that are more immersive, engaging, and enjoyable for users.

3.1 Introduction

Since the early days of virtual reality (VR), the consumer market has been a major focus of the field. With the introduction of cheaper and more capable devices in recent years, consumer VR has continued to gain popularity and is now increasingly inspiring creators to integrate VR in entertainment experiences in innovative ways. For example, amusement parks offer VR roller coaster rides [75], entire theme parks are dedicated to VR[76], and chefs in New York City host culinary experiences that incorporate VR [77]. Location-based VR has emerged as a new category of entertainment with venues set up in warehouses or existing malls and movie theaters. Companies create bespoke high-end experiences[78, 79] where groups of people roam freely in large physical spaces. While location-based VR setups vary and can utilize a wide range of tracking spaces from 5x5m rooms in Dreamscape [79] to much larger warehouse-sized spaces in The Void [78], in this work we consider tracking spaces larger than 10x10m as wide area. Real walking is a primary form of locomotion in these experiences. Sometimes, the VR space size requirements are constrained by physical space size, layout, and obstacles. If a large open-tracked interaction space is available for wide-area VR, VR environments can flexibly be layered on top of the physical space.

A compelling use case for virtual and augmented reality (AR) deployed in wide-area environments is the possibility of storytelling for educational and entertainment purposes in the physical world. Location-based AR applications such as Pokémon Go have already shown tremendous success. With a few more technological advancements, one could imagine creating *immersive* narratives similar to theme parks anywhere without the theme park infrastructure. Immersive AR content could seamlessly transition to immersive VR content depending on the type of physical space a user walks through. We are interested in exploring the technical and cognitive feasibility and side effects of such scenarios. In pursuit of these future possibilities, we present what we believe is the first study evaluating cognitive impact of real walking in VR over wide-area spaces. Prior work has explored real walking in wide-area VR, in both indoor and outdoor spaces. Hive [80] is a 570 m^2 indoor space tracked with an outside-in World Viz PPT X8 tracking system. The virtual environment is rendered on backpack-worn computers to enable mobility in the tracking space. In contrast, VRoamer [81] uses a head-mounted device with inside-out tracking to dynamically generate virtual elements in ways that allow users to safely walk in indoor spaces. DreamWalker [82] is a system for walking to a pre-defined real world destination while staying fully immersed in VR with pathfinding and obstacle avoidance in a pre-authored VR environment. Though some of these works include user studies to validate the design aspects of the system, there remains a gap in the literature on studying the effects of walking in wide-area VR on presence, simulator sickness, and cognitive map building. There has also been limited research on comparing natural walking and teleportation in these domains. Teleportation has been compared to joystick based movement [83, 84] and other forms of locomotion in room-scale VR. Similarly, natural walking in small environments has been compared to multiple forms of joystick control with respect to several aspects of cognition [85]. Our work aims to address this gap by comparing wide area natural walking with teleportation for presence, simulation sickness, cognitive map building, and user preference.

The benefits of supporting natural body movement have been extensively studied in VR. For example, walking has been shown to result in higher self-reported presence than walking-in-place and joystick-based locomotion [86]. Walking has also shown superior performance on search tasks [87] with benefits for spatial orientation [88] and attention [89]. Despite well-known advantages of walking in VR, it is not often employed in room scale experiences because a direct mapping of physical walking to virtual motion makes it impossible to reach virtual spaces that fall outside of the boundaries of the physical tracking space [90].

Since walking in VR has been studied since the mid 90s, it is not very surprising that some study results are contradictory. We believe the differences are probably due to the studies being conducted in different decades with different hardware and virtual environments. For example, in 1999 Usoh et al. [86] showed walking to elicit higher presence than walking-in-place or flying in an indoor environment of 5x4m, while in 2005 Zanbaka et al. [91] found no differences in simulator sickness between real walking in a small room and several virtual travel techniques, and in 2009 Suma et al. [92] showed walking to cause high motion sickness in a complex maze environment [92].

For location-based VR, walking is the primary form of locomotion, while teleportation tends to be the primary form for room-scale VR experiences. Both primarily stem from tracking space availability, though location-based VR experiences are also co-located social VR experiences [79, 78] for which walking works better than teleportation. As each locomotion technique varies in its usability, influences the user's sense of presence differently, fatigues the users to varying degrees, and elicits different levels of motion sickness, it also has a different impact on virtual task performance [87].

Most of the previous comparative studies on natural walking use joystick control as an alternative interface. In a more recent study [84], Buttussi et. al. showed that a point and teleport interface can be superior to joystick control with regard to simulator sickness, presence, and ease of use. We chose point and teleport as our comparison locomotion technique. In this work, we explore natural walking and teleportation in a wide-area space to understand their influence on presence and cognitive map building. Walking in large physical spaces has only recently become affordable, due to the availability of standalone VR headsets such as the Lenovo Mirage Solo or the Oculus Quest that have built-in inside-out tracking.

To the best of our knowledge this is the *first comparative study of natural walking in a wide-area VR experience*. We compared a variety of metrics against controller-based point and teleport. We also investigated the transfer of previous findings about real walking both to wider areas and to state-of-the-art lightweight mobile VR headsets in high-fidelity outdoor and indoor virtual environments.

We encouraged scene exploration via an object collection task and compared *virtual scene coverage* and *mental map formation* through a series of pointing tasks after exploration. We also assessed *user preference*, and administered *presence* and *simulator sickness* questionnaires.

Our results provide insights into the effects of walking and teleportation in wide-area VR experiences. They indicate decided *advantages of natural walking over teleportation* in wide-area VR in terms of user preference and induced disorientation with some indications of better mental map formation.

3.2 Related Work

Here we discuss related prior work in three categories: locomotion in AR/VR, redirected walking and spatial cognitive map building.

3.2.1 Locomotion in AR/VR

Navigation is a universal task performed in both real and virtual environments [93]. AR users can easily navigate and avoid obstacles as the world is visible through AR displays. Games like Pokémon Go [94] and Human Pacman [95] are successful examples that enabled interaction with virtual objects while moving in the real world with high levels of enjoyment and sensory gratification. However, research shows risks of injury even when the physical environment is visible [96]. This risk is multiplied in VR when the physical world is not visible.

Walking in VR has been desirable due to its ease of use [87] and its ability to elicit higher presence compared to other techniques like walking-in-place or flying [86], and joystick-based locomotion [83]. However, effectively navigating VR environments without provoking VR sickness continues to be a major obstacle for VR development [97]. A lot of research has focused on creating novel techniques to enable walking in room scale setups, such as redirected walking[98], resetting [99], or perceptual illusions in VMotion[100]. However, these techniques are not yet commonly available in at-home VR experiences as they require at least some tracking space for the user to move. For example, VMotion requires the user to have at least a 4x4m space while for curvature gain to remain undetected, the circular walking arc needs to have a radius of at least 22m [101]. More recent work shows that users can be redirected on a circular arc with radius of either 11.6m or 6.4m depending on the estimation method used [102] As a consequence of tracking space limitations, teleportation has become the most commonly used locomotion technique in room scale VR experiences [103].

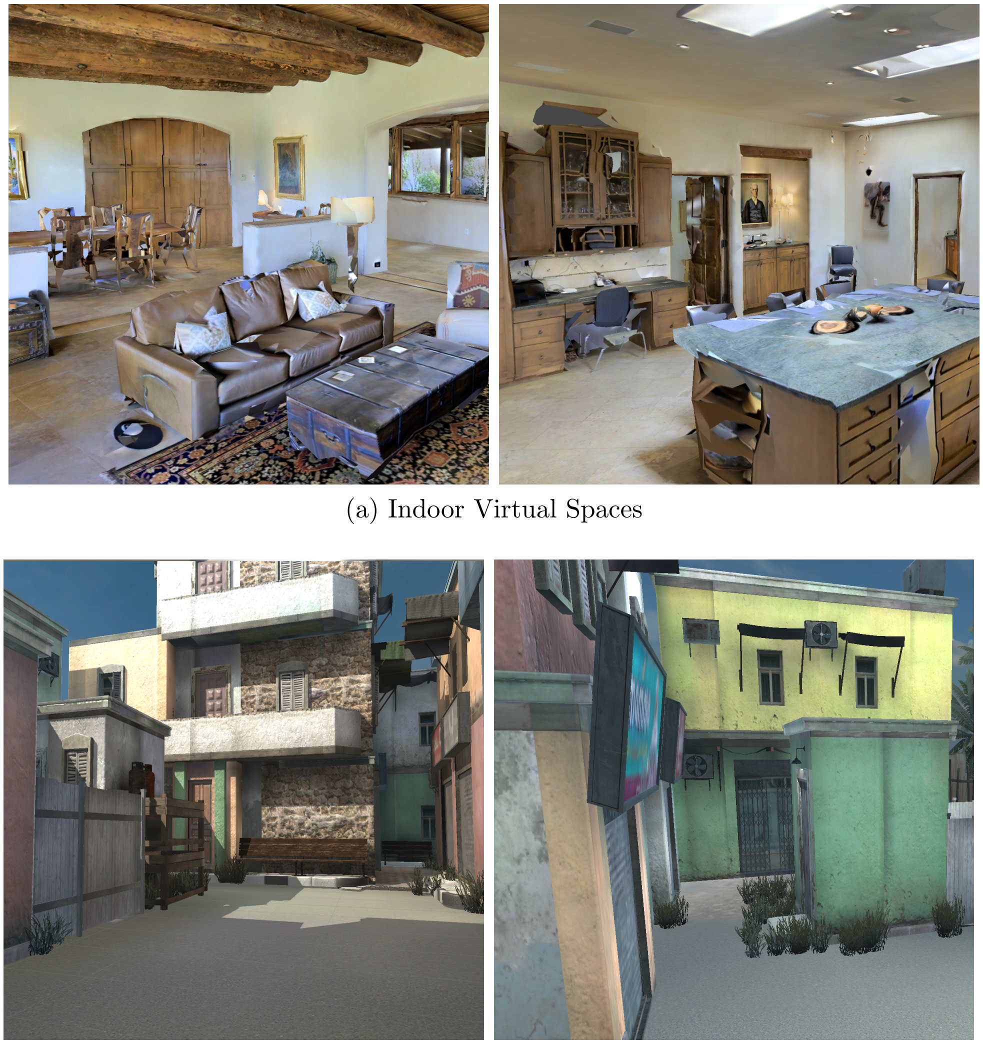
The most basic form of teleportation involves the user pointing a controller towards a position in the virtual world and clicking a button to instantly move there [104]. Since it discontinuously translates the viewpoint, instant teleportation does not generate any optical flow, and thus reduces the risk of vection induced VR sickness [105, 106, 107, 108]. However, this beneficial reduced VR sickness comes with an increase in disorientation and break in presence [109]. To address this many variations of teleportation have been proposed that each show improvement in certain aspects. For example, the scene blinks to blackness momentarily as one moves to a new location in Blink [110]. Or in Telepath [111], users move smoothly along a hand-drawn path at walking speed. In Dash [108], user viewpoint is discontinuously but rapidly translated to the point of interest, which leads to better path integration.

3.2.2 Redirected walking

Redirected walking allows users to walk naturally in virtual worlds through continuous manipulation of mapping between physical and virtual rotations that steer the user away from the tracking space boundaries [98]. To overcome tracking space limitation, a number of redirection techniques have been proposed that manipulate the user's perceived self-motion such as translation [112, 113], rotation [101], or curvature gain [114], motion compression [115], and virtual scene manipulation such as portals [116], saccadic redirection [117], and non-Euclidean geometry [90] to enable walking in the larger virtual area without exiting the smaller tracking area. All of these techniques either rotate the virtual world or scale the user's motion to allow them to cover more virtual ground. They typically require a large physical space [101] or have difficulty changing a user's direction when the user gets close to tracking space boundaries [118], and are thus some-what limited in their use in consumer applications. Reorientation or resetting is a class of techniques that stop and reorient the user when they are close to the boundary of the tracking space, unlike redirection, which is applied continuously. Williams et al. [99] proposed reset techniques that direct the user to walk backward or to physically turn around while the virtual world remains frozen. As resetting interrupts the natural walking experience, it may decrease the user's sense of presence. To mitigate disruption, VMotion [100], combines the perceptual illusion of inattentional blindness with various visibility control techniques to mask the virtual world rotation by engaging the user in the story.

3.2.3 Cognitive map building

People learn the spatial layout of new environments (locations, distances, directions) relatively quickly and acquisition begins as soon as they arrive in a place [119]. They create a world-reference frame for the layout representing the environment in an allocentric form also called a cognitive map, survey knowledge, mental model, or mental map [120]. Peck et al. [121] compared redirected walking to walking-in-place and joystick for navigational ability (i.e.,, the performance during a search task) and reported that participants "traveled shorter distances, made fewer wrong turns, pointed to hidden targets more accurately and more quickly, and were able to place and label targets on maps more accurately" when using redirected walking. Langbehn et al. [83] also showed that redirected walking better enabled users to unconsciously acquire spatial knowledge about the virtual world than teleportation or using a joystick. In another study where portals were used to reorient the user, teleportation was compared to joystick and walking through portals [122] and was found to be faster than walking, but worse than joystick for determining orientation. Proprioception and translational body-based information



(a) Indoor Virtual Spaces

(b) Outdoor Virtual Spaces

Figure 3.1: (a): Indoor virtual spaces, which are two adapted halves of the same Matterport 3D dataset model. (b) Outdoor virtual spaces, created by authors in two different shapes.

was found to significantly improve navigational performance and accuracy of cognitive maps [120, 87]. Since walking provides proprioceptive information and is inherently translational, it should help in better cognitive map building than non-translational locomotion techniques like teleportation, and this forms the basis of our experiment.

3.3 Formative Experimentation

We used the Lenovo Mirage Solo VR headset with inside-out tracking and a 3DoF handheld controller and Google VR SDK for Unity for the experiment. By default, for safety, the device fades to black if the user moves around more than 1m, a measure that we disabled in developer mode to allow for walking large distances. The SDK, however, does not provide access to the localization and mapping process, so it was not possible for us to tune or improve the tracking programmatically. Therefore, we needed to find, by trial and error, physical locations with favorable tracking conditions, to run a wide-area walking experiment.

We tested several outdoor areas at different times of the day to determine where our device's tracking system would work best. While tracking worked intermittently in all these spaces, outdoor dynamic lighting proved to be problematic. Even though most of our testing was done during the day in shady spots or close to dusk, we damaged the display on one device due to accidental exposure to sunlight when transporting the device to the test site. Lighting also changed dramatically during the hour of the experiment, causing uneven tracking performance. The inside-out tracking would also fail often as there were fewer distinct texture features in these wide area spaces than necessary for tracking to work seamlessly. We conducted tests in a baseball field, a soccer field with floodlights, on a grassy campus quadrangle, and on a marked track and field area. While we expected tracking to work due to sufficient texture, field markings and other such features, it failed repeatedly, most likely due to the self-similarity of grass and ground textures and other landmarks being too far away to create sufficient parallax.

We experimented with a mixed-reality locomotion technique in which we would fade in the camera feed from the headset into the user's peripheral vision each time they started moving and fade back to full VR view when the user stopped and simply looked

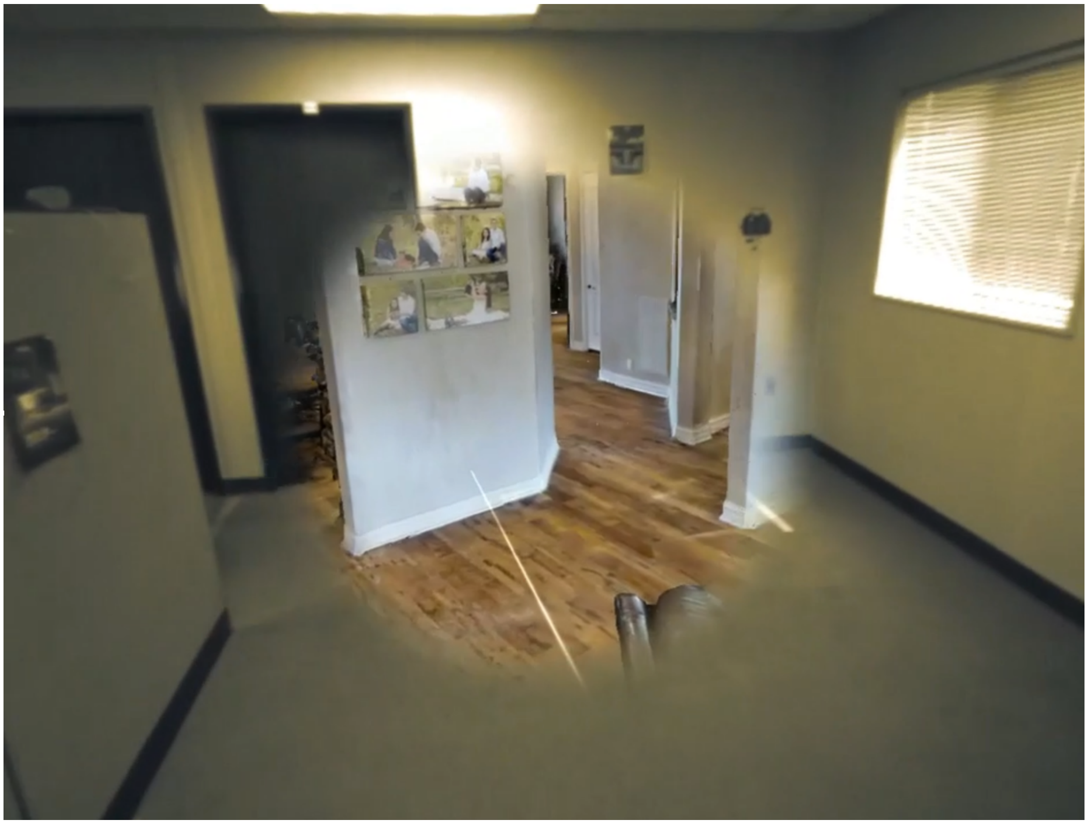


Figure 3.2: Blending in the real environment on the periphery to help users avoid physical obstacles. This technique was not used in our final walking experiment, but is an interesting option in smaller spaces.

around (see Figure 3.2). Pilot users testing this interface successfully maintained focus on the virtual environment and felt immersed even while walking and seeing the real world in their periphery. While this technique creates an effective AR + VR interface in a single device, it also introduces differences for different users and different scene explorations, as users would see the physical world shining through at different times.

The lighting and tracking issues led us to test indoors in a few different basketball courts and gyms on our campus, and in those environments, there was no risk of bumping into obstacles or stumbling on uneven ground. Therefore, we did not end up employing this technique for our experiment in the end, but it is a compelling UI for eventual deployment of wide-area VR walking.

3.4 Experiment

In this experiment, we analyze the effects of two locomotion techniques, namely, natural walking and instant teleportation, on cognitive map building in VR. We also consider other aspects of virtual experiences, such as motion sickness, sense of presence, and user preference of the locomotion technique.

Instant teleportation is a point-and-click mechanism that does not require the user to physically move while allowing them to change their virtual location instantly. It can be implemented easily for commodity VR setups without any additional bulky equipment and is included by default with the primary VR plugins like SteamVR for Unity3D. As mentioned before there are many variations of teleportation with varying benefits for minimizing user discomfort. We chose this basic implementation for our experiment as it was the most used interface in the previous studies. In our implementation, the controller touchpad button is pressed down and held and the teleportation destination is indicated by a ray followed by a circular marker on the ground (see Figure 3.3). Instant movement to the destination is accomplished by releasing the touchpad. The visual leap can cover both short and long distances as long as the user is able to point and place the circular marker on the virtual floor plane. This means the user can go from one end of a large virtual environment to the other end in an instant, if there are no walls in between blocking their line of sight. The user's initial body orientation determines their arrival orientation at the destination.

The experiment was conducted in an indoor roller hockey rink of size 61m x 26m (NHL regulation size) that was prepared for robust 6DoF tracking with our VR headset with the inclusion of extra ground texture as described below in 3.4.2 (see Figure 3.4). Based on dominant results from prior work in room-scale VR that have shown walking to perform better than other techniques on presence, spatial mapping and motion sickness,

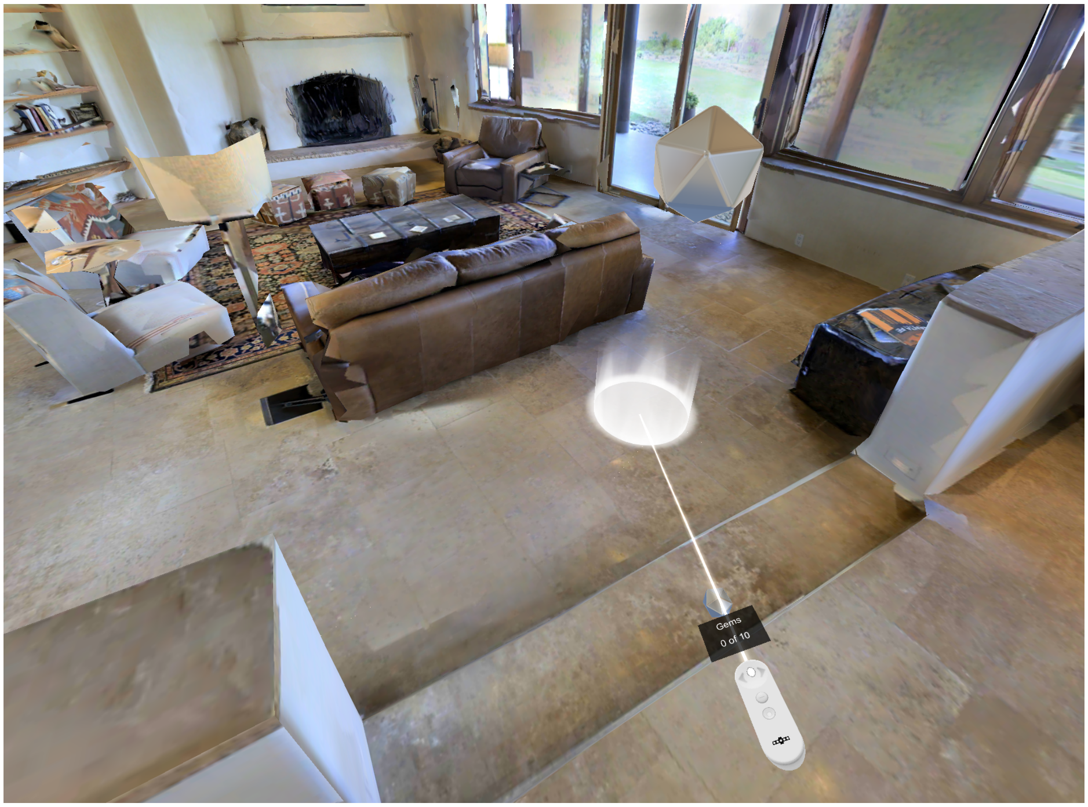


Figure 3.3: Gem collection and Teleportation tool, both visible on the controller. Users pressed the touchpad button to teleport to the white spot. They pressed the second button on the controller while pointing the 'laser beam' at the gems (metallic icosahedron) in order to collect them. The UI on the controller kept track of the number of gems collected.

we defined the following hypotheses:

- H1: Natural walking will provide better spatial layout information than teleportation.
- H2: Teleportation will induce higher disorientation, except when inside-out tracking fails in natural walking.
- H3: Natural walking will provide a higher sense of presence than teleportation.

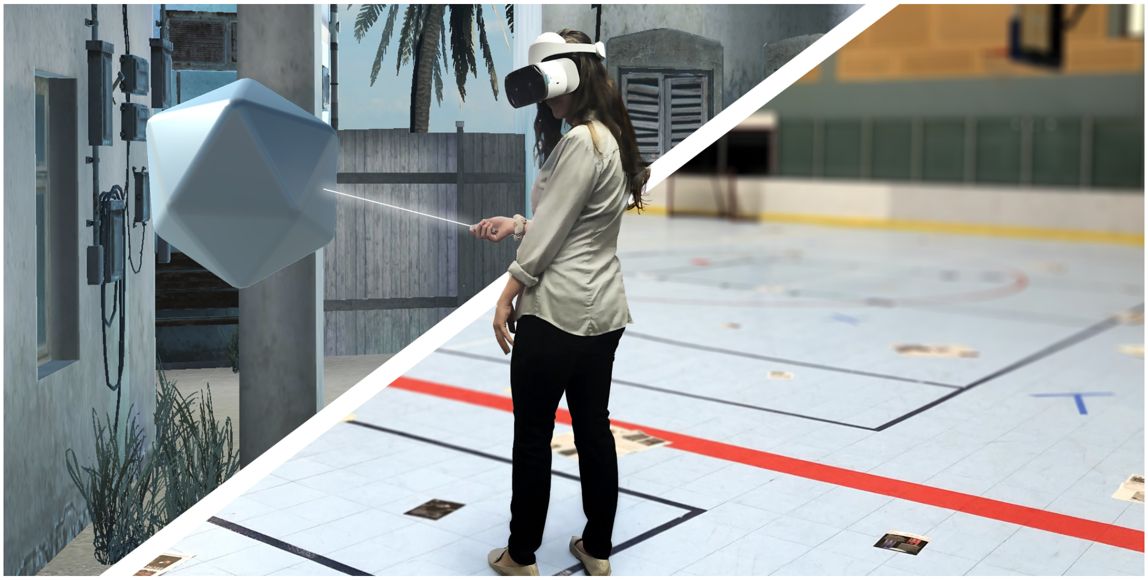


Figure 3.4: User shown in both our real wide-area study environment (roller hockey rink) and virtual environment with gem target.

3.4.1 Participants

We recruited 16 participants (7 male and 9 female) with an average age of 19. Most of them were students or employees of our university as we recruited on campus. 10 participants had at least some experience with VR with the mode being 1 and mean of 2.0 on a scale of 1 (no experience at all) to 7 (very experienced). The total time per participant, including pre-questionnaires, instructions, in-study tutorials, experiment conditions, and post-questionnaires was 1.5 hours. Each study was conducted in a single session and participants were compensated at the rate of 15 USD per hour. Participants wore the headset for 5 minutes per locomotion condition and navigated through two indoor and two outdoor virtual spaces during the experiment. Participants also went through two tutorials, two minutes long each. The study was approved by our university's office of research and all participants provided informed consent. Participants were given instructions on how to perform the experiment tasks before starting the experiment. They were asked to pay attention to their virtual surroundings during their exploration. Before starting, participants filled out a demographic questionnaire about their experience with gaming and VR along with the Santa Barbara Sense of Direction questionnaire (SBSOD) [123] and the Kennedy-Lane Simulator Sickness questionnaire (SSQ) [124].

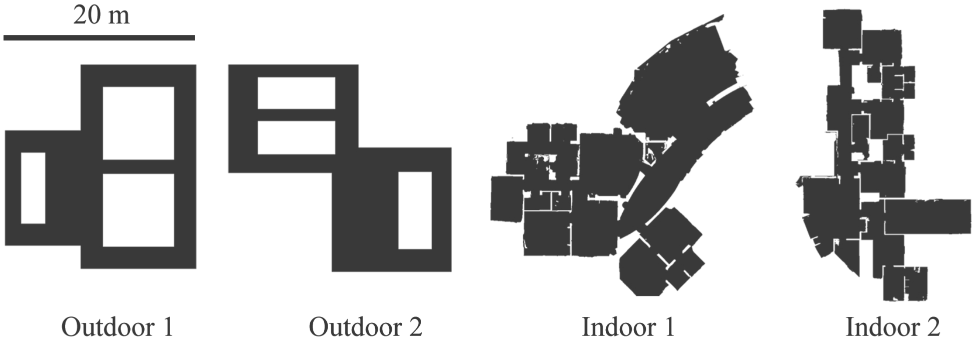


Figure 3.5: Top-down view of the outdoor and indoor environments.

3.4.2 Apparatus

Our final experiment was conducted in the aforementioned indoor sports arena, with 61m x 26m of open walkable area on campus using the Lenovo Mirage Solo VR device. To increase the accuracy of the headset's inside-out tracking system, we placed several additional markers (newspaper and other printed material) on the ground to augment the tracking environment with extra features. With these preparations, 6DoF tracking of our participants performed reliably most of the time. Participants wore noise cancelling headphones to follow verbal instructions in the two tutorials and kept wearing them during the experiment. Noise cancelling headphones were used to help reduce the natural environment noise in the university recreation center (see Figure 3.6).

To minimize direct interaction with users during the experiment and to make the experiment process more robust, we created a custom remote control web app that let the experimenters start the tutorial or the virtual scene from a distance or monitor the user in the virtual environment while simultaneously seeing them in the real space, helping the experiment run smoothly. The interface was implemented using Google's mobile platform Firebase and it relayed information like time, number of gems collected, and user location and orientation in each virtual environment to the experimenter. The app was built such that multiple instances could be run in any web browser with a key code. This helped us run and control parallel experiments using multiple devices in the same tracking space. The app recorded all user interactions including position, orientation and controller button clicks at 20Hz and saved it for future playback and analysis. An example of the user interaction playback can be seen in Figure 3.5.

The four views were rendered in Unity, with two indoor scenes and two outdoor scenes. Both pairs were similar in style and feel to make locomotion in them comparable but were different enough to avoid any learning effects (see Figure 3.1 and 3.5). Indoor scenes are two halves of the same scene[2] taken from the Matterport3D dataset [125]. The scenes were capped and edited in a 3D editing software to feel coherent to the users. We also removed steps and kept all the walkable surfaces flat. Outdoor scenes were made in Unity based on a designed map. All assets used to assemble the scene were downloaded from the Unity AssetStore. Assets were placed by hand around a pre-designed top-down map. precautions were taken to avoid using distinguishable assets like store signs in both scenes. this prevented creating identical areas and causing learning effects.

3.4.3 Procedure

We designed a within-subjects experiment and each participant did teleportation and walking during the experiment. They navigated through a total of six virtual scenes that consisted of one tutorial and two virtual environments per locomotion technique. Users were not allowed to walk through virtual walls and objects. Hence, they walked paths as

[2] Scene ID:ac26ZMwG7aT

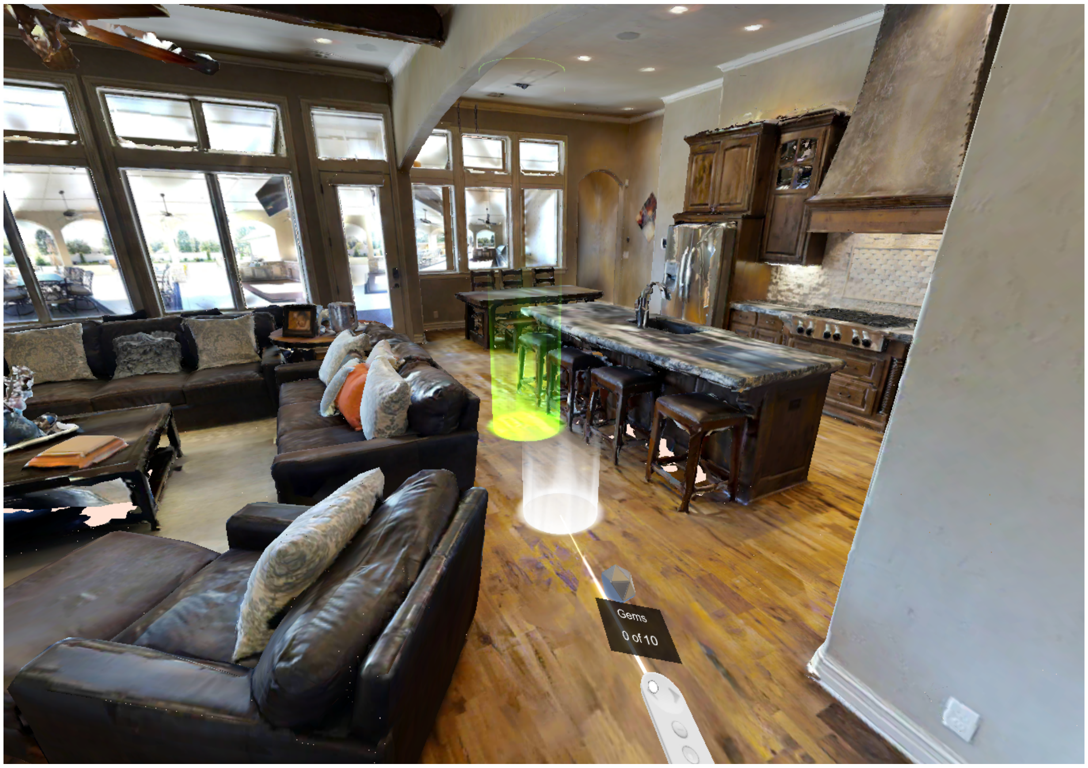


Figure 3.6: View of the tutorial environment. Users spawned in this environment and were provided with prerecorded interactive instructions on how to use the interfaces. This included teleportation/walking, collecting the gems and answering the pointing questions. The green glowing spot is shown to ask the user to teleport to that location. Users were verbally instructed through their headphones.

one would in equivalent brick and mortar places (see Figure 3.7). They started with a tutorial for either teleportation or natural walking. Their first tutorial taught them how to collect gems and answer questions at the end of each locomotion condition. These user interactions were not recorded nor considered in the analysis. Participants were allowed to take as long as they wanted in answering the questions, both in the tutorial and the experiment conditions. However, the actual locomotion task was limited to 5 minutes for each condition. The 5 minute time was derived from the amount of time required to walk through both the indoor and outdoor virtual spaces as determined by our pilot study. Since we did not want to integrate any self-motion or virtual environment manipulations (see Section 3.2), the indoor sports arena was the right sized physical space to run the walking experiment. The goal was to ensure participants had enough time to walk through the virtual spaces at least once as that would be the bare minimum required to build a mental map of the space. The same amount of time was used for teleportation to make for equivalent comparison. While teleportation allowed for exploring the space much more quickly than walking, participants had enough time to move around the space multiple times, motivated by the gem collection task, to potentially overcome the limitation of path integration seen in teleportation techniques [108, 105].

During the walking condition, the experimenter used a tablet device that ran the custom remote control web app. This allowed the experimenter to follow from a short distance away what the participant saw and did. At the same time they were in a position to prevent the user from walking into the gym boundary in case of tracking failure. Over the course of the experiment, we had to intervene and restart the tracking system a total of three times during the 16x2 trials. With each trial lasting 5 minutes, that amounts to only three tracking failures in 160 minutes of VR walking. For these cases, we resumed the experiment by asking users to close their eyes while we restarted tracking by realigning a virtual and a real wall, before they continued the experiment.

Typical tasks used to assess cognitive map building include sketching a map of the environment or pointing to *non-visible landmarks* in the environment [123]. We used a pointing task for our assessment. The tutorial was followed by the first scene that participants needed to explore, which was either an indoor or outdoor scene using either the natural walking or teleportation. After each scene, users were asked to complete a set of ten pointing tasks in VR (see Figure 3.8), fill the Slater-Usoh-Steed (SUS) [126] presence questionnaire, and the SSQ. Each locomotion technique was used in one outdoor and one indoor scene. The order of the trials was counter-balanced using a latin-square assignment by systematically varying the order in a full permutation, which was 16. Participants were asked to collect 10 gems scattered throughout each virtual environment

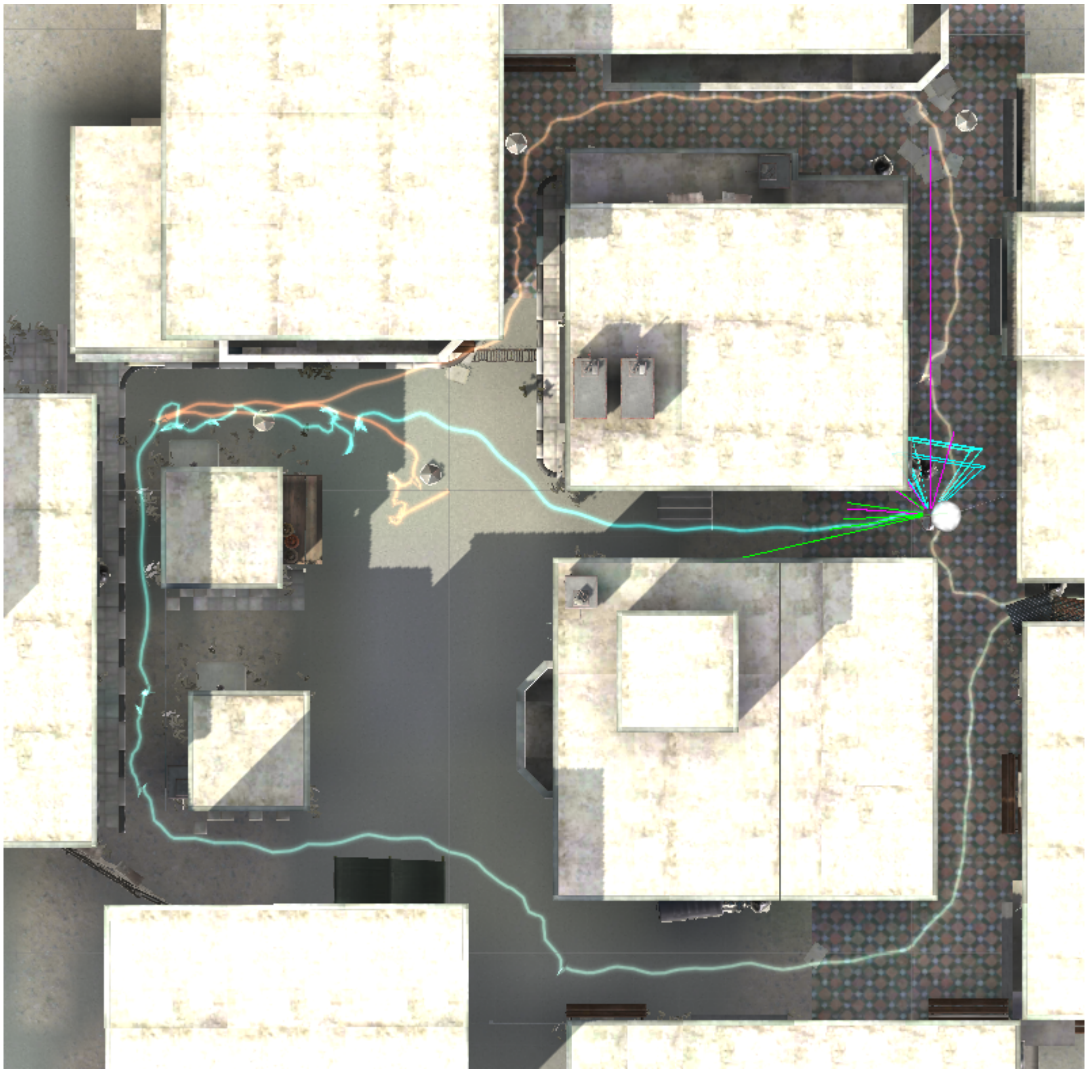


Figure 3.7: Top-down view of a user's walking path. Orange meaning old and Cyan meaning recent. Users would naturally avoid obstacles while they did not have any obstacles in the physical space. Lines are for the purpose of visualization and were not visible to users at the time of experiment.

though only 8 were placed in each scene. This was done to encourage exploration. The gem placement was balanced across the scenes in count and findability. Participants collected a gem by pointing a virtual laser at a gem and clicking a button on the controller. After the study, participants filled out a custom questionnaire that asked about their preference for a locomotion technique and any additional comments they had about their experience.

3.4.4 Assessment

In order to analyze the effects of the different locomotion techniques on cognitive map building, participants had to complete a VR pointing task [127, 123]. We had asked participants to pay attention to the virtual world they were exploring and additionally encouraged exploration through the gem collection mechanics. Now, after each condition, we presented a series of ten VR pointing tasks to gauge their mental map building. The virtual space was something we expected them to have learned in the five minutes of exploration time as shown in our pilot study. At the end of the exploration time, we presented participants with pictures of one part of the virtual world (see Figure 3.8) and asked them to point to the spot they thought the person taking the picture had been standing when taking the picture. Using this metric, 0 degrees meant perfect estimation of the virtual photographer's location while 180 degrees was maximum error. The error was calculated on the 2D projection of the 3D aim vector onto the ground plane.

For each pointing question there was a possibility that user did not have the chance to see the space appearing in the question. Each question consisted of two parts: the location we teleported the user to, and the picture we showed them. We define "validity" for each question as follows: The question is valid if the user has seen both the location they are being teleported to, and the view they are being shown on their controller. In order to make sure we assessed the results for only the valid questions, we performed a measurement after the experiment using the recorded movement data. We checked for each frame if the user had had a direct line of sight with any question location and whether they had been within a threshold of three meters from that point. We also took the same steps for the location of the virtual camera that created the target view, plus

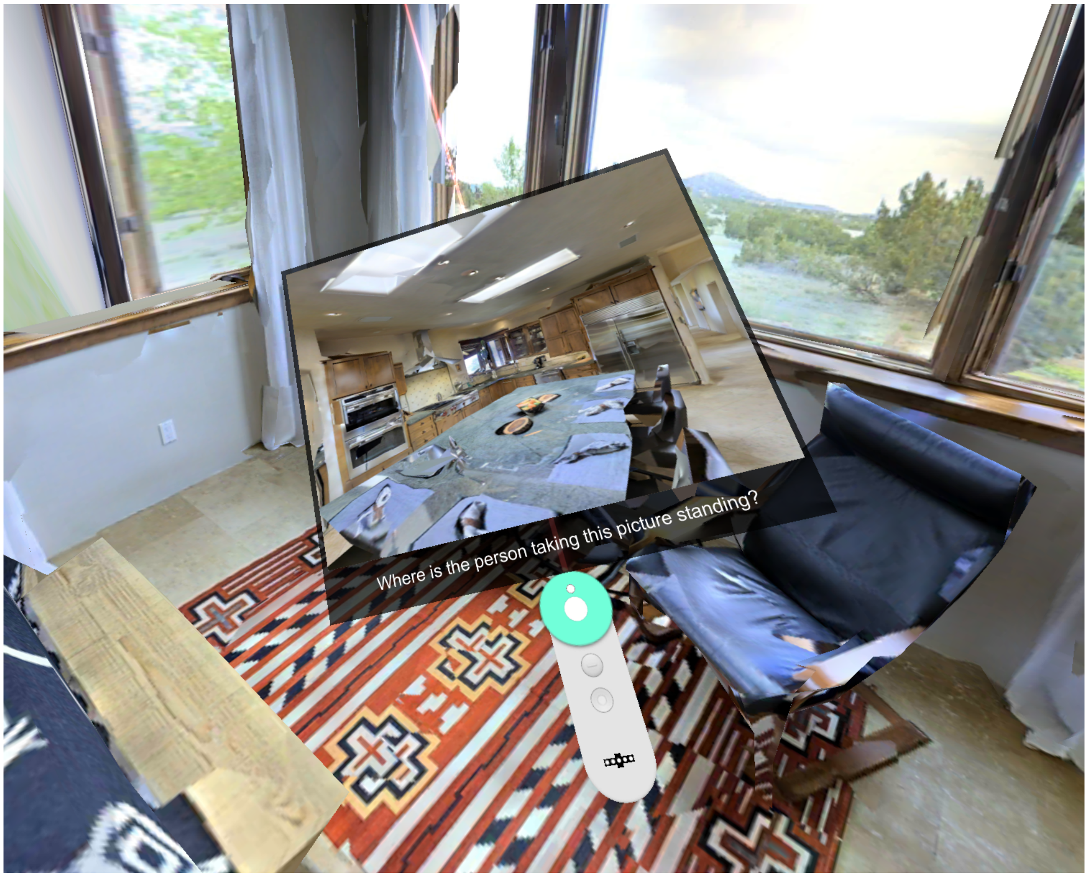


Figure 3.8: An example pointing task question. The participant was teleported to a specific spot in the scene and shown a picture of a target point in the scene not currently in view. They were expected to look around and point in the target direction.

checking if the user has had a similar angle view (angle difference threshold of 45 degrees). This ensured that the user had a chance to see the view depicted in the question. At any frame in which these values were true all at once for location or view, we considered that location or view “valid” and recorded it with a flag variable. When analyzing the data, we excluded the questions that did not have a valid flag for both location and view.

Participants were asked to stand and not move while answering the questions. Looking around was allowed and encouraged. Pointing to the target view locations from within the virtual environment required the participants to update their mental map of the virtual space with respect to each picture, especially because participants were

Table 3.1: Wilcoxon signed-rank test on Simulation Sickness components by Locomotion type (W for Walking and T for Teleportation). Listed are the Simulation Sickness medians, critical z, and p values. Participants experienced less simulation sickness while walking.

SSQ Component	M W	M T	z	p
Total Score	16.83	24.31	2.012	0.044
ΔTotal Score	1.75	11.92	2.012	0.044
ΔSSQ N	-1.49	4.77	2.055	0.040
ΔSSQ O	2.84	11.13	1.834	0.067
ΔSSQ D	3.48	16.96	1.965	0.049

moved to a different virtual location in the scene before each picture was shown. Having a mental map of a place implies knowing the relative placement of the various spaces. For example, at home, one is easily able to point to where the kitchen is when standing in a bedroom and vice versa.

3.5 Results

Our dependent variables included self-reported simulator sickness, self-reported presence, accuracy from post-trial pointing tasks, coverage of observed area, and finally, stated preference between locomotion methods.

We now report, in turn, on the results for each of these metrics.

3.5.1 Simulator sickness

Before the first locomotion trial, we collected a simulator sickness baseline assessment from each participant, using the SSQ questionnaire.

From the questionnaire answers, we compute the four metrics of Nausea-related subscore (N), Oculomotor-related subscore (O), Disorientation-related subscore (D), and SSQ Total Score (Ts) according to the scoring functions for SSQ [124]. After exploring each scene, via either walking or teleportation, participants were asked to fill out the SSQ again. The average difference values to the baseline for each of the four metrics, split up by locomotion technique, are summarized in Figure 3.9.

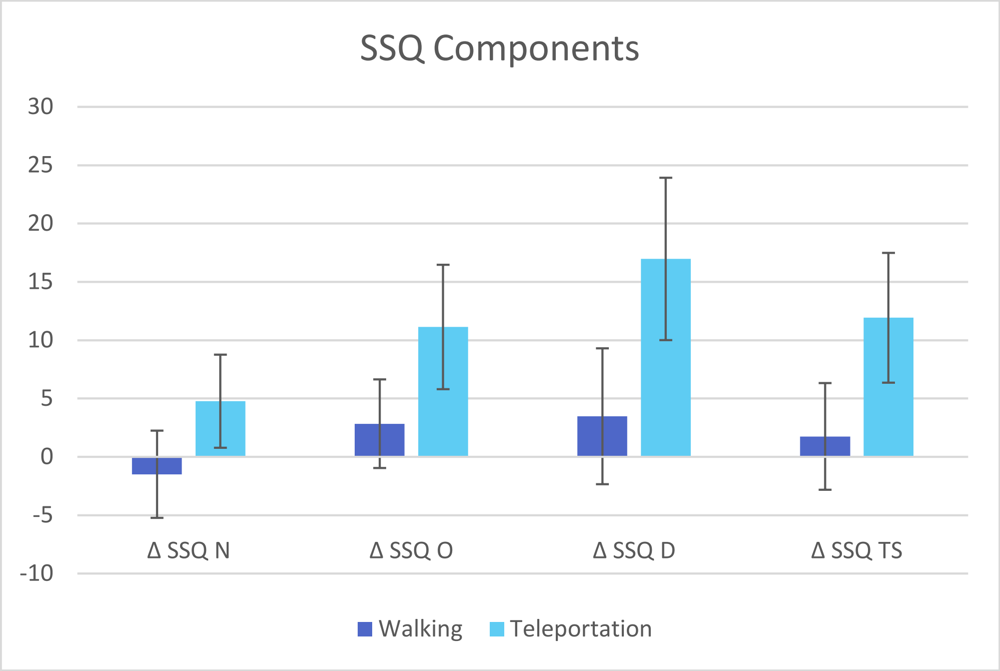


Figure 3.9: Mean values for SSQ Questionnaire components measured as deltas to a baseline condition before the start of the experiment, split by locomotion technique. Error bars depict standard error.

The distribution of these deltas not being Gaussian, as determined by a Shapiro-Wilk test of normality, we performed multiple Wilcoxon signed-rank tests on the four SSQ metrics to check for any statistically significant difference between teleportation and walking. There were significant differences between walking and teleportation for Total Score (Ts), Nausea-related subscore (N), and Disorientation-related subscore (D) (see Table 3.1 for medians, z, and p values), with walking leading to lower reported simulation sickness. The delta means in the Oculomotor-related subscore (O) were narrowly not significantly different between locomotion techniques ($p = 0.067$). However, teleportation did induce significantly higher delta SSQ Nausea (N) and Disorientation (D) subscores ($p = 0.04$ and $p = 0.049$ respectively), resulting in a significantly higher Total Score (Ts) than natural walking ($p = 0.044$).

These results confirm hypothesis H2 and even indicate higher overall simulator sickness for teleportation.

3.5.2 Presence

After experiencing each scene, via either walking or teleportation, participants were asked to fill out the Slater-Usoh-Steed (SUS) presence questionnaire. As suggested by Usoh et al. [126], the SUS score was computed as the number of answers r out of n that have a score of '6' or '7' on the 1-7 Likert scale. We additionally report the average value of all SUS questions in Figure 3.10.

Results from a Wilcoxon signed-rank test show no significant difference in SUS score between the walking and teleportation conditions ($z = 1.653$, $p = 0.098$). A plot (Figure 3.10) and analysis of the raw averaged SUS questionnaire scores may however indicate a trend towards better self-reported presence for Walking: the averaged raw SUS questionnaire responses were significantly higher ($z = 2.534$, $p = 0.011$) for walking compared to teleportation (see Table 3.2).

Overall, these results provide some limited evidence for hypothesis H3, but additional studies are needed for a more clear confirmation.

Table 3.2: Wilcoxon signed-rank test on overall averaged questionnaire score from the SUS Questionnaire by Locomotion type (W for Walking and T for Teleportation). Listed are the Presence medians, critical z, and p values.

Presence Component	M W	M T	z	p
SUS score	2.25	1.00	1.653	0.098
SUS Likert answer mean	4.53	4.00	2.534	0.011

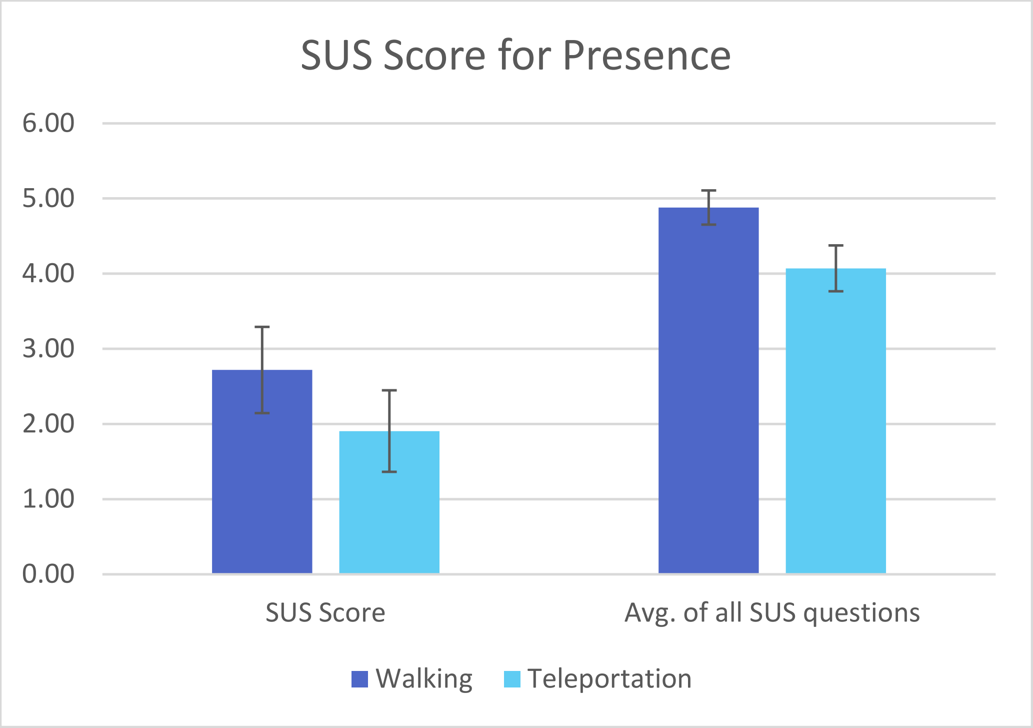


Figure 3.10: Mean values for differences in SUS Questionnaire, categorized by locomotion. Error bars depict standard error.

3.5.3 Pointing tasks

A Friedman test was run to determine if there were differences in Pointing task error in each of the four different scenes($\tilde{\chi}^2(2)$ = 16.200, p =0.001). Pairwise comparisons were performed with Bonferroni correction for multiple comparisons. Post-hoc analysis showed that effect of scene on the results from the aiming task is statistically significant. Pointing errors are significantly different between "Indoor 2" and "Indoor 1" (p = .006) and between "Indoor 2" and "Outdoor" 1(p = .006), indicating that in spite of our attempt to create two comparable versions each of indoor and outdoor scenes, one scene and/or pointing task quiz in particular ("Indoor 2") was easier to make sense of and answer.

Walking users had a mean pointing error of 33.95° and teleportation users had a mean pointing error of 34.21°. Wilcoxon signed-rank tests showed no significant differences in pointing task error for walking vs. teleportation (see Table 3.3).

Table 3.3: Wilcoxon signed-rank test on effect of locomotion on pointing error. There was no significance between locomotion techniques.

Dependent Variable	M W	M T	z	p
Pointing Error	33.95°	34.21°	0.052	0.952

3.5.4 Coverage of virtual scene

We calculated view coverage using the collected interaction playback after the experiment by casting 20 rays at 0.1 second intervals from eye to the scene. We placed a particle where the ray hit the scene and then divided the covered area by the total visible area of each environment. The resulting metric was consistent among all users and provided a measure for coverage. A significant view coverage difference according to a Wilcoxon signed-rank test of coverage percentages between the walking and teleportation conditions is only evident for Scene "Outdoor 1", with the walking condition producing significantly higher coverage for that scene (see Table 3.4 and Figure 3.11). Wilcoxon signed-rank tests did not show any significance in difference of coverage for Teleportation vs. Walking for the rest of the scenes.

Table 3.4: Wilcoxon signed-rank test on our coverage factor by locomotion type (W for Walking and T for Teleportation). Listed are the coverage medians, critical z, and p values. Only scene "Outdoor 1" showed significant difference.

Scene Coverage factor	M W	M T	z	p
Indoor 1	0.0208	0.0208	2.012	0.674
Indoor 2	0.0199	0.0196	2.012	0.208
Outdoor 1	0.0243	0.0225	2.055	0.012
Outdoor 2	0.0241	0.0253	1.834	0.327

The combined results on pointing tasks and virtual scene coverage do not either confirm or reject hypothesis H1. However, it is noteworthy that both techniques enabled participants to perform comparatively well on the pointing tasks (average of 34 degrees error, with 3 degree standard error). Likely, the 5 minutes exploration time gave plenty of opportunity for either locomotion technique to form reasonable mental maps. It is interesting that in spite of an overall speed advantage for teleportation, the only significance in difference of view coverage comes in favor of real walking.

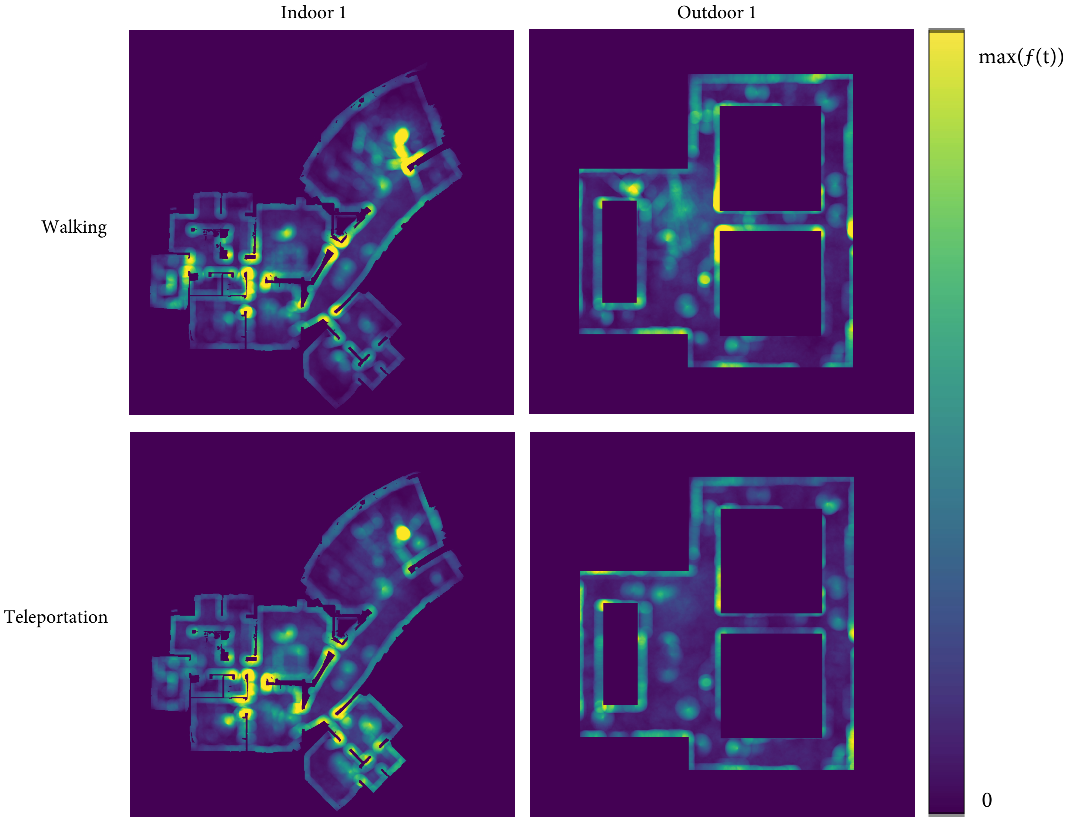


Figure 3.11: View Coverage map for 2 of 4 scenes in Teleportation vs Walking. View coverage is the accumulation of projected user's camera frustum on the 3D scene over time.

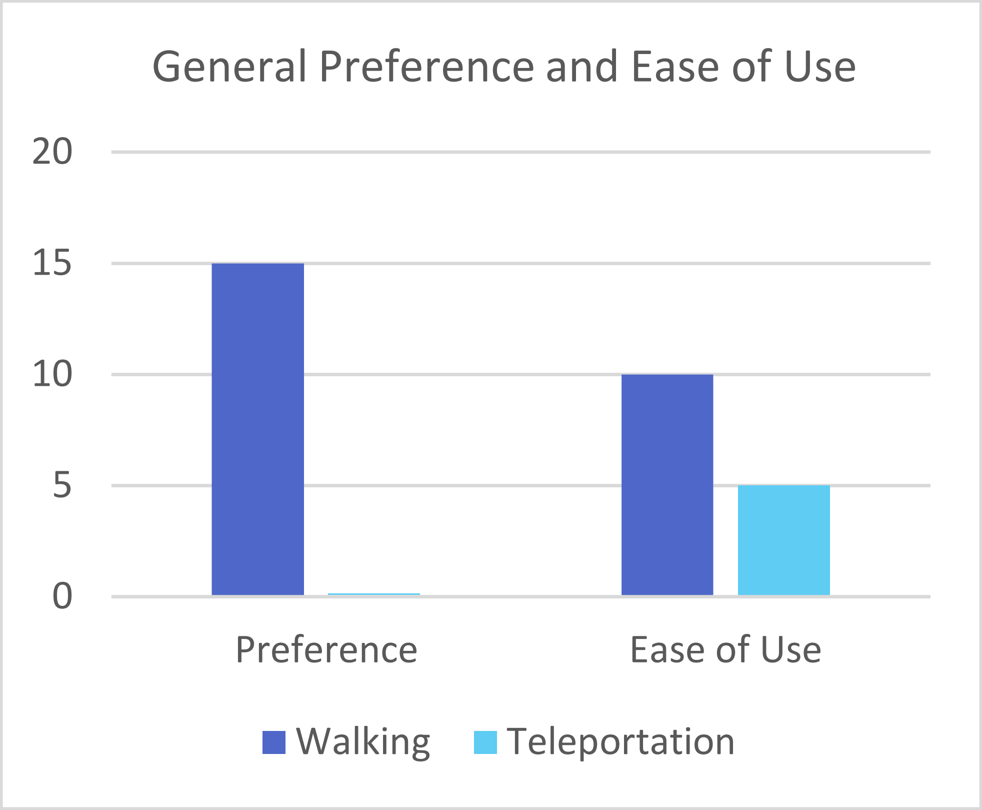


Figure 3.12: Number of responses choosing Walking or Teleportation respectively for overall Preference and Ease of Use. Walking was the preferred method among all the participants. Ten users considered walking easier to use than Teleportation (one abstaining).

3.5.5 Participant preference

In the post study survey, none of the participants reported a preference for teleportation over walking: 15 preferred walking, 1 did not answer. On the question “Which movement method did you find easier to use?”, ten participants chose walking and five users chose teleportation (see Figure 3.12).

3.6 Discussion

The results of our user experiment comparing two VR locomotion techniques for exploring indoor and outdoor virtual environments indicate an overall clear winner and many interesting discussion points. Natural walking was almost universally preferred over teleportation for navigating the house and an urban market environment. This was despite the large amount of walking participants did in the indoor sports arena. Teleportation led to significantly higher self-reported disorientation, nausea, and, consequently, total score components of the SSQ, and we observed a trend towards higher self-reported presence for natural walking.

While we did not find prior work that compares natural walking directly with teleportation, teleportation has been compared with joystick control [84, 83] and redirected walking [83] in smaller environments. In both these studies, teleportation caused less nausea than joystick and was preferred over joystick. It was comparable to redirected walking in terms of user preference [83] and there was no significant difference in presence between teleportation and leaning [84]. Our results showing teleportation to cause higher disorientation is thus, slightly unexpected, though unlike prior work our comparison is with natural walking in a large space. In what might be perceived as a diverging finding, Suma et al. [92] showed walking to cause higher nausea, occulomotor discomfort, and disorientation compared to a simulated walking technique in a virtual maze, which they described as a “navigationally complex environment.” Participants in our study walked in natural environments such as a single-family home and an outdoor market. While the virtual environments are not directly comparable, we believe advances in VR hardware technology may explain the different outcome for walking in our work versus the study by Suma et al. [92].

Previous research demonstrated benefits of real walking compared to more indirect and passive locomotion techniques mostly for smaller virtual environments, but for larger scale scenes there was a real possibility that improved speed and flexibility of a 'super-natural' technique such as teleportation, could yield some benefits for wayfinding and survey knowledge acquisition. At the same time, teleportation is a technique that is widely used in VR games, and reported presence also point towards natural walking as being preferable.

It was clear to us experimenters, both from the design of the methods and from observations of our users during the trials, whose paths through the various environments we could follow in real time through our monitoring app, that teleporting afforded faster navigation than natural walking, as participants could jump to any visible part of the scene in one fell swoop. However, since they had to collect gems, they needed to build a mental map of the respective environment in keep track of spaces they had already visited and collected a gem from versus other spaces. In fact, for one of our four scenes (Outdoor 1), Walking resulted in significant higher view coverage of the space within the 5 minute exploration. We interpret this to mean that while teleportation has the inherent ability to enable faster movement in the virtual space, it does not necessarily mean users always use it in quick succession. It is possible that other tasks (such as, e.g., path-oriented wayfinding in non-convoluted environments) would better bring out advantages of teleportation.

Overall, our results do not present any absolute red flags for the teleportation method. Some participants reported teleportation as easier to use (possibly alluding to less physical effort needed). While there were significant differences in self-reported simulator sickness deltas, overall symptom scores remained low for both conditions. It is worthwhile to point out that according to our results, eye-strain is not at the root of the increased simulator sickness for teleporting, as the O score (Oculomotor-related subscore) deltas were not significant. The Nausea-related subscore (N) was a main contributor and the

Disorientation-related subscore (D) may have contributed as well. It is possible that using a different implementation of teleportation like Dash [108] may have helped reduce the Disorientation-related subscore (D).

The results revealed some significant effects of the locomotion technique on cognitive map building. Regarding the pointing task and scene coverage percentages, there were some differences between techniques, again slightly favoring walking. This suggests that the choice between the locomotion techniques may impact mental map making ability. Even though people start making a mental map of a new place as soon as they arrive there [119], we encouraged exploration of the space through the collectible mechanic and participants were compelled to view the same places multiple times as they searched for the gems. This may have helped improve overall ability for both conditions.

3.7 Conclusion

As standalone VR headsets are now able, through inside-out world tracking, to track user pose in large physical spaces, new possibilities arise for mixed reality experiences that involve significant amounts of real walking. Few user studies exist that evaluate natural walking in virtual environments over such large areas as soccer fields or sports arenas.

We designed an experiment to test and compare natural walking and teleportation as two main types of locomotion for sports-arena-sized virtual environments, and our results indicate decided advantages of natural walking over teleportation in such larger environments in terms of user preference, induced disorientation, and some other metrics, including an observed trend towards higher presence and some isolated indications of better mental map formation.

Overall, this suggests that walking-based VR (or more immersive AR) in wide-area environments can become feasible and attractive for general audiences. Several technological advancements would be required to fully enable the educational and entertainment possibilities sketched in our introduction. Tracking would need to be more robust to challenging and varying indoor and outdoor conditions as our attempts to run this user study outdoors or within a completely unprepared sports arena were not successful. Headsets would need to be further miniaturized and their display capabilities expanded for comfortable immersion over extended periods of time along with the ability to seamlessly transition between AR and VR modes. For safe unattended VR walking, the user would need awareness of the physical world while maintaining immersion in VR as demonstrated by our design prototype from Section 3.3. Battery life and heat optimization would have to be improved, and the use of headsets in direct sunlight would need to be made effective and safe. But even in the absence of the technological breakthroughs needed for truly robust real-world implementations of wide-area VR walking, our work demonstrates that we can already define and successfully evaluate user experiences in this domain.

Chapter 4

Machine Learning for Visual Content Generation

Machine learning is a subfield of artificial intelligence that arises from pattern recognition. It's based on the premise that computers can learn to perform specific tasks without using explicit instructions by relying on patterns and inference. The iterative aspect of machine learning is vital as ML models can adapt when exposed to new data. Though being very popular in recent years, machine learning is not a new field. It has been a research topic in AI and information theory for decades. However, recent advances in computational power with the rise of GPUs and data accessibility that has come with the internet have given this field a new life.

In this chapter, we first discuss various areas where generative machine learning models are used. We then introduce DeepDive which is our initial work using the combination of machine learning and an approachable user interface. We then present our work content aware semantic editing and inpainting (CASEIn) which is a generative adversarial network (GAN) designed to fill missing regions of an image, given the labels for those missing areas.

4.1 Introduction

Machine learning has been getting a lot more attention with the recent improvements in parallel computing power and the availability of large-scale datasets. Many of the old algorithms, like backpropagation and gradient descent, could not be utilized since they were computationally expensive. But nowadays, different types of deep neural networks are being developed thanks to the fast hardware, the extensive amount of data, and numerous machine learning APIs. An ML model can be a neural network that is trained on existing data using backpropagation to estimate a function often referred to as the probability distribution function or PDF. PDF is the function that describes the distribution of a dataset. This makes ML be by nature, an excellent tool for regression and data interpolation. And also, that's why an ML model is, at best, as good as the training data [128].

The revival of neural networks under the name of deep learning has dramatically improved the ability of ML methods to learn models from big data. An extensive set of architectures, including Markov Models, Auto-Encoders, Generative Adversarial Networks are developed and used. These methods also show significant success in generating speech, text, images, textures, and 3D models. In this section, we explore different topics researchers use ML and specifically neural networks to complete missing visual and spatial data.

4.1.1 Inpainting

Image inpainting is the reconstruction of the degraded parts of the photos and videos. This should be done in an imperceptible and nondestructive way. Image inpainting can be used to reconstruct damaged images or to delete or replace selected objects in a picture. This is not limited to 2D images. 3D photography is an emerging technology in recent years due to advances in sensor technology. Depth cameras can be found on consumer mobile phones, and they often produce incomplete results due to inevitable sensor issues. The damaged and covered parts that we want to recover in the image are often called holes [129].

Recent Literature reviews categorize inpainting in three different areas: sequential-based approaches, CNN-based approaches, and GAN-based approaches [130]. We will explore these methods a little more extensively since some of these concepts occur in other subsections as well. Two main approaches in Sequential based are patch-based and diffusion-based. Path-based methods fill the missing information by searching through the image and finding a relevant patch that can create a coherent hole filling. Many methods are introduced to find the best patch for this task. A non-parametric method by Efros et al. [131] was introduced for synthesizing texture, which performed better than many of the previous model-based methods. Simakov et al. and Cho et al. Introduced Image reshuffling algorithms that allow the user to capture and move part of an image around. The algorithm then automatically fills in the holes, similar to the original image, taking into account the relocated part [132, 133]. Barnes et al. proposed using random sampling to find matches throughout the image and increased the speed to an interactive level [134]. Zongben et al. provided a method that works based on a novel patch priority metric on the sparseness of the patch's nonzero similarities to its neighboring patches. Jin and Ye introduced a method based on low rank structured matrix and annihilation property filter [135]. Several other patch-based inpainting methods have been introduced [136, 137, 138, 139, 140, 141, 142, 143].

Diffusion-based methods fill in lost areas (i.e.,, holes) by seamlessly spreading image content from the boundary area to the inside of the boundary. The earliest work in that area is introduced by Bertalmio et al. [129]. more recent works include: localization of diffusion-based inpainting [144], using genetic algorithm [145], fractional-order nonlinear diffusion [146].

Sequential methods show promising results worked in some types of image inpainting, such as filling narrow texture details. However, capturing the global structure of the image is still a difficult task [147]. Autoencoders are proven to perform really well for the inpainting task [148]. However, convolutional autoencoders usually generate blurry results. Using U-Net has been shown to increase the sharpness and quality of the inpainted area [148, 149]. Hsu et al. proposed using VGG architecture [150]. Chang et al. [151] Introduced Video Object Removal Network (VORNet), which combines optical flow warping and image-based inpainting models.

Generative adversarial networks or GANs introduced by Goodfellow et al. [152] are capable of producing very sharp results. This makes them a popular method in inpainting and many other fields that rely on a high-quality generative network. GANs consist of two feed-forward networks. A discriminator network (D) and a generator network (G). where the generator's role is to produce realistic-looking results and discriminator's role is to determine if a sample is real or fake (generated by the generator network). This relationship is usually depicted as police vs. counterfeiter. The better the police gets, the more advanced the counterfeiter has to get in order to survive. The objective function of a GAN can be described as a minimax game between the discriminator and the generator the See equation (4.1). Where z is random noise vector and x are the real images sampled from the p_z and real data distribution p_{data}, respectively.

The GAN-based methods use a coarse-to-fine network, and the contextual attention Unit provides good performance and has been shown to be useful for inpainting [153, 154, 155, 156]. Dong et al. suggested a deep convolutional generative adversarial network (DCGAN) for the inpainting task. Pix2Pix proposed by Isola et al. showed success in performing inpainting tasks by using a U-Net and PatchGan. Chen et al. proposed a GAN-based semantic inpainting method called progressive inpainting [154], where a hierarchical strategy was implemented from low-resolution images to higher-resolution images to fill the holes in the images. [157].

$$\min_{G} \max_{D} V_{gan}(G, D) \tag{4.1}$$

$$V_{gan}(G, D) = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{data}} \log D(\mathbf{x}) + \frac{1}{2}\mathbb{E}_{z\sim p_z} \log(1 - D(G(z))) \tag{4.2}$$

4.1.2 Panorama completion and view extrapolation

In View Extrapolation, The camera position is considered stationary, and the image is extended by either increasing camera field of view or rotating the view. If a full 360° reconstruction is intended, the original image is remapped to a portion of a set of 360° coordinates like equirectangular mapping. Then empty pixels are filled using a variety of techniques. Earlier work in this area focuses on using patch-based synthesis [158, 159, 160] and shift maps [161], which are borrowed from inpainting research. Such a problem will be solved more efficiently if external data is taken into consideration [162]. More recent works are focusing on using big data and neural networks for this task.

4.1.3 Neural rendering and novel view synthesis

In classic computer graphics, rendering is the process of showing a set of data on the screen to create an image. This process can happen through different pipelines. Modern real-time rendering is often using rasterization techniques, while offline rendering is usually performed by tracing rays from the camera to the scene. Neural rendering, however, focuses on generating an image of a scene directly from a neural network. This field is relatively new and immature for practical use cases, but It's essential to mention as it somehow mimics the way we humans see. We do not see with our eyes, but rather with our brains. Eyes are only the “input” part of our complex visual system.

Research in neural rendering divides into two major blocks. Complete end to end neural rendering in which a deep network called a representation network observes a set of input images and camera parameters to create an encoded version of the space and a generator creates prediction based on query camera parameters [16, 163, 164]. The second group of works use neural renderers as a final step of a handcrafted system. The renderer could be a conditional generator that receives an already rasterized unshaded view called a buffer and shade it based on learned priors [165, 15, 166].

4.1.4 3D scene completion

3D data gathered from sensors or through reconstruction algorithms are usually incomplete in some areas. Especially the areas that are hard to reach for depth sensors, laser scanners, and cameras. Some methods also have problems with dark featureless, highly reflective, or transparent surfaces. Lack of spatial information in these data can show up as holes in a point cloud or holes in a reconstructed triangular mesh. This problem can be roughly described as inpainting for 3D surfaces. A body of research focuses on solving these issues by applying surface heuristics, geometric methods, and data-driven methods. Just like in inpainting, the rise of machine learning has contributed to this field as well. 3D voxel CNNs [143, 167] and Graph CNNs [168] are the most used type of neural networks in this area.

4.1.5 Depth from single image

The idea of computing depth from a single image dates back to the early stages of computer graphics and computer vision. It’s an exciting research field humans show they can infer a lot of spatial information just by seeing a 2D picture. We know that this spatial inference is thanks to a series of computations and sensory inputs. A good portion of this information comes from the rule of perspective. Objects appear smaller the further away they are. This by itself was the idea behind some early work in this area [169]. Comparing object sizes also is a factor. When seeing an image of a stapler by a cup, we most likely imagine that the camera was close to the scene. This type of inference comes from the prior that staples are small. Therefore, if they are huge in a picture, they should be close to the camera. It's a combination of our brain, learning perspective, and learning object semantics. This tied relationship with object semantics made using deep neural networks a great candidate, which was already proven to perform quite well for semantic segmentation [170, 171, 172, 173]. Later on, Neural fields were used to consider the continuous characteristic of the depth values [174]. The current state of the art uses Convolutional Autoencoders encoder a feature fusion and a refinement module [175], A pattern that has been shown to be effective in the super-resolution field.

4.1.6 Inverse rendering and spatially variant BRDF estimation

Inverse rendering is, as the name suggests, is the reverse process of rendering task, which means giving an image as an input, extract complete lighting, camera, and scene description (geometry and shading). As expected, it's a complicated process that usually consisted of multiple steps. There is a significant overlap between Inverse rendering and depth from a single image as Inverse rendering tries to solve the geometry while solving for BRDF, lighting, and camera intrinsics.BRDF or bidirectional reflectance distribution function, is the function that describes surface reflectivity distribution in each incoming light direction. BRDF in Layman's terms, is the surface material. Spatially variant BRDF is when surface material has different properties in different locations, which is the case in most real-world scenarios. Research in this area usually tackles rendering as an optimization problem. You have a complex function that generates a rendering, and you optimize the input to match that output. These types of functions need to be differentiable in order to get optimized more efficiently. That's why most of the recent research in this area uses different kinds of differentiable renderers [176, 177, 178].

4.1.7 AI based 3D synthesis

Synthesizing 3D scenes using AI is more of a feature or use case rather than a specific area of research. Though considering the sparse input problem, It makes perfect sense to include it in this list. Creating 3D models and environments is a great example of data extrapolation. Rules and heuristics can be extracted from sparse inputs or rich datasets, and using traditional computer graphics techniques and a library of 3D models and textures, an environment or a 3D model is produced. The first steps regarding the heuristic and guideline extraction can include classic computer vision [179, 180, 181, 182], natural language processing [183] , 2D convolutional neural networks [184]. In more recent years, GANs have also been used for this purpose [185, 186].

One specific area worth mentioning is layout estimation using neural networks. RoomNet is the initial work that uses this end to end approach [187]. Many other works [188, 189, 190, 191] have been published that improve on RoomNet. Some can also estimate the furniture position and create a 3D model of the scene [192].

4.2 DeepDive

Super-resolution is a subfield in computer vision that focuses on increasing the quality of an image based on previous observations of similar images. There are many variations of super-resolution, and DeepDive is a domain-specific super-resolution system that relies on the fractal features of natural formations. In nature, many formations, like rocks and

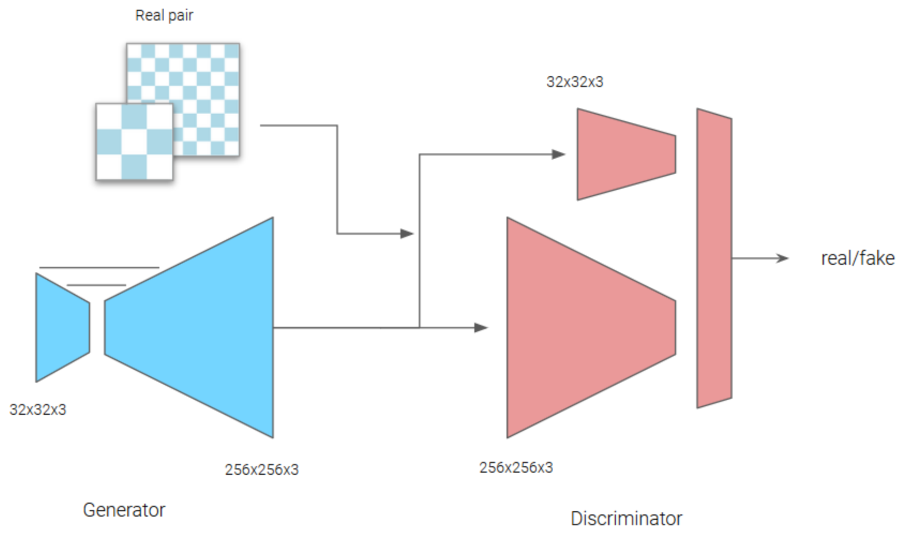


Figure 4.1: DeepDive network architecture.

mountains, have self-similarity. In the DeepDive project, we observed that by observing a set of low-resolution and high-resolution satellite images, we could create an infinite zoom effect and an imaginary landscape that users can explore indefinitely.

4.2.1 DeepDive ML architecture

DeepDive architecture is inspired by SRGan [193], which is a generative adversarial network. In consists of an asymmetric U-Net [194] that has a larger decoder. Which receives a 32 × 32 image and outputs a 256 × 256 image. The discriminator network is a fully convolutional PatchGan with patch size of 16 × 16 that outputs real vs fake for each patch (see Figure 4.1.)

The network is trained using the traditional GAN loss and MSE loss for the down-sampled output vs low resolution input.

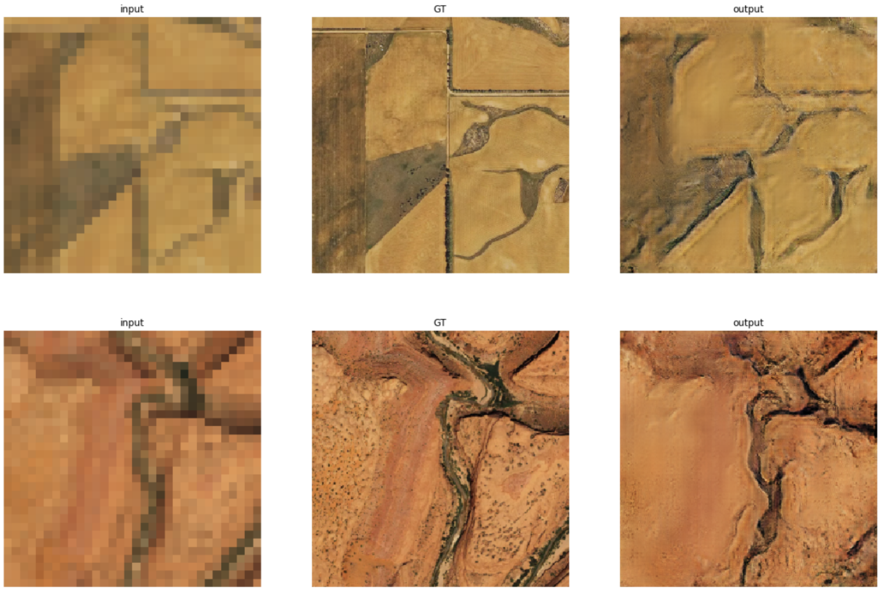


Figure 4.2: Some examples of super-resoltuion created using DeepDive network.

4.2.2 DeepDive interface

DeepDive's UI is a 3D user interface that mimics teleportation. Users can aim across an initial satellite image and choose a section they would like to dive into. the image of that section is sent via socket to the ML server, and the output is presented at any frame. By clicking the trigger button, the small section is then up-scaled under the user's feet, which creates a feeling of diving into the map 4.3.

4.3 CASEIn

Image completion, inpainting, and extrapolation are crucial computer vision and graphics techniques with a variety of real-world use cases. Filling in missing or obscured areas of an image is known as image inpainting, which also could be used for restoring or repairing damaged or deteriorated areas of an image. The process of extrapolation, on the other hand, involves extending an image's content past its original bounds. With

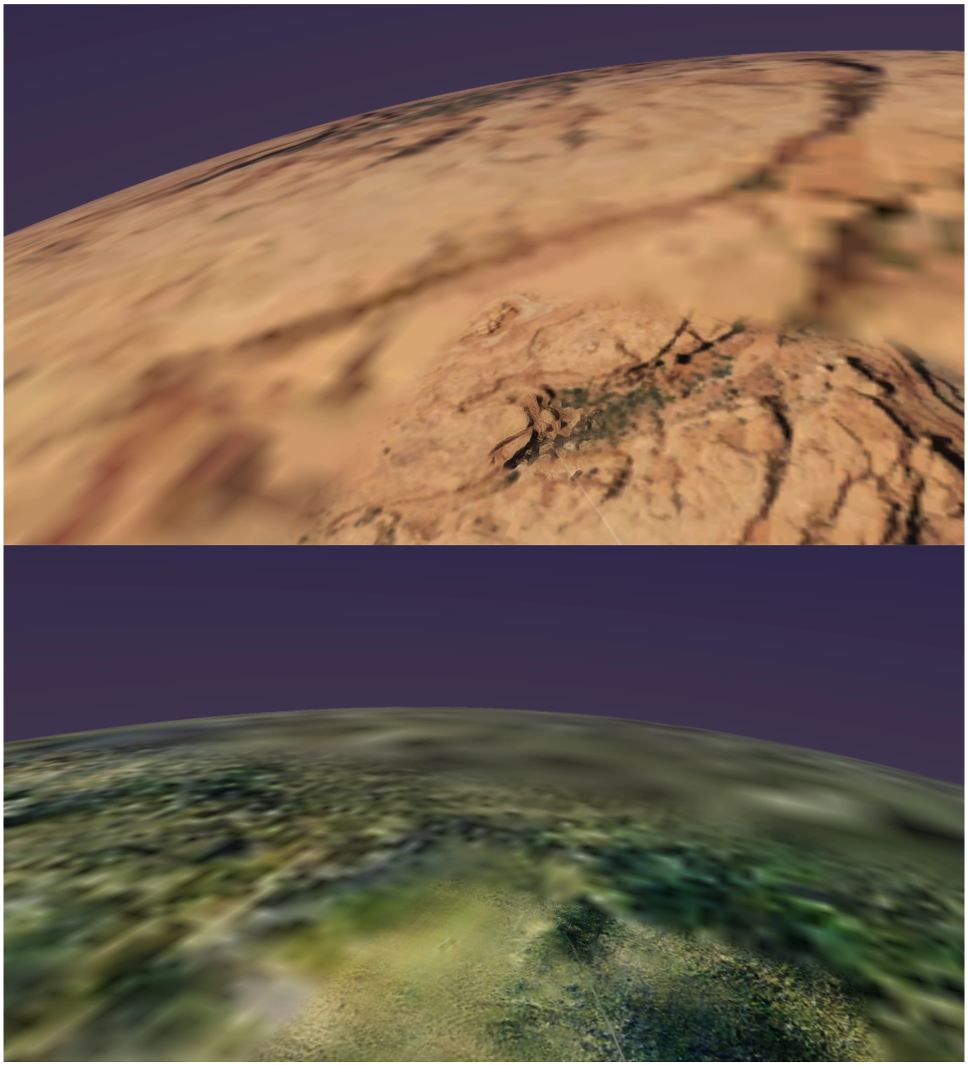


Figure 4.3: DeepDive 3D user interface in Virtual Reality.

the help of these methods, images can be made to be of higher quality for a variety of purposes, such as enhancing their visual appeal, making them more useful for tasks like changing the aspect ratio or allowing for novel image manipulation.

The development of machine learning-based methods for image extrapolation, inpainting, and completion has advanced significantly in recent years. Due to their ability to extract patterns and features from sizable datasets, these techniques have the advantage of producing results that are both highly realistic and semantically meaningful.

The need to handle large missing regions, maintain fine details and textures, and handle complex image structures are just a few of the issues that still need to be resolved in the development of these techniques. Despite these obstacles, the continual advancement of inpainting, and extrapolation techniques is crucial for a variety of applications in industries like computer graphics, image processing, and multimedia.

CASEIn is a generative adversarial network that tackles this problem in a novel way. It can respect existing RGB and produce images based on semantic labels while keeping the flexibility of style mixing. CASEIn consists of an encoder, a generator, and a discriminator network. The inputs to the system are partial RGB Images, masks, and partial semantic segmentation maps. the output of the generator is a completed RGB image. CASEIn makes use of a global style feature palette extracted by the encoder network and uses that to generate locally and globally coherent RGB based on existing RGB and semantic labels.

we use GANs in CASEIn due to the several advantages they have over other methods. They can learn a high-dimensional distribution of images and recognize complex image structures and patterns, GANs can produce results that are incredibly realistic and semantically meaningful. Additionally, they can handle sizable missing regions while retaining small details and textures. GANs are a desirable option for applications where it is crucial to perform image inpainting quickly and efficiently enabling real-time applications for the network.

4.3.1 Encoder

We use an encoder network to extract the style of the image. The goal for the encoder network is to process RGB pixels I_{RGB} and their corresponding semantic labels I_{label} and output a style vector W that can be used by the generator to reconstruct or inpaint an

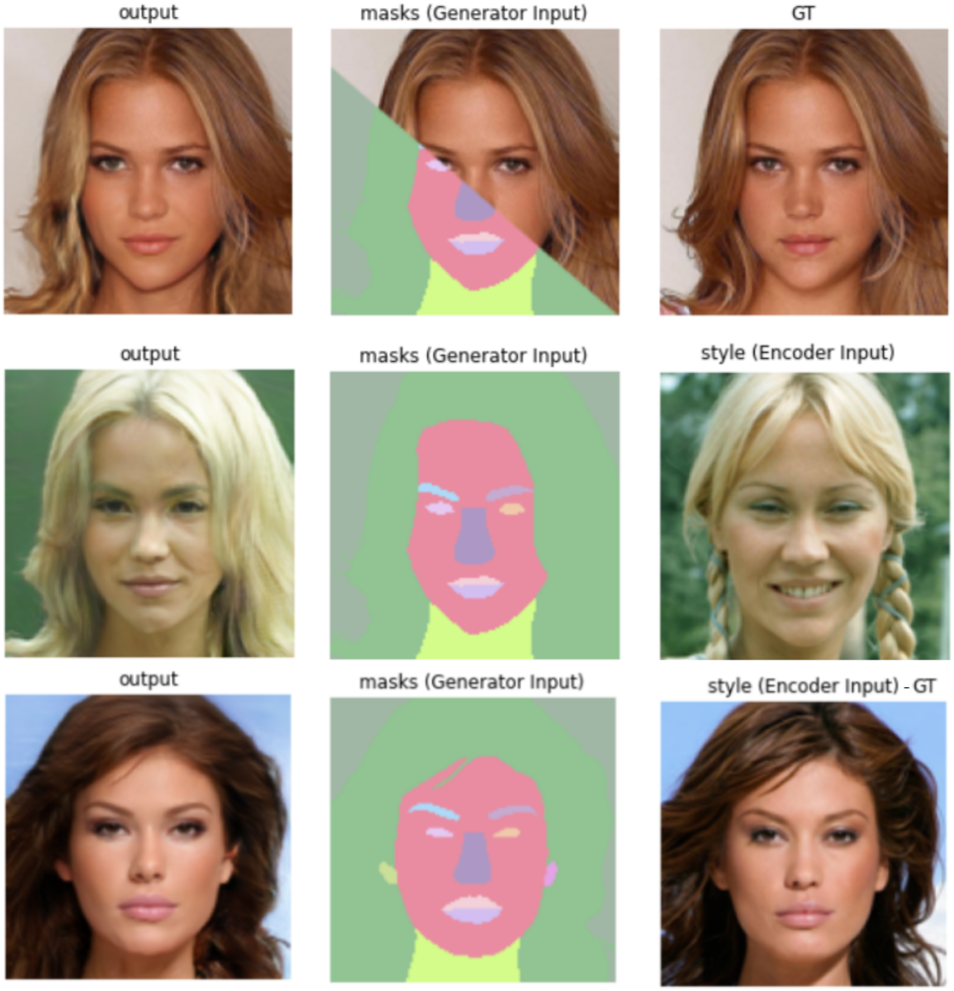


Figure 4.4: Example input and output to CASEIn generator. The first row shows an example of inpainting, the second row shows an example of style transfer, and the third row shows the reconstruction of the original image.

image. We are assuming that the existing RGB comes with labels, in our real tests we use a pre-trained segmentation model to generate labels for existing RGB pixels.

Due to the inherent similarity between StyleGan generator and SPADE-based generators and shown benefits of pyramid scene parsing for encoding style for StlyeGan [195], we use PSP to extract the style at multiple resolutions. The output of the encoder is a vector of size $N \times C \times$ K where N is the number of mip-maps, C is the semantic dimension and K is the number of classes in the dataset.

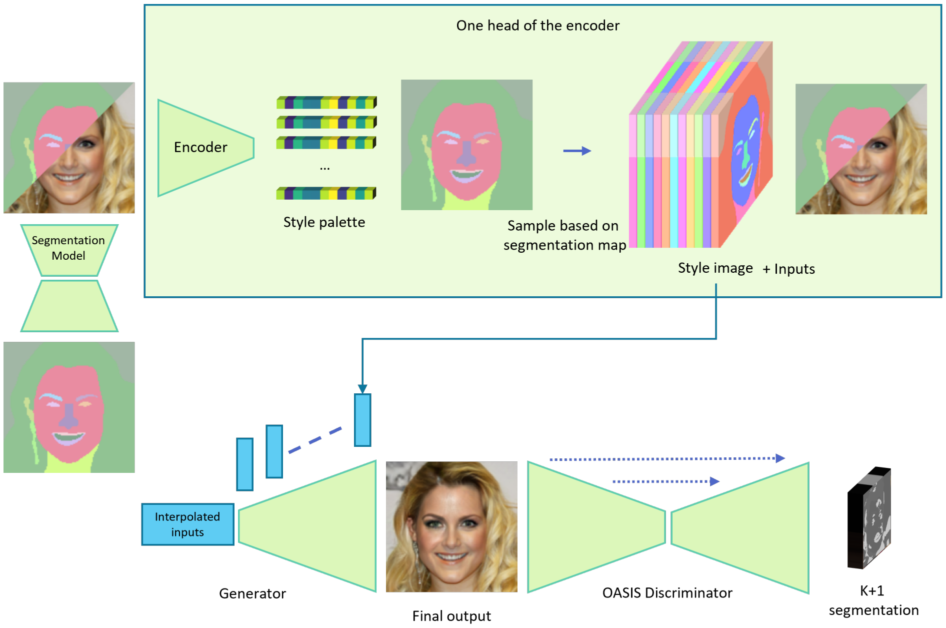


Figure 4.5: Overview of the CASEIn architecture. A pre-trained segmentation network creates a semantic mask for the existing RGB pixels. Then the multi-headed encoder creates a style palette based on the overlapping segmentation label and the existing RGB pixels. The style palette is then sampled into a set of 2D style images based on the existing segmentation map. These style images are then sent to the generator network to create the final completed RGB. the discriminator is actively trying to label these generated pixels as fake. and classify the real pixels as their respective segmentation class.

At each up-sampling step in the generator, the corresponding scale (N) of the encoded feature vector is used as input.

During inpainting, the number of input RGB pixels can vary between 1 and $h \times w$. traditional convolution layers are not invariant to this number thus leading to the network disregarding Input where the area is small. For example, our network needs to deduce the hairstyle even if a small portion of it is shown. To overcome this limitation, we use Partial Convolution [149] at each down-sampling stage.

We use SPADE layers to guide the inputs with semantics, we compare SPADE to concatenation in our ablation studies. Traditionally SPDAE has been used with up-sampling. We observe that paired with MaxPooling it can be an effective tool for feature encoding. To best of our knowledge, this has not been accomplished in the previous work. There have been multiple techniques to incorporate semantics and RGB, like the two-stream method in SESAME discriminator [196] or concatenation in Pix2PixHD [197].

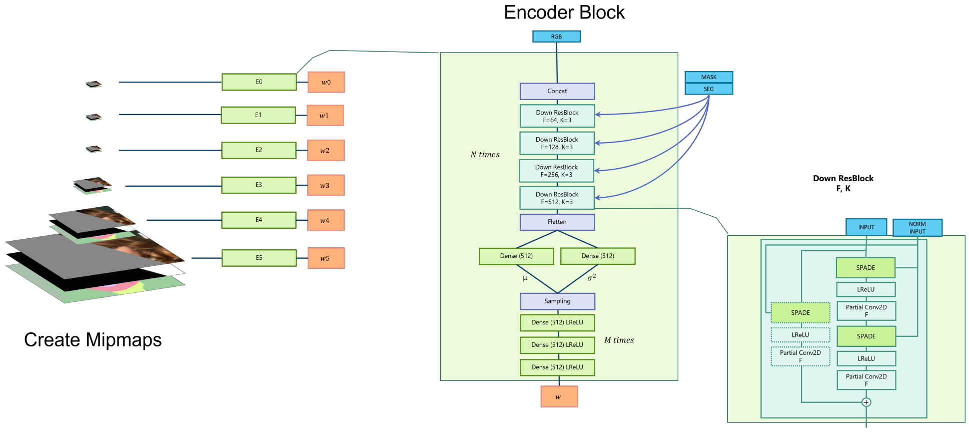


Figure 4.6: Overview of the CASEIn encoder. In each encoder body, Mask and partial RGBs are passed through partial convolution, while partial segmentation is used in SPADE layers. The network consists of N encoder bodies that each operate on a mip level. We observe that 4 mip levels are enough.

In each encoder body we use K number of Linear layers to disentangle the feature space, followed by two linear layers to output a mean μ and a variance σ^2 vector for each segmentation class. These vectors are then used to create a stochastic style vector by the reparametrization trick [198].

We call the collection style vectors generated by the encoder network the style palette. this style palette is then propagated into a 2D grid and then used directly in SPADE layers to guide the semantic synthesis process. We observed that using this method we can avoid using individual one-hot semantic segmentation images as the segmentation class can be embedded in the style vector itself, therefore, reducing the memory requirements.

The encoder network receives the as input and uses partial convolution layers [149] followed by a fully connected layer to output mean μ and variance σ^2. A noise vector then is sampled using these parameters that represent the style. To train the encoder We use the reparametrization trick and add a KL Divergence loss with the weight of 0.05 to the generator loss terms similar to previous work [198, 199], to backpropagate the loss.

4.3.2 Generator

CASEIn generator is trained jointly with the encoder network that generates style vector W. The generator starts with a concatenation of down-sampled mask, RGB and the first feature image, which is generated by the encoder that is passed through the first convolution layer. the output is then fed into a series of Spade residual blocks and up-sampling.

Each block starts with a global style block, two residual SPADE blocks, followed by an up-sampling layer. Similar to [20, 200] we output $W \times H \times 3$ RGB value for every CASEIn block to prevent vanishing gradients. These mip-map outputs are then fed to their respective layers in the discriminator.

The generator can be trained in MSG mode, where using a 1×1 convolution, we output an RGB image directly at every upsampling stage and send all those outputs to the discriminator. We try both MSG and non-MSG methods and need to run an official ablation as well. so far, at least in 128×128, MSG results are less reliable. This might change in higher resolutions as the subpixel sensitivity is less of an issue.

The generator loss needs to address three objectives. The output of the generator should match the semantic segmentation map. Generated pixels close to the mask boundary should have high coherency with their neighbors on the existing RGB pixels. And all generated pixels should follow the general style of the known RGB region.

We found that the best way to incorporate the encoded style and segmentation is through SPADE layers. The generator uses the pyramid-style vectors W generated by the encoder, and via a sampling, operation fills the 2D I_{label} to create I_{style}. I_{style} is then passed directly to SPADE residual layers [201] normalization inputs. We use 64 channels for the feature palette, and we found that enough for encoding the segmentation map as well so we removed the segmentation map from the spade layers, which saves memory for higher resolution attempts.

The existing RGB is overwritten over the final output, as we do not want any modifications to those pixels. In order to reduce a visible seam due to pixel level discrepancies, we erode and smooth the mask via a combination of max-pooling with the stride of 1 and a Gaussian kernel. This way, we get a smooth transition between GT RGB and generated RGB at the expense of losing some ground truth pixels near the edges. in MSG mode [200], this happens at every output stage. We also experimented with directly adding RGB to spade layers. However, we disabled this mode and relied only on the encoder. we observed that the results generated by only using the encoder are more consistent and the chances of producing seams in the output are lower.

$$\mathcal{L}_{G_{adv}} = -\mathbb{E}_{(z,t,m,x)}\left[\sum_{c=1}^{N}\beta_c\sum_{i,j}^{H\times W} t_{i,j,c}\log D(G(z,t,m,x))_{i,j,c}\right] \tag{4.3}$$

The equation 4.3 represents the adversarial loss for a generator network, denoted as $\mathcal{L}_{G_{adv}}$. The generator network is used to generate images that are meant to be indistinguishable from real images, and the adversarial loss is used to train the generator to perform this task. The loss is calculated based on the output of a discriminator network, denoted as $D(G(z,t,m,x))$, for generated images produced by the generator. The input variables z is the style noise from the encoder, t is the semantic label for the image, and m is the mask. These variables are passed to the generator network along with a real image x to guide the generation process. The loss is formulated as categorical cross-entropy, weighted by a factor β_c for each channel c in the output. The loss is calculated over the entire output of the discriminator network, which has a size of $H \times W$ for each channel

c. The total loss is the sum of the losses for each channel. The goal of the generator is to minimize this loss, which can be achieved by producing generated images that are classified as real by the discriminator network.

$$\mathcal{L}_G = \mathcal{L}_{G_{adv}} + \lambda_{recon}\mathcal{L}_{recon} + \lambda_{kl}\mathcal{L}_{kl} \tag{4.4}$$

The equation 4.4 represents the total loss for a generator network, denoted as $\mathcal{L}_G$. The total loss is a combination of three different losses: the adversarial loss $\mathcal{L}_{G_{adv}}$, the reconstruction loss $\mathcal{L}_{recon}$, and the KL divergence loss $\mathcal{L}_{kl}$. The adversarial loss is used to train the generator to produce images that are indistinguishable from real images, as described in the previous response. The reconstruction loss is used to ensure that the generated images are similar to the input image x, while the KL divergence loss is used to ensure that the distribution of the generated images is similar to the distribution of the real images.

The total loss is calculated as the sum of the adversarial loss, the reconstruction loss, and the KL divergence loss, with the reconstruction loss and KL divergence loss being weighted by factors λ_{recon} and λ_{kl}, respectively. The values of λ_{recon} and λ_{kl} can be chosen to balance the importance of the different losses in the overall loss function. The goal of the generator is to minimize the total loss, which can be achieved by producing generated images that are classified as real by the discriminator network, is similar to the input image, and follow the distribution of the real images.

4.3.3 Discriminator

We implemented a discriminator similar to [201]. The discriminator network has a U-Net-like [194] architecture, which is a PatchGan discriminator with a patch-size of 1 that outputs $K + 1$ classes for K segmentation and 1 fake class.

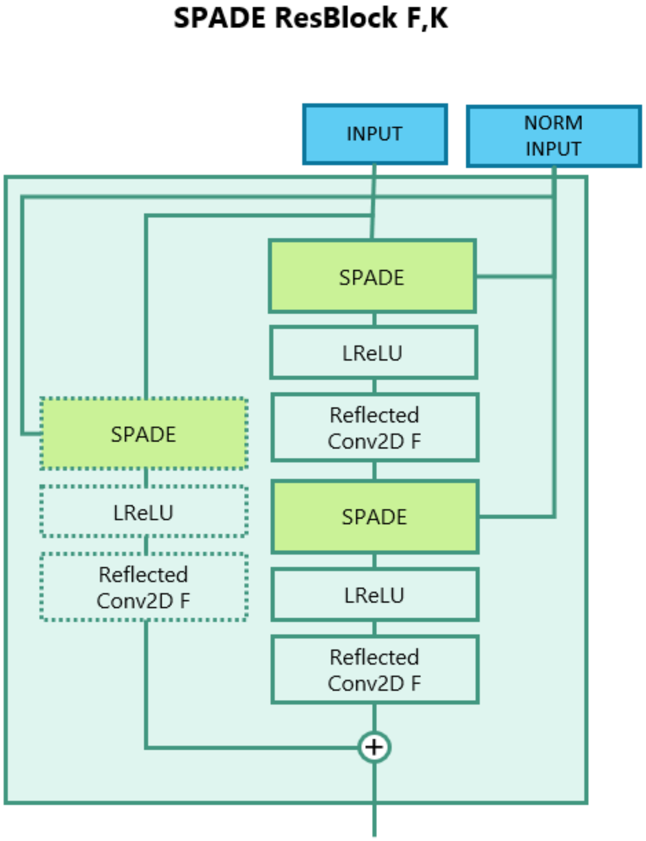


Figure 4.7: Architecture of SPADE residual blocks.(This is not our contribution so this might need to go.)

Similar to Shusko et al. [201] we use labelmix as a regularization loss. our modified labelmix includes an additional cross-label cut mask that helps with inpainting tasks where there is no guarantee that the inpainting mask is corresponding to object boundaries. The regular label mask is used in the image-to-image translation samples while the cross-label mixing masks are used for image inpainting samples.

$$\mathcal{L}_{D_{adv}} = -\mathbb{E}_{(x,t)}\left[\sum_{c=1}^{N}\beta_c\sum_{i,j}^{H\times W} t_{i,j,c}\log D(x)_{i,j,c}\right] - \mathbb{E}_{(z,t,m,x)}\left[\sum_{c=1}^{N+1}\beta_c\sum_{i,j}^{H\times W}\log D(G(z,t,m,x))_{i,j,c}\right] \tag{4.5}$$

The equation 4.5 represents the adversarial loss for the discriminator network, denoted as $\mathcal{L}_{D_{adv}}$. The discriminator network is used to distinguish between real and generated images, and the adversarial loss is used to train the discriminator to perform this task by predicting a K+1 class label for the image where K is the number of real classes and the additional 1 label is the fake class. The loss is calculated based on the output of the discriminator network, denoted as $D(x)$, for real images x and generated images $G(z,t,m,x)$, where z and t are the noise and label map pair, and m is the mask image. The loss is calculated as the negative expected value of the logarithm of the output of the discriminator network, weighted by a factor β_c for each channel c in the output. The first term of the equation represents the loss for real images, while the second term represents the loss for generated images. The loss is calculated over the entire output of the discriminator network, which has a size of $H \times W$ for each channel c. The total loss is the sum of the losses for each channel.

$$\mathcal{L}_D = \mathcal{L}_{D_{adv}} + \lambda_{cons}\mathcal{L}_{cons} \tag{4.6}$$

Equation 4.6 represents the total loss for a discriminator network, denoted as $\mathcal{L}D$. The total loss is a combination of two different losses: the adversarial loss $\mathcal{L}D_{adv}$ and the cutmix+labelmix constraint loss $\mathcal{L}_{cons}$. The adversarial loss is used to train the discriminator to distinguish between real and generated images, as described in the first and second responses. The cutmix+labelmix constraint loss is a regularization term that helps to improve the generalization ability of the discriminator by encouraging it to be invariant to certain image transformations.

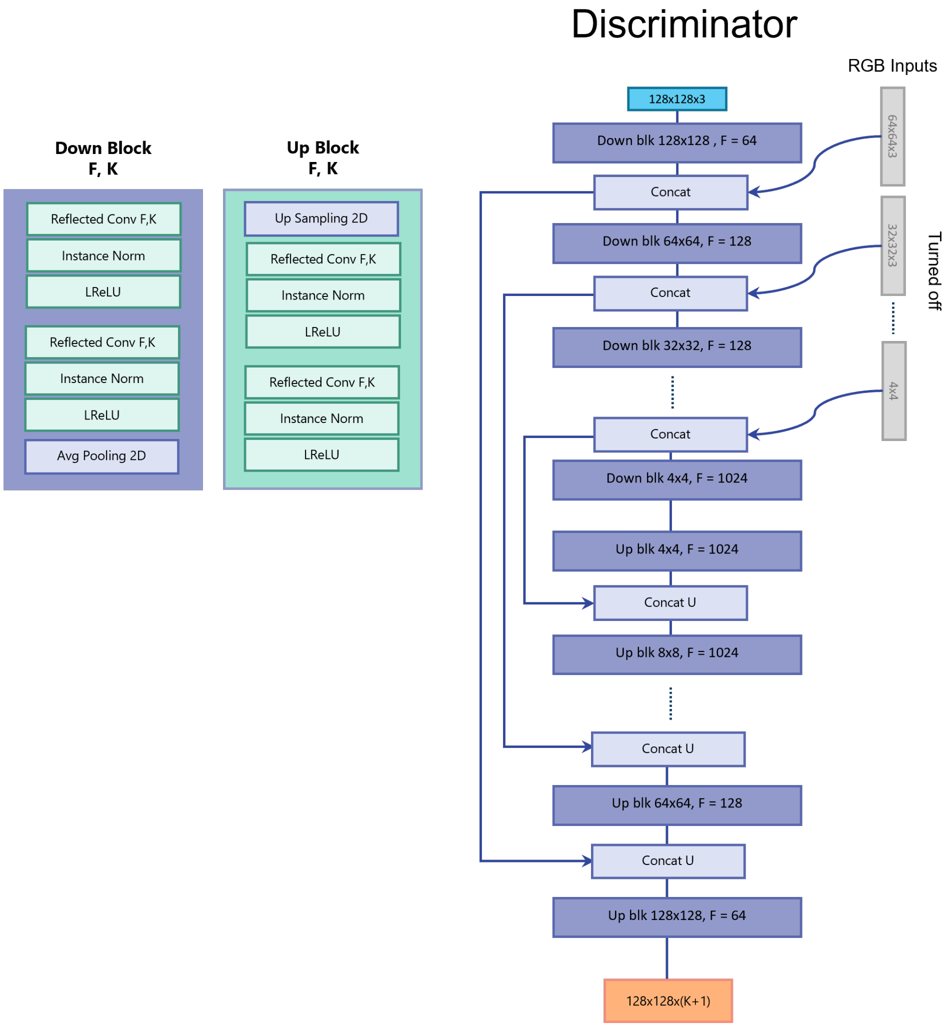


Figure 4.8: CASEIn discriminator. Multi-scale inputs are incorporated with concatenation, similar to the U-Net concatenation itself.

4.4 Experiments

We designed our experiments around the unified objective of guided inpainting and image completion. We did so by defining the image-to-image translation problem as an edge case for guided inpainting where there are zero pixels with existing RGB values. By doing so, we could train a single network that is capable of doing both tasks without a loss in quality.

We purposefully avoid using the segmentation maps as our mask since it's way easier for the generator to fool the discriminator if mask and segmentation maps align. The harder case for image inpainting is when there is a straight line across a segmentation map. This happens very often in image extrapolation since RGB images are captured in a rectangular frame that creates straight lines across the image boundary. Therefore, We randomly generate masks that are uniform and triangular or rectangular shape. We observed that a model trained on these scenarios performs equally well on more organic masks. Like when a user overrides the RGB by drawing a segmentation map and a mask simultaneously.

Our model is trained end-to-end. For each I_{RGB} and I_{Label} in dataset mini-batches we generate a random 2D mask I_{mask} of 0s and 1s that is either a randomly transformed polygonal shape or completely empty (0.25 empty vs 0.75 partial). This number gives us a 1-1 portion of overall real vs. fake pixels.

In order to train inpainting and image-to-image translation simultaneously, we prepare two sets of inputs, one for the encoder and one for the generator. In case of inpainting, the input to the encoder is partial RGB and partial overlapping label and the input to the generator are the same partial RGB, partial Labels and mask, and style output from the encoder. In the case of image-to-image translation, the input to the encoder is the complete RGB and Complete overlapping labels but in the generator, the inputs are the style output from the encoder, complete labels and 0 s for both mask and RGB.

We combine these two cases relying on whether the random mask I_{mask} is completely zero or not. In each minibatch, we generate a set of stochastic and deterministic outputs. for the stochastic outputs, we use both adversarial loss reconstruction loss and KL divergence, and for the deterministic outputs, we only use adversarial loss. Algorithm 1 shows the training loop for CASEIn.

Algorithm 1 Training loop

```
for $I_{RGB}$ and $I_{Label}$ in dataset minibatches do
    initialize $I_{mask}$ as random masks that are 75% partial and 25% completely empty (this number gives us a 1-1 portion for total count of real vs fake pixels.) multiply real images by masks.
    initialize $I_{mask_{style}} = I_{mask}$
    for Each mask instance $I_{mask_i}$ in $I_{mask}$ do
        if sum of all pixels == 0 then
            $I_{mask_{style}} = 1$
        end if
    end for
    $I_{RGB_{style}} \leftarrow I_{mask_{style}} * I_{RGB}$
    $\mu, \sigma^2 = E(I_{mask_{style}}, I_{RGB_{style}}, I_{Label})$
    $z \leftarrow U(\mu, \sigma^2) where U$ is the disentanglement FCN.
    $X \leftarrow$ fill function that fills $I_{Label}$ with corresponding $z$.
    $W \leftarrow X(I_{Label}, z)$
    $O_G \leftarrow G(I_{mask}, I_{RGB} * I_{mask}, I_{Label}, z)$ generator output.
    $O_{D_G} \leftarrow D(O_G)$ discriminator output on generated images.
    $O_{D_R} \leftarrow D(I_{RGB})$ discriminator output on real images.
    back-propagate through $E, G and D$ after computing losses.
end for
```

4.4.1 Datasets

We chose a variety of challenging segmentation datasets to test our work.

- **ADE20K [202]** consists of more than 20,000 training samples and 2,000 validation samples with 150 classes.
- **ADE20KBedroom** A subset of ADE20k with 2,000 training samples and 49 classes.
- **CelebAMaskHQ [203]** Is a dataset of segmented celebrity faces with 19 segmentation classes. This dataset has 20,000 training samples and 1,000 validation samples.

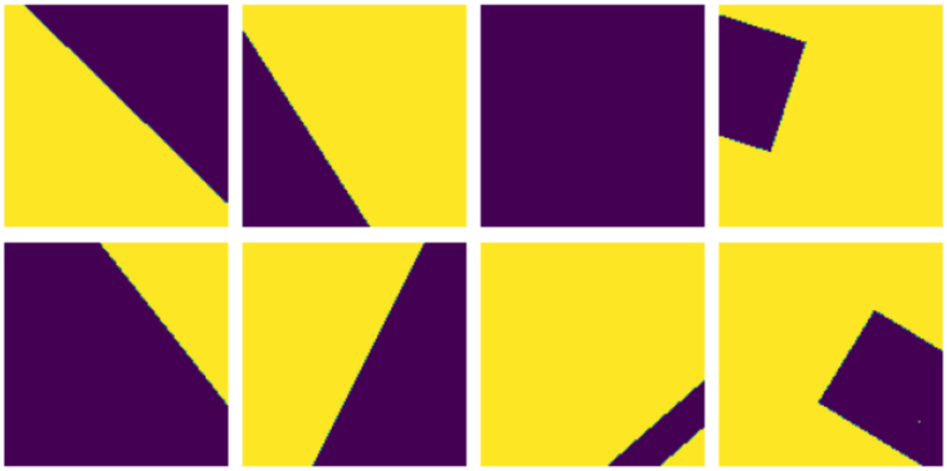


Figure 4.9: Examples of our randomly generated masks. Purple regions representing masked areas and yellow regions are unmasked areas.

4.4.2 Implementation

We use learning rate of 0.0001 for the generator and 0.0004 for the discriminator similar to [201]. We use weight of $\lambda = 10.0$ for the reconstruction loss. $\lambda = 1.0$ for adversarial loss and $\lambda = 0.05$ for KL divergence. We also experimented with multiple other losses including encoder feature-matching loss between real and fake samples, discriminator Wasserstein, and hinge adversarial loss.

We use L2 loss as our reconstruction criterion. we experimented with L2 loss that is class-balanced and class-balancing gives better eye reconstruction while regular L2 produces better background reconstruction. This would have less importance in datasets with less area discrepancy.

In order to get optimal results we train the system for around 5 Million samples.

4.4.3 Results

In this section, We demonstrate CASEIn's ability to perform inpainting, image reconstruction, and style transfer throughout various datasets by providing qualitative results.

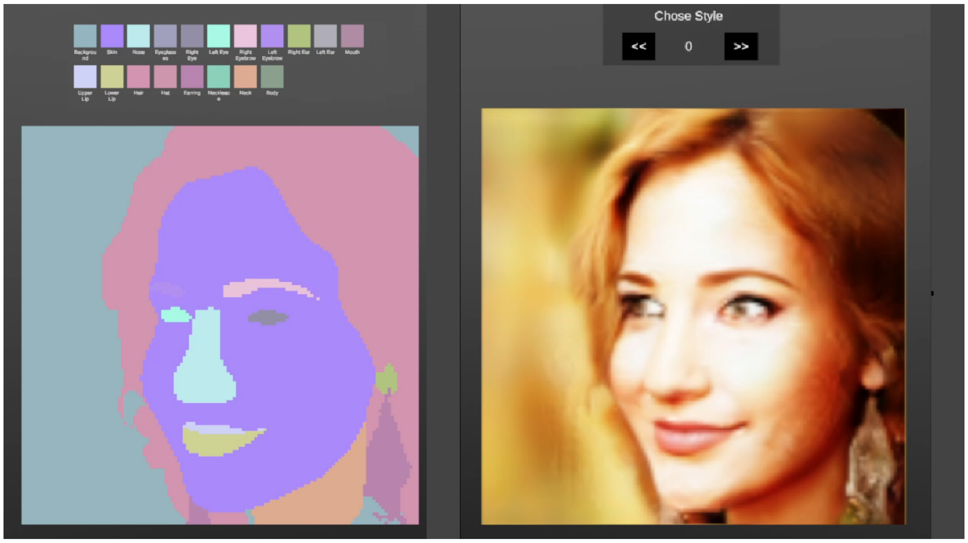


Figure 4.10: Our user interface, This interface lets user to paint and edit the labels and fill them with style of choice.

We show that the model is very effective in extracting existing RGB pixel regardless of their size. We also demonstrate the ability of the model to semantically synthesize images that lie outside of the training distribution 4.11.

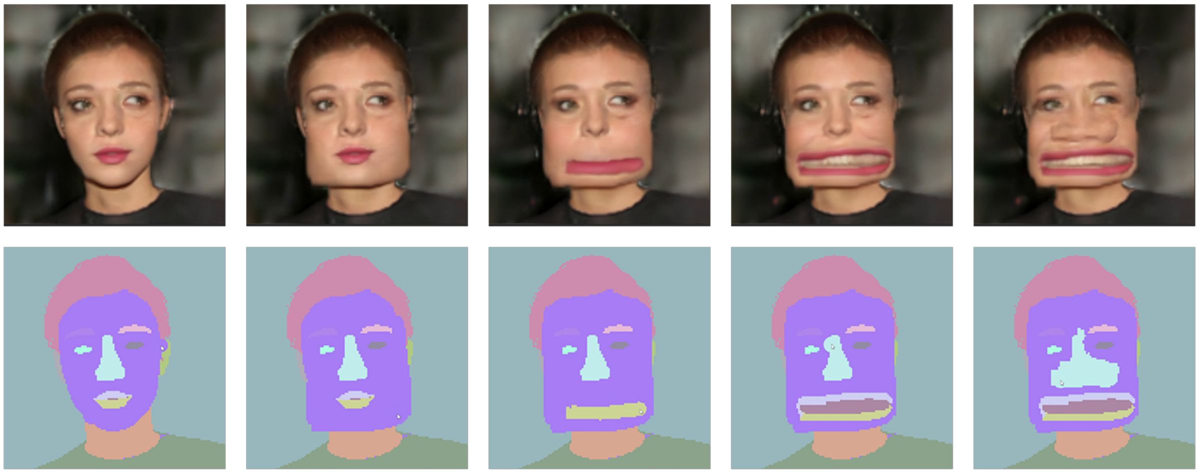
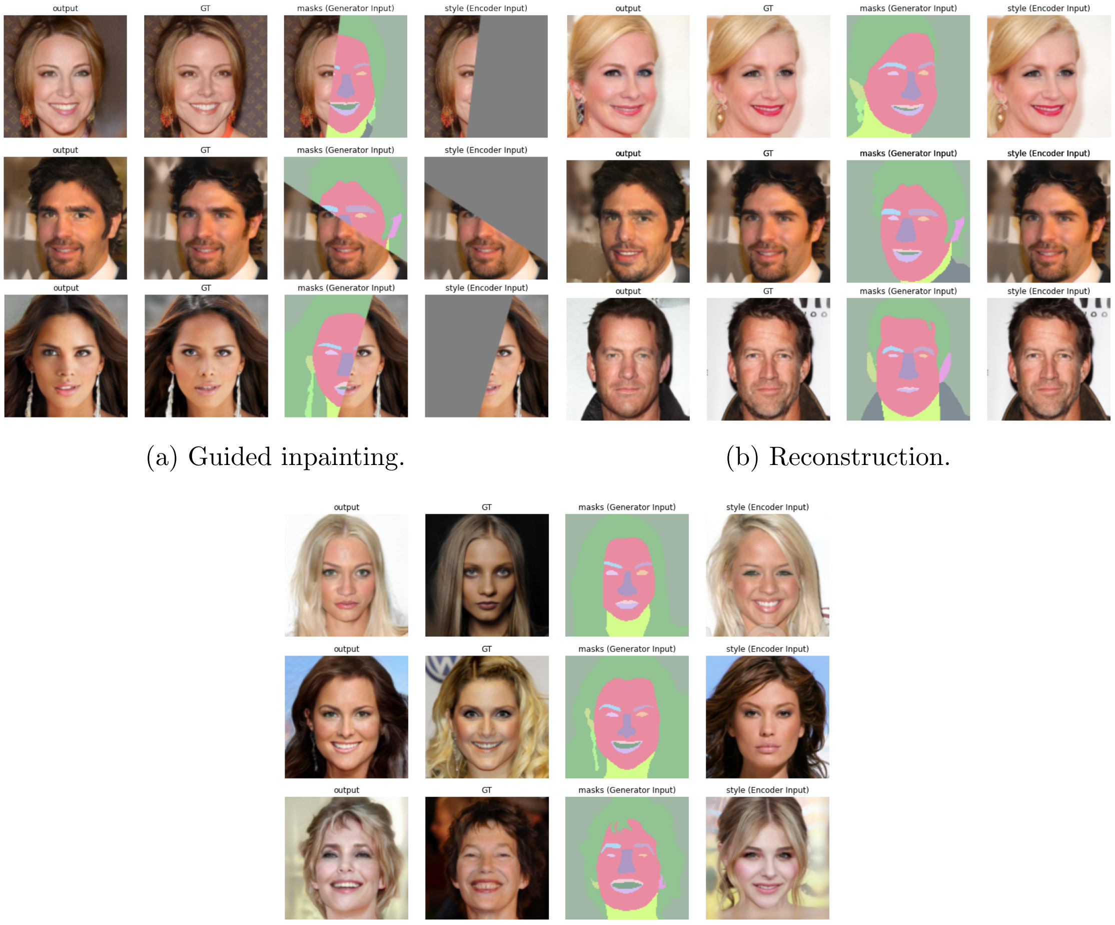


Figure 4.11: Progress of deforming a face using the CASEIn interface.



(a) Guided inpainting.

(b) Reconstruction.

(c) Style transfer.

Figure 4.12: Results for inpainting, style-transfer and image-to-image translation, trained on CelebAMask dataset.

4.5 Conclusion

In this work, we presented CASEIn, a convolutional generative adversarial network for content-aware semantic editing and inpainting. Image inpainting, completion, and extrapolation are essential techniques in computer vision and graphics with numerous real-world applications. With the help of these methods, images can be enhanced for various purposes, such as improving their visual appeal, making them more useful for tasks like changing the aspect ratio, or enabling novel image manipulation.

One of the key challenges in the development of inpainting and extrapolation techniques is the ability to handle large missing regions, maintain fine details and textures, and address complex image structures. CASEIn addresses these issues by utilizing a global style feature palette extracted by the encoder network to generate locally and globally coherent RGB based on existing RGB and semantic labels.

In this work, we made use of partial convolutions in the encoder network of CASEIn to enable the model to generate inpaintings that are coherent with the existing RGB values of the input image, regardless of their initial portion. By using partial convolutions, the encoder network can effectively ignore the masked-out pixels and focus on the available pixels when extracting features from the input image. This helps to ensure that the generated inpaintings are consistent with the existing content of the input image and do not accidentally ignore information if the existing region is small. The use of partial convolutions in CASEIn therefore plays a crucial role in enabling the model to produce high-quality, realistic inpaintings that are coherent with the existing content of the input image regardless of the mask size.

In our experiments, CASEIn demonstrated superior robustness and expressiveness in handling extreme inpainting problems compared to previous work. By utilizing partial convolutions, the CASEIn encoder is invariant to existing pixels, resulting in generated inpaintings that are coherent with the existing RGB values regardless of their initial portion. We also showed that by defining semantic image synthesis as a subproblem of guided inpainting, we can train a network that excels at both tasks simultaneously.

Finally, we provided a user interface that enables real-time inpainting and style transfer, showcasing the potential practical applications of CASEIn. Overall, our work makes significant contributions to the field of image inpainting, completion, and extrapolation and has the potential to impact a variety of industries, including computer graphics, image processing, and multimedia.

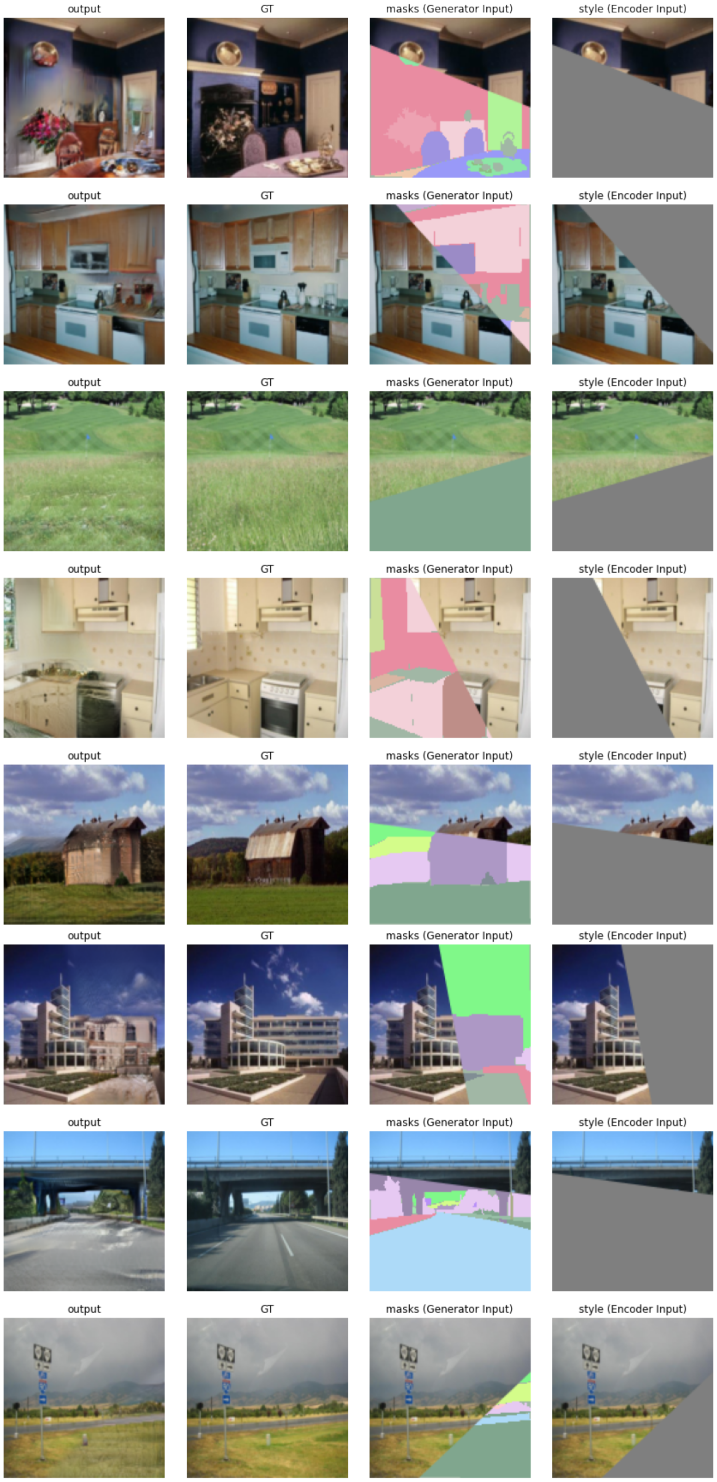


Figure 4.13: Results for inpainting, trained on the full ADE20K dataset.

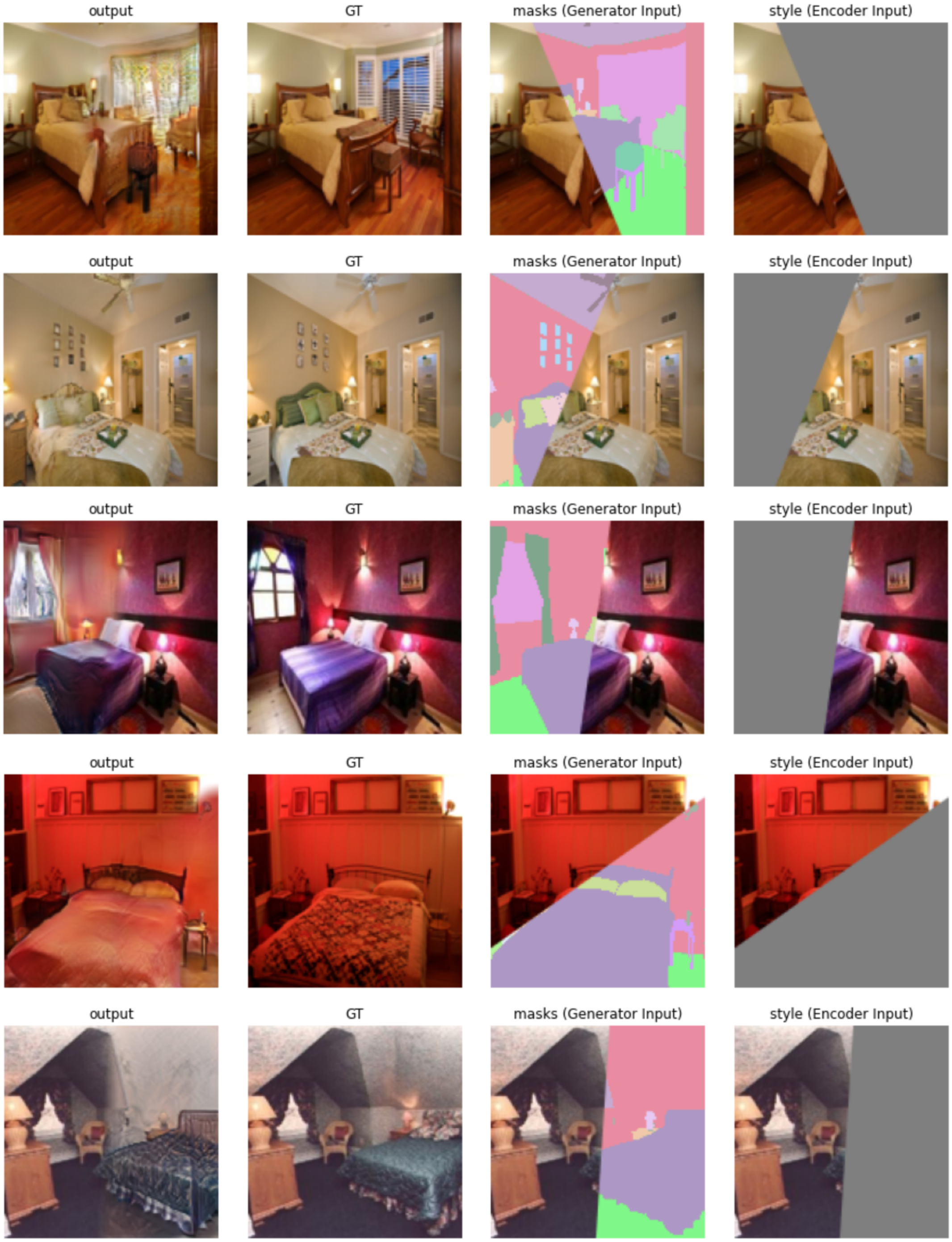


Figure 4.14: Results for inpainting, trained on ADE20K bedroom subset dataset.

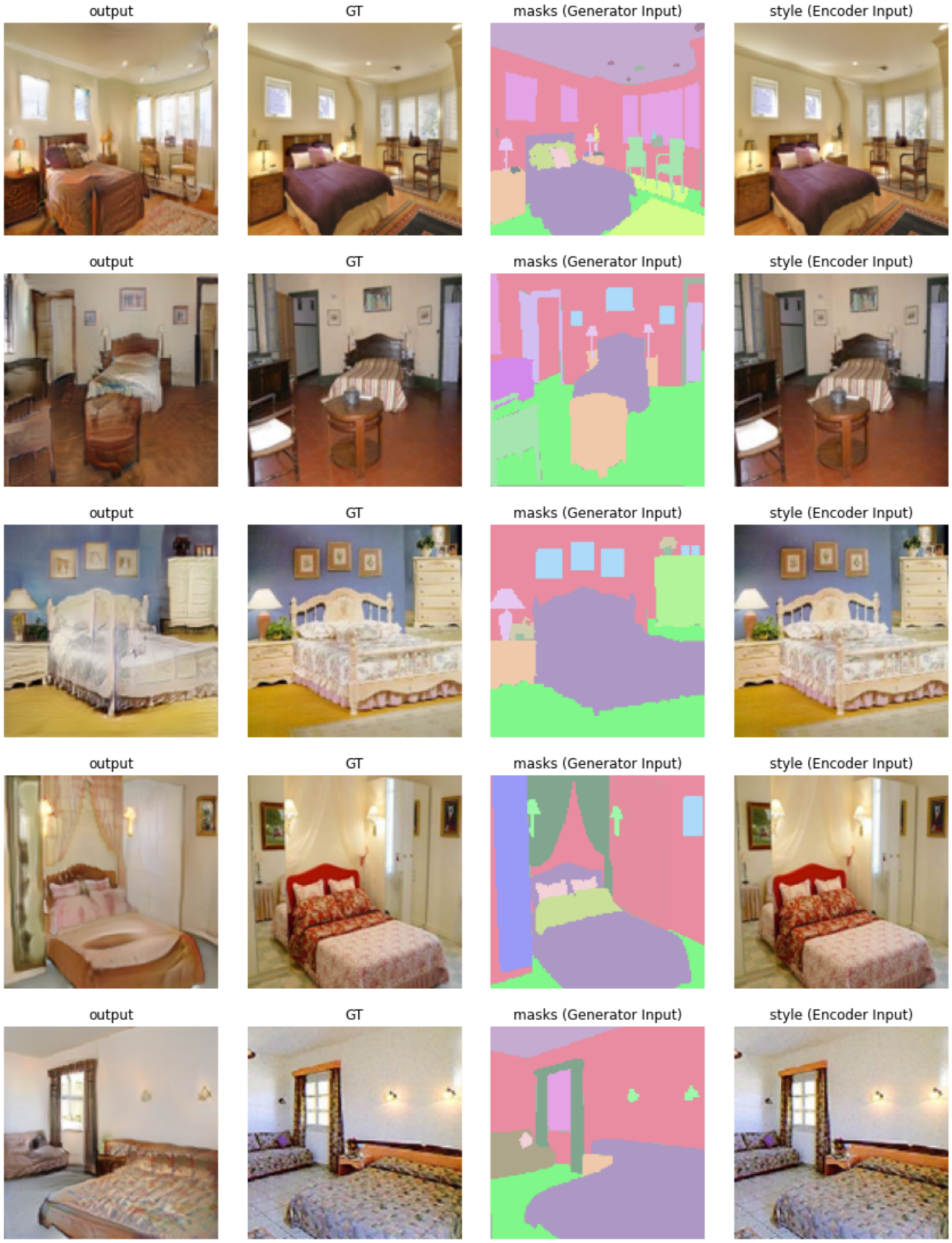


Figure 4.15: Results for reconstruction, trained on ADE20K bedroom subset dataset.

Chapter 5

Faded: A VR Application for Memory Reconstruction

5.1 Introduction

Memory is an intricate, multidimensional process and it is crucial to the production of art and content. It serves as the basis for each of our unique experiences, sense of self, and capacity to understand and engage with the outside world. Memory is a tool for recording and preserving our experiences as well as a source of inspiration for creating art.

Most of our memories only exist in our minds or are only partially supported by visual data, such as pictures and videos. For many of us, revisiting the past through various media is an important ritual because these memories shape who we are as people. This also explains our efforts to create artistic representations of our memories.

There is value in creating a true-to-form recreation of memories, even though the artistic impression of a memory can be a potent and evocative way to recreate an experience. Reconstructions of personal experiences that are accurate and true-to-form can be used to preserve historical events, give people a sense of continuity and belonging, and facilitate comprehension of complex concepts.

Memory has long been a topic of study in psychology and neuroscience [204, 205, 206, 207, 208, 209]. Understanding the mechanisms and processes by which we encode, store, and retrieve memories has significant implications for a number of industries, including technology, design, education, and healthcare.

Given the significance of memory and its role in our personal and collective histories, there is a clear need for tools and techniques to help us access and recreate our memories with accuracy and detail.

It may be challenging to reconstruct a location utilizing sparse data sources, such as a few pictures or just the user's memories. Typically, more information is required to depict the complexity of the real world. However, combining a powerful machine-learning model and an approachable and intuitive interface may boost the system's effectiveness by allowing users to communicate their understanding of the space in combination with their existing visual footage.

Generative machine learning models are well-suited to accompany such a memory reconstruction pipeline for several reasons. These models can learn the statistical patterns and relationships present in a dataset and use this knowledge to generate new, synthetic samples similar to the training data. This makes them particularly useful for tasks such as image generation, where the goal is to produce novel images that are consistent with a given set of constraints or characteristics.

In the context of memory reconstruction, generative models can be used to fill in missing or uncertain information in a user's memory. For example, suppose a user remembers an event but does not have a complete set of images or details. In that case, a generative model could be used to synthesize additional visual information based on the user's descriptions and the patterns present in a training dataset. This could improve the accuracy and detail of the reconstructed memory significantly but also carries the risk of manipulating the memory in the process.

An efficient and intuitive user interface is critical for processing memory reconstruction using machine learning. A well-designed interface allows users to easily input and communicate their memories and visual information. It provides clear and concise feedback on the status and progress of the reconstruction process. By considering the needs and goals of the user, it is possible to create an interface that is both practical and enjoyable to use.

Virtual reality is an excellent choice for this type of interface due to its ability to immerse the user in a realistic and interactive environment. VR can create a sense of presence and spatial awareness that can greatly enhance the user's ability to input and visualize their memories. VR can also provide various interactive and expressive tools and controls that allow users to manipulate and explore their reconstructed memories naturally and intuitively.

In this chapter, we introduce Faded, an interactive authoring tool to create 3D experiences from sparse images. The goal of this system is to rely on the user's understanding of memory to create a layout of the space and place the existing data, like images, into the scene and later on to use machine learning tools to extrapolate the missing parts of the environment based on the existing data. Faded is designed to be mainly used in VR but can also switch to the desktop mode for better compatibility.

5.2 System Architecture

Faded's system consists of multiple components. A floor designer, an object editor, a projection system, segmentation-based bundle adjustment and ML subsystems and networking, and a serializer. Each component has been developed with simplicity in

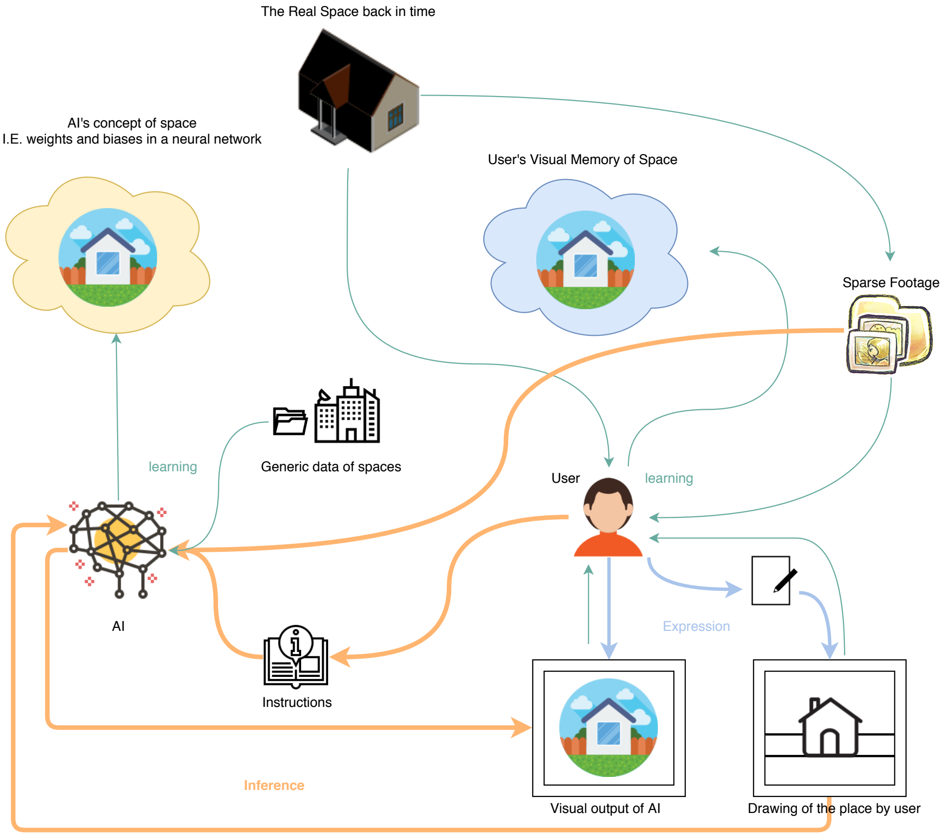


Figure 5.1: Information flow in a user-centered memory reconstruction system. Note that the user's expressive actions would encourage learning thus creating or reinforcing false memories.

mind. letting the users achieve results without complicated interactions.

5.2.1 Floor designer

Floor Designer lets users create a map design with straightforward interactions that work both in 2D and VR. The interface starts from a single polygon that users can modify by dragging the corners. While the user modifies the 2D version, a 3D version of the room updates in real-time on both desktop and VR modes.

The user can add vertices by pointing anywhere on the edge of the polygon and dragging. An existing vertex can be deleted by moving the pointer on the vertex and

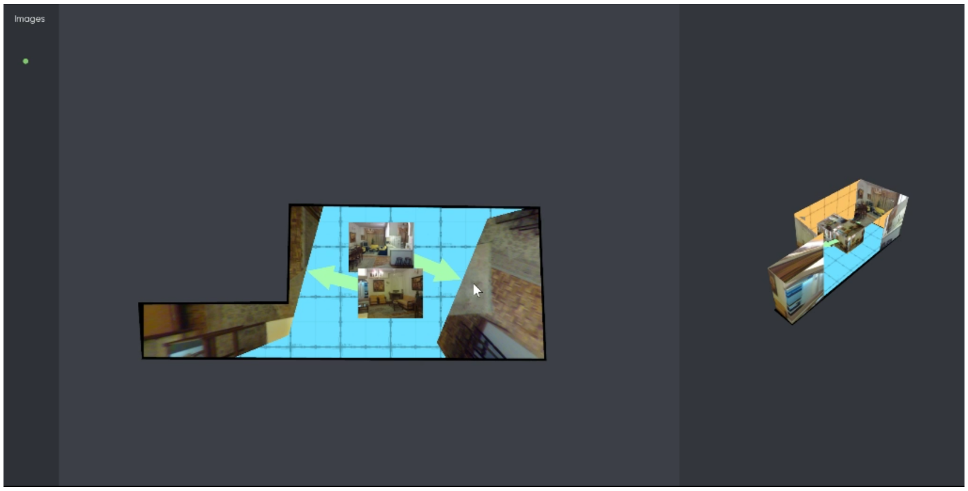
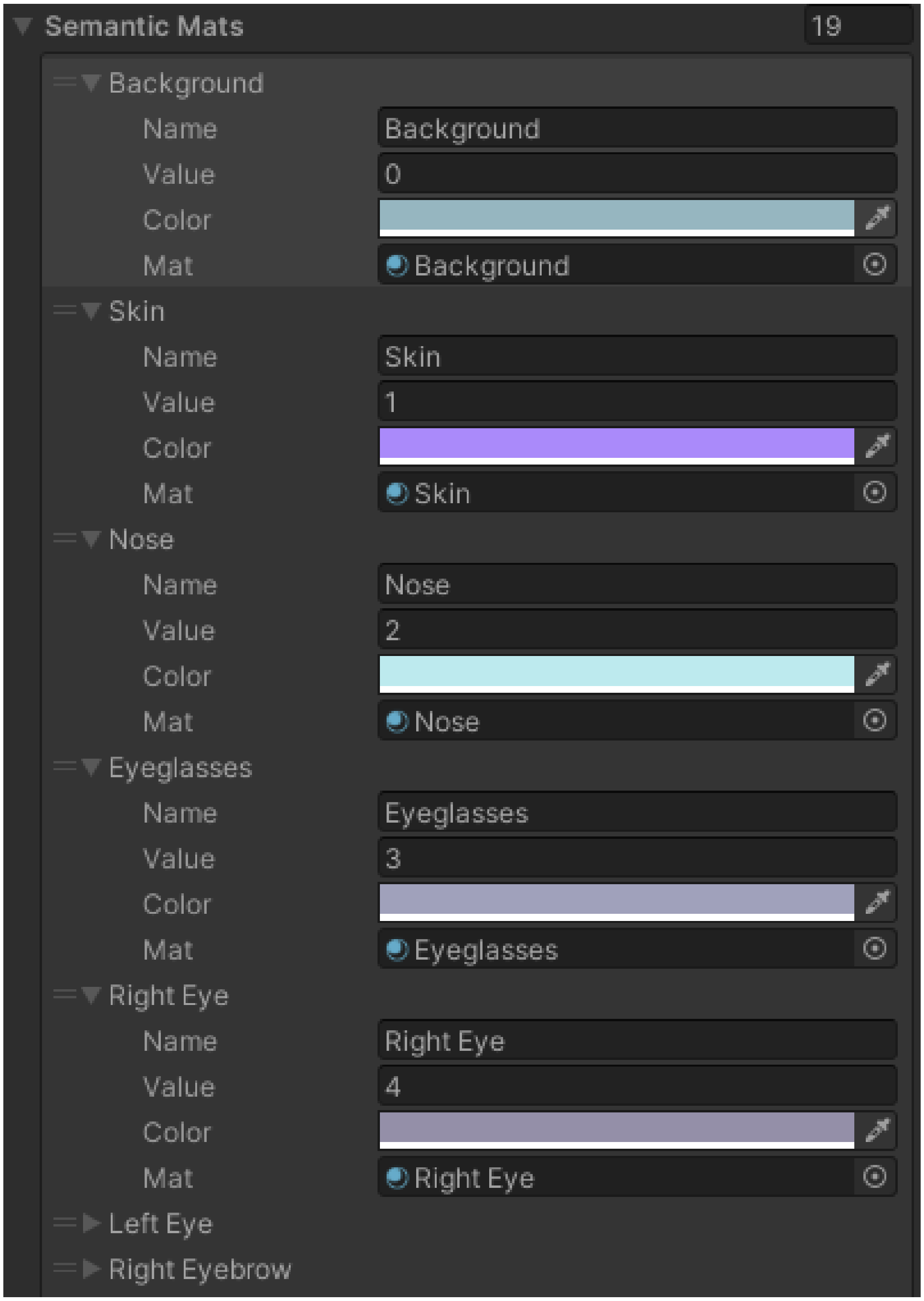


Figure 5.2: A floor map made with our floor designer with user pictures placed inside.

using the 2nd button of the controller. The user can adjust the height by pressing the height button and dragging the controller. Using this interface, users are capable of creating shapes that would encapsulate the indoor environment they are recreating.

5.2.2 Object editor

All objects in the program are labeled based on ADE20K segmentation labels, including the walls, floor, and ceiling created with the Floor Designer tool. Users might remember specific objects being present in an environment. The object editor tool presents a menu that lets users search through Shapenet 3D models and instantiate them in the scene. Each imported object has the corresponding label from ADE20K that is used in semantic segmentation. Users can drag and drop the objects from the object menu and, later, scale and rotate them in the scene using direct manipulation. In order to represent the ADE20K classes, we generate a set of color-coded materials that are randomly generated based on the ADE20K class label as the seed. This way, each object in the scene receives a distinct color that separates it from the surroundings while representing a corresponding class in ADE20K dataset.



Semantic Mats 19

Background

Name: Background
Value: 0
Color: ■
Mat: ● Background

Skin

Name: Skin
Value: 1
Color: ■
Mat: ● Skin

Nose

Name: Nose
Value: 2
Color: ■
Mat: ● Nose

Eyeglasses

Name: Eyeglasses
Value: 3
Color: ■
Mat: ● Eyeglasses

Right Eye

Name: Right Eye
Value: 4
Color: ■
Mat: ● Right Eye

Left Eye

Right Eyebrow

Figure 5.3: List of randomly generated materials based on CelebAMask dataset.

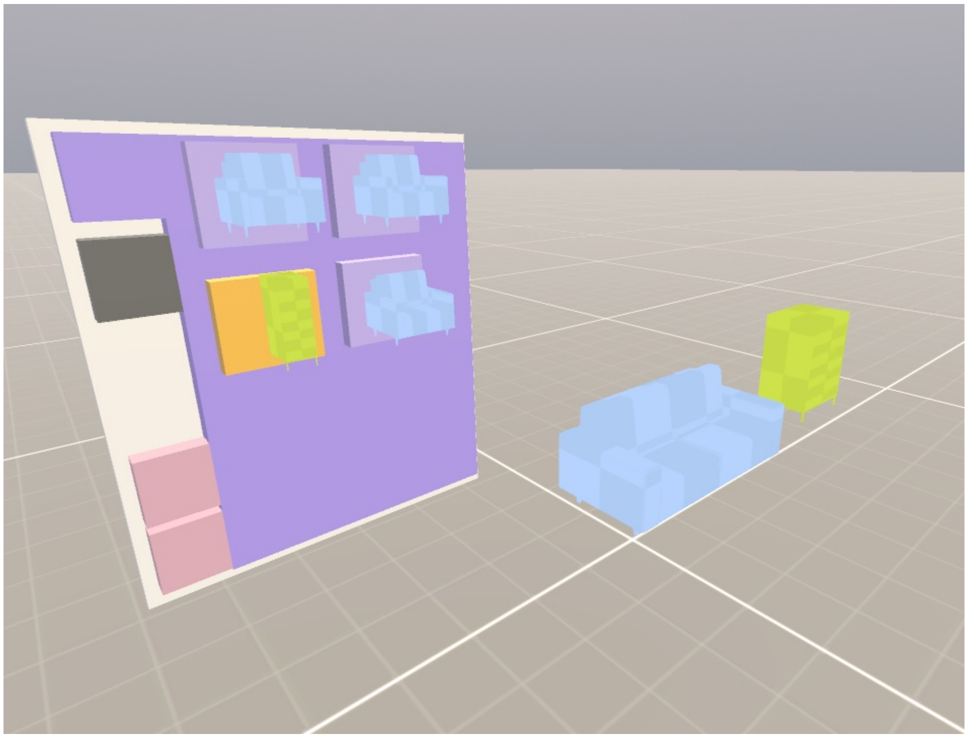


Figure 5.4: 3D drag and drop menu for objects. Objects are replenished on the menu after being dropped in the 3D scene.

5.2.3 Projection system

Faded focuses on bridging the capabilities of 2D ml models and 3D user experiences. For this purpose, we chose projection texture mapping as the method of choice for communication between our 3D and 2D interfaces. When the layout is assembled in the object editor, we render a set of 2D buffers in unity and send them to our ML server. The 2D buffers include a 2D Segmentation map, 2D RGB, and mask. The existing RGB values are provided by projecting a 2D image into the 3D scene using projection texture mapping.

The user roughly places the image projector by selecting them through the menu and positioning them in the space. The image is projected in the 3D environment. The user then can use a manual or automatic sweeping that moves a camera and generates the buffers while sending them to the ML network. The received RGB image is then projected back into the environment.

In the automatic sweeping, the user can adjust the frequency and coverage of each projection in the next camera instance. We found that a 50 percent coverage usually produces the highest quality output.

5.2.4 Segmentation-based bundle adjustment

Guessing the field of view of an existing image is a complicated task. In our case, the user has already placed the 3D layout and the rough initial pose of the image using the floor designer and object editing tools. Therefore, We can optimize the field of view and the correct position of the image using the available data. We do so by running a segmentation algorithm on the image and solving for the position and field of view that minimizes a loss between the 2D image and the rasterized segmentation mask from the 3D environment. In order to run the optimization in our real-time environment, we use a gradient-free optimization technique called particle swarm optimization.

Each particle is essentially an instance of the solver randomly swarming the parameter space to find the minimal loss. In this case, the loss function is defined as Particles communicating with each other using the particle swarm formula.

Particle swarm optimization (PSO) is a computational method that uses the collective intelligence of a swarm of simple agents to find solutions to optimization problems. It is inspired by the social behavior of birds in a flock and the swarming behavior of insects.

In PSO, a population of particles is initialized with random positions and velocities in the search space. Each particle represents a potential solution to the optimization problem. The particles move through the search space according to their own velocity and the influence of the best position found by the particle itself and its neighbors. The velocity of each particle at time step t, v_t, is updated using the following equation:

$$v_t = w * v_{t-1} + c_1 * r_1 * (pbest_t - x_t) + c_2 * r_2 * (gbest_t - x_t) \tag{5.1}$$

where w is the inertia weight, c_1 and c_2 are acceleration constants, r_1 and r_2 are random numbers between 0 and 1, $pbest_t$ is the best position found so far by the particle itself, $gbest_t$ is the best position found so far by the whole swarm, and x_t is the current position of the particle.

The position of the particle at time step t, x_t, is then updated using the following equation:

$$x_t = x_{t-1} + v_t \tag{5.2}$$

The process is repeated until a satisfactory solution is found or a pre-determined number of iterations has been reached. PSO has been applied to various optimization problems in various fields, including engineering, economics, and computer science.

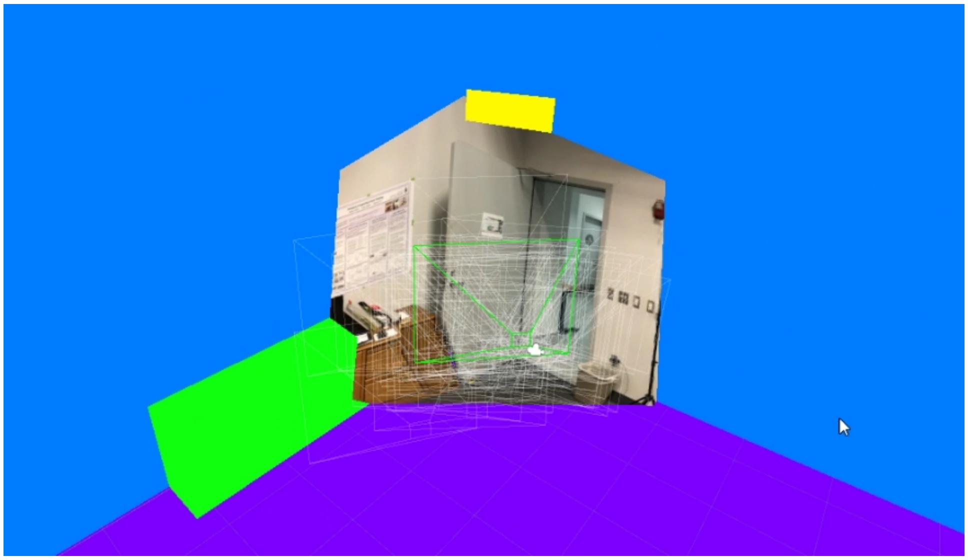


Figure 5.5: Bundle adjustment using particle swarm optimization. the image is segmented using an ML model and is localized to match the 3D layout of the room.

We use multi-class cross-entropy loss 5.3 as our loss function where y is the binary indicator (0 or 1) if class label c is the correct classification for observation o and p is predicted probability observation o is of class c.

$$\mathcal{L} = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{5.3}$$

In cases where the image does not have enough matching segmentation features with the layout, like when the image includes many people or when the object in the image has moved, the user can manually match the 2D features of the image with their 3D correspondence. This manual feature matching usually results in better quality bundle adjustment.

5.2.5 ML models and networking

For 2D inpainting and extrapolation, we use CASEIn (Content aware semantic editing and inpainting). In combination with a U-Net++ segmentation model, both trained on

ADE20K. For each inpainting pass, an RGB, segmentation, and a mask buffer are passed to the inpainting network, CASEIn.

We use Flask to create an ML server that receives and sends web requests via HTTP POST. The buffers are encoded to PNG and sent as a byte array to and from the server. Initially, all existing RGB pixels are sent to U-Net++, and each pixel is labeled with a segmentation mask. This segmentation label is then combined with the segmentation buffer directly rendered from our 3D space. The new combined segmentation mask alongside mask and RGB are then sent to the inpainting network, CASEIn.

CASEIn network consists of an encoder and a generator. The encoder receives the mask, RGB, and corresponding segmentation and produces a style palette that describes the style of every single class in the image. The style network is then used with the RGB and mask images to create a complete square image. The inpainted image is then sent back to the client and reprojected as a new RGB image into the scene.

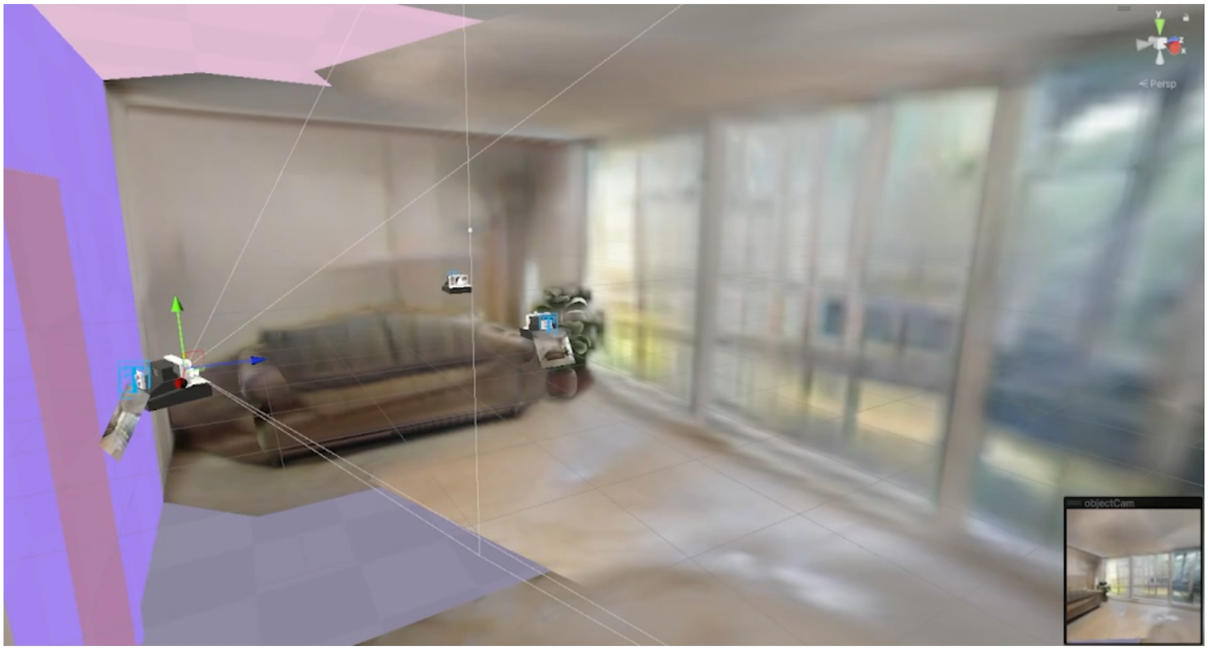


Figure 5.6: Remote inpainting ML model in action, with each iteration the inpainted image projected into the semantic room.

5.2.6 Serialization

The serialization component combines all 3D objects in the scene with the projected textures into a usable asset. Initially, all triangular meshes are combined into a single Unity Mesh object. The object is then UV-Mapped using unity automatic UV unwrapping in Unity. We use a rendering shader to unproject the pixels into the UV space and save that RenderTexture as a 2D PNG file. The Unity mesh is also saved using an OBJ converter plugin. the combination lets the user retrieve the model without the need to have access to the projective shader, which enables use cases in various 3D applications.

5.3 Conclusion

In this chapter, we discussed the importance of memory and its role in creating and preserving personal experiences and historical events. We proposed using machine learning and a user-friendly interface, such as virtual reality, to aid in the reconstruction of memories with accuracy and detail. We suggested that generative machine learning models can be used to fill in missing or uncertain information in a user's memory and that VR can provide a realistic and immersive environment for inputting and visualizing memories.

Through our research, we have demonstrated the potential for using these approaches to enhance the accuracy and detail of memory reconstruction. Our results show that combining a powerful machine learning model with an intuitive interface can significantly improve the effectiveness of memory reconstruction. We presented Faded, an interactive authoring tool to create 3D experiences from sparse images in virtual reality. This system relies on the user's memory of space in combination with existing footage of space. It enables the user to recreate an environment by creating a layout from memory and placing the pictures into the space. it then uses a machine learning model to complete the missing areas in the scene.

Overall, our work highlights the importance of developing tools and techniques to help people access and recreate their memories with more accuracy and detail. We believe that this research has the potential to have significant impacts in a variety of fields, including psychology, education, therapy, and creative domains.

Chapter 6

Conclusion

With the increasing need for 3D content resulting from the rise of the popularity of AR/VR, we need to explore new and more accessible ways to create 3D content. The research reported in this dissertation addresses the importance of designing these practical tools for 3D content generation. Throughout this dissertation, we study various forms of interaction with 3D content and explore the areas that affect the immersiveness of these experiences. We demonstrate several ways to increase the intuitiveness of 3D content generation tools by designing natural and easy-to-use user interfaces and incorporating artificial intelligence to do the heavy lifting for complicated tasks.

This dissertation explores various methods for designing and evaluating 3D user interfaces for creating and interacting with 3D content in augmented reality (AR) and virtual reality (VR). We have investigated a variety of issues regarding 3D user experiences and the usability of content creation in VR and AR through the projects PanoTrace, VR Walking and Teleportation, DeepDive, CASEIn, and Faded.

By examining the effects of various viewing degrees of freedom (3DoF vs. 6DoF) and stereoscopic depth on immersion and simulator sickness, as well as the effects of various modes of locomotion and lighting conditions on task performance and comprehension of

3D space, our research has advanced our understanding of 3D user experiences in virtual and augmented reality. In terms of content creation, our projects have investigated the use of intelligent tools along with approachable user interfaces to enable users to quickly produce and modify 3D content. In Figure 6.1 we show the effect of the work in this dissertation on pushing visual media forward in domains of creation accessibility, experience accessibility, and immersiveness. While this chart visualizes a qualitative effect on these media, it's based on the quantitative results demonstrated throughout this dissertation. In section 6.1 we argue the relationship between these impacted media and our individual projects in more detail.

While our work has clarified a number of significant aspects of 3D user interfaces, there are many other research avenues that are yet to be explored. Effective use of AI in content generation especially for AR and VR would bring up many interesting research topics. Overall, there is a need for more research and development in this field, and the projects discussed in this dissertation lay a foundation for future work in this fascinating and quickly developing field.

6.1 Contributions

The work reported in this dissertation has implications for both content creators and consumers. By providing automated pipelines, and approachable user interfaces for 3D content generation, a non-expert user can also participate in creating 3D content. We discussed the importance of personalized 3D content with the increased popularity of native 3D user experiences like augmented and virtual reality.

Our works, Faded and Panotrace, focus on moving from existing 2D data into a more immersive 3D form. By enabling users to add extra depth values and extrapolate an existing 2D image in a 3D environment, we positively move 2D panoramas and digital

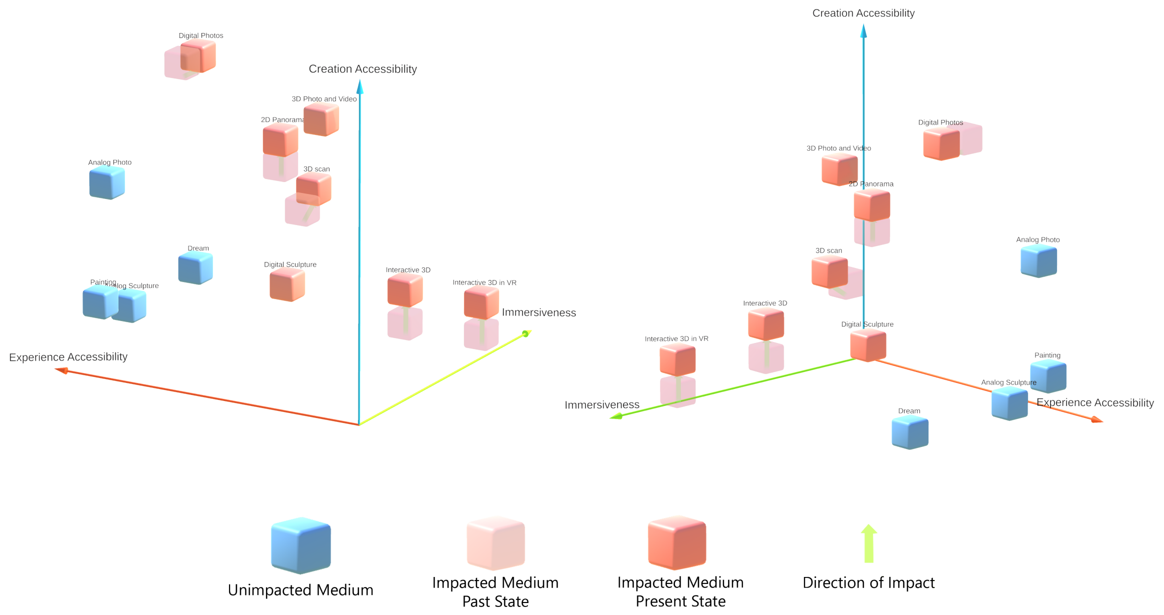


Figure 6.1: Perceptual map for how various visual media is affected by this thesis in the domain of creation accessibility, experience accessibility, and immersiveness. Note that the arrows are used to show the direction in which each medium is moved, and are not meant to be a quantitative measure of impact.

photos in the domain of creation accessibility and immersiveness.

Extrapolation of 3D scans and 2D images into a unified environment in our work Faded improves the creation accessibility of 3D scans. In our Walking and Teleportation in wide-area VR study, we identify the benefits of various methods of locomotion while exploring 3D environments, which helps improve the immersiveness of 3D scans.

Both PanoTrace and Faded also contribute to the creation accessibility of interactive 3D content by providing innovative 3D user interfaces for 3D modeling.

6.2 Generative AI: a Double-Edged Sword?

Generative AI technologies have received a lot of interest in recent years because of their capacity to generate very realistic visuals and art based on input prompts or examples. These technologies can potentially transform a wide range of industries and services, including advertising, product design, film, and television. However, there are also possible drawbacks to consider, such as the risk of these technologies being used to create fraudulent or misleading content, the possibility of job loss or disruption in the art and design industries, and the ethical implications of these events.

It is critical to consider the potential advantages and disadvantages of generative AI and the risks and concerns that these technologies may present. By being aware of both their potential benefits and limitations, we can appropriately assess the role of these technologies in today's environment and ensure that they are used responsibly and ethically.

We should first clarify what we mean by "visual generative AI technology." These algorithms or systems employ machine learning techniques to create unique images or art from input prompts or samples. These technologies can produce extremely realistic visuals that are frequently indistinguishable from those made by people. Generative AI has the potential to significantly speed up and streamline the process of creating such images and art. Our work CASEIn, which we demonstrated in Chapter 4 is an example of a generative AI model based on GAN architecture. There are however many other types of generative networks, each having advantages and disadvantages in creating content. Recently, diffusion models [21] have gained immense popularity due to their ability to create high-quality images.

One potential benefit of these systems is that they can save time and costs for businesses and organizations that rely on visual content for their operations. An advertising agency, for example, may employ generative AI to swiftly generate a huge number of pictures for a new campaign, reducing the time and effort required to make each image manually. Similarly, a corporation that needs a lot of original concepts for product ideation may be able to employ this technology to generate a large number of images quickly rather than commissioning individual work from artists.

In the case of 3D virtual environments, missing data is very common throughout the process of creating a space. We demonstrated an extreme case of that in our memory reconstruction work Faded in Chapter 5. We showed how generative models could become very helpful in the case of filling large missing areas. Even in the less extreme cases of missing data where the goal is to use reconstruction pipelines with lots of images, it's very common to end up with models with partially missing areas and holes. Generative models can be a fantastic tool to reconstruct these areas with plausible information and increase the quality of the final product.

Another potential advantage of generative AI is that it can be used to create images that humans would find difficult or impossible to create. for example, They can produce highly detailed images or incorporate complex patterns or visual effects. This could be valuable in areas like architecture, where complex visualizations are frequently required to demonstrate design concepts, or in the film industry, where similar technologies could be utilized to create special effects that would be impossible to achieve with traditional techniques. Any area with a high skill ceiling for producing visual content will be more accessible through these technologies.

Overall, the capacity of generative AI to save time and money while producing extremely realistic or complicated images is expected to be a massive benefit to a wide range of professionals and organizations.

Despite having the potential to be very helpful in various scenarios, there are some possible drawbacks to consider regarding generative AI. One of the major concerns is the potential for these technologies to produce fake or deceptive images to trick or mislead an audience. The use of visual generative AI technologies to produce altered images or videos that appear to show people or events in a certain way but are not accurate depictions of reality could be one example of this. This could be especially worrying in the present political context, where fake news and misinformation are pervasive, as these manufactured photos could propagate inaccurate or misleading information or negatively affect public opinion.

The potential for these technologies to produce visuals meant to manipulate individuals by appealing to their emotions or biases is another possible problem. These technologies, for instance, may be used to produce visuals intended to appeal to people's feelings of nationalism or their fears of particular individuals or groups of people or events in order to affect their opinions or behavior.

The possibility for prejudice and bias to show in the pictures and artwork created by visual generative AI is a big concern. If the data used to train these models is biased or unrepresentative, the resulting pictures may reinforce or amplify existing prejudices or preconceptions. This may have detrimental consequences for individuals impacted by these prejudices, as it can marginalize or exclude certain groups. For instance, if a visual generative AI model is trained on data that primarily reflects a particular race or gender, it may generate visuals that reflect and magnify existing prejudices or assumptions about these groups. This may substantially affect social and economic equality, stressing the need to carefully examine the data and algorithms used to train these models. In Figure 6.2 we demonstrate an example of this bias in a generative model by prompting the model to generate a "Portrait of an American politician" without additional context. As you might see in the results, The model has a tendency towards a certain gender and racial group.

Overall, it's crucial to be aware that generative AI could be applied to produce fake or deceptive images and to take precautions to ensure that these technologies are used ethically and responsibly. This could entail forming guidelines or standards to control how these technologies are used, as well as instruments or techniques to identify and stop the usage of harmful content.

We must keep the possible benefits and risks in mind as we continue to investigate and

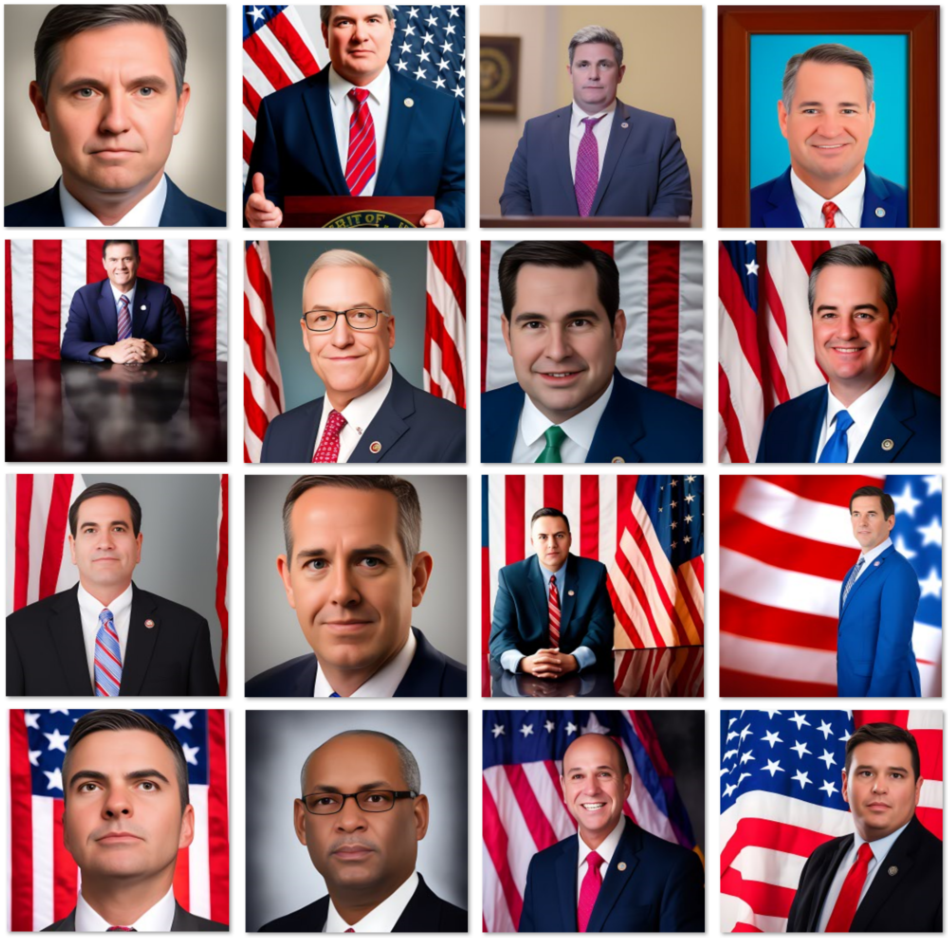


Figure 6.2: "Portrait of an American politician, 4K photo" 16 random samples generated using Stable Diffusion [25]. The images are not cherry-picked.

improve generative AI systems. While these technologies have the potential to provide significant benefits in a variety of areas, we must also be cognizant of the risk of abuse or misuse. This requires us to be vigilant in our attempts to prevent the fabrication and spread of false or misleading images, as well as in setting rules or legislation to control the use of these technologies. We must approach the development and deployment of these technologies with caution and attention. By considering both the potential downsides and upsides of these technologies, we can move forward in a methodical thoughtful way that minimizes the risk of harm.

6.3 Future Work

One area that should be further investigated is utilizing machine learning to generate content for VR and AR. Although the DeepDive and CASEIn projects show that using generative models to create 3D content is feasible, many unanswered questions remain regarding how to successfully incorporate such models into 3D user interfaces for creative applications. Future research might, for instance, look at how various machine learning algorithms (such as diffusion models and transformer-based encoders) can be used to produce a diverse range of 3D content or how different input modalities (such as gestures and voice commands) can be used to control the generation process. The use of text-to-image generative models, which could enable users to specify 3D content using natural language descriptions, is another promising area of research.

Future research can also examine further how various 3D user interfaces affect user performance and experience. Many other aspects could be investigated, such as the use of haptic feedback, the impact of visual complexity, or the effects of different display technologies. The projects PanoTrace and VR Walking and Teleportation offer insightful analyses into the role of 3DoF vs. 6DoF and locomotion techniques in 3D spaces. Conducting user studies to measure the effects of different interface design decisions on various metrics of interest may be a part of this type of research (e.g., task completion time, perceived immersion).

Another area that we touch on in this dissertation but has lots of potential for future research is the use of AI in 3D user interfaces. Numerous real-world applications, such as education, training, or entertainment, may benefit from intelligent 3D UI. Research might, for instance, look into how 3D user interfaces can be made to support a particular type of training or how they can be used to make immersive experiences more enjoyable. Overall, the field of Intelligent 3D UI has considerable potential for future research and development, and the projects presented in this dissertation lay a foundation for that.

There are several possible future research avenues regarding generative AI and 3D space. Future work can focus on providing the building blocks of a 3D experience. For example, diffusion models that instead of outputting a single image, are trained to generate multiple elements that construct a scene via a differentiable renderer. These building blocks can include meshes, volumes, textures or materials. Such networks can be integrated directly in 3D applications and enable everyday users to create sophisticated reusable 3D assets and scenes.

Bibliography

[1] A. Revonsuo, *Consciousness, dreams and virtual realities*, *Philosophical Psychology* **8** (1995), no. 1 35–58.

[2] K. Andersen, *The geometry of an art: the history of the mathematical theory of perspective from Alberti to Monge.* Springer Science & Business Media, 2008.

[3] "Meta spark studio - create immersive ar experiences." `https://sparkar.facebook.com/ar-studio/`. (Accessed on 01/06/2023).

[4] "Lens studio." `https://ar.snap.com/en-US/lens-studio`. (Accessed on 01/06/2023).

[5] "Tiktok effect house." `https://effecthouse.tiktok.com/`. (Accessed on 01/06/2023).

[6] "The death of photography: are camera phones destroying an artform? — photography — the guardian." `https://www.theguardian.com/artanddesign/2013/dec/13/death-of-photography-camera-phones`. (Accessed on 01/06/2023).

[7] "Dreams — media molecule." `https://www.mediamolecule.com/games/dreams`. (Accessed on 01/06/2023).

[8] "Gravity sketch — 3d design and modelling software." `https://www.gravitysketch.com/`. (Accessed on 01/06/2023).

[9] "Top 3d sculpting tools for virtual reality authoring — medium by adobe." `https://www.adobe.com/products/medium.html`. (Accessed on 01/06/2023).

[10] "3d modeling software - adobe substance 3d modeler." `https://www.adobe.com/products/substance3d-modeler.html`. (Accessed on 01/06/2023).

[11] "Instalod – everything you need for the production and automatic optimization of 3d content.." `https://instalod.com/`. (Accessed on 01/06/2023).

[12] "3dcoat 2022." `https://3dcoat.com/`. (Accessed on 01/06/2023).

[13] "Nanite virtualized geometry in unreal engine — unreal engine 5.0 documentation." `https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine/`. (Accessed: 10/20/2022).

[14] S. Gosh, *History of photogrammetry*, *Laval University, Canada* (1981).

[15] J. L. Schonberger and J. M. Frahm, *Structure-from-Motion Revisited*, 2016.

[16] S. M. Ali Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis, *Neural scene representation and rendering*, *Science* **360** (2018), no. 6394 1204–1210.

[17] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, *Digital twin in industry: State-of-the-art*, *IEEE Transactions on industrial informatics* **15** (2018), no. 4 2405–2415.

[18] A. Valli, *The design of natural interaction*, *Multimedia Tools and Applications* **38** (2008), no. 3 295–305.

[19] A. Brock, J. Donahue, and K. Simonyan, *Large scale gan training for high fidelity natural image synthesis*, *arXiv preprint arXiv:1809.11096* (2018).

[20] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, *Analyzing and improving the image quality of stylegan*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020.

[21] P. Dhariwal and A. Nichol, *Diffusion models beat gans on image synthesis*, *Advances in Neural Information Processing Systems* **34** (2021) 8780–8794.

[22] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, *Gans trained by a two time-scale update rule converge to a local nash equilibrium*, *Advances in neural information processing systems* **30** (2017).

[23] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, *Hierarchical text-conditional image generation with clip latents*, *arXiv preprint arXiv:2204.06125* (2022).

[24] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, *et. al.*, *Photorealistic text-to-image diffusion models with deep language understanding*, *arXiv preprint arXiv:2205.11487* (2022).

[25] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, *High-resolution image synthesis with latent diffusion models*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

[26] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, *et. al.*, *Make-a-video: Text-to-video generation without text-video data*, *arXiv preprint arXiv:2209.14792* (2022).

[27] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, *Dreamfusion: Text-to-3d using 2d diffusion*, *arXiv preprint arXiv:2209.14988* (2022).

[28] Y.-J. Kim, R. Kumaran, E. Sayyad, A. Milner, T. Bullock, B. Giesbrecht, and T. Höllerer, *Investigating search among physical and virtual objects under different lighting conditions*, *IEEE Transactions on Visualization and Computer Graphics* **28** (2022), no. 11 3788–3798.

[29] E. Sayyad, P. Sen, and T. Höllerer, *Panotrace: interactive 3d modeling of surround-view panoramic images in virtual reality*, in *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, pp. 1–10, 2017.

[30] R. Szeliski and H.-Y. Shum, *Creating full view panoramic image mosaics and environment maps*, in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 251–258, ACM Press/Addison-Wesley Publishing Co., 1997.

[31] S. DiVerdi, J. Wither, and T. Höllerer, *Envisor: Online environment map construction for mixed reality*, in *Virtual Reality Conference, 2008. VR'08. IEEE*, pp. 19–26, IEEE, 2008.

[32] Y. Xiong and K. Pulli, *Fast panorama stitching for high-quality panoramic images on mobile phones*, *IEEE Transactions on Consumer Electronics* **56** (2010), no. 2.

[33] C. Weissig, O. Schreer, P. Eisert, and P. Kauff, *The Ultimate Immersive Experience: Panoramic 3D Video Acquisition*, pp. 671–681. Springer, Berlin, Heidelberg, 2012.

[34] L. E. Gurrieri and E. Dubois, *Stereoscopic cameras for the real-time acquisition of panoramic 3D images and videos*, p. 86481W, International Society for Optics and Photonics, mar, 2013.

[35] D. Gledhill, G. Y. Tian, D. Taylor, and D. Clarke, *Panoramic imaging–a review*, *Computers & Graphics* **27** (jun, 2003) 435–445.

[36] N. Barber, "Start Caring About VR And 360-Degree Video." `https://www.forbes.com/sites/forrester/2017/01/10/start-caring-about-vr-and-360-degree-video/`, 2017. Accessed: July 1, 2017.

[37] "360 virtual reality camera." `https://ozo.nokia.com/`, 2017. Accessed: 2017-07-01.

[38] S. E. Chen, *Quicktime vr: An image-based approach to virtual environment navigation*, in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, (New York, NY, USA), pp. 29–38, ACM, 1995.

[39] "Virtual Reality Videos." `https://support.google.com/youtube/answer/6316263?hl=en`, 2017. Accessed: 2017-07-01.

[40] "Cardboard Camera VR Photo Format." `https://developers.google.com/vr/concepts/cardboard-camera-vr-photo-format`, 2017. Accessed: 2017-07-01.

[41] "360 Video." `https://facebook360.fb.com/`, 2017. Accessed: 2017-07-01.

[42] P. E. Debevec, C. J. Taylor, and J. Malik, *Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach*, in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 11–20, ACM, 1996.

[43] M. Levoy and P. Hanrahan, *Light field rendering*, in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, (New York, New York, USA), pp. 31–42, ACM Press, 1996.

[44] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, *The lumigraph*, in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, (New York, New York, USA), pp. 43–54, ACM Press, 1996.

[45] B. M. Oh, M. Chen, J. Dorsey, and F. Durand, *Image-based modeling and photo editing*, in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 433–442, ACM, 2001.

[46] F. Huang and R. Klette, *Stereo panorama acquisition and automatic image disparity adjustment for stereoscopic visualization*, *Multimedia Tools and Applications* **47** (2010), no. 3 353–377.

[47] S. Peleg, M. Ben-Ezra, and Y. Pritch, *Omnistereo: Panoramic stereo imaging*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23** (2001), no. 3 279–290.

[48] K. Kwiatek and R. Tokarczyk, *Immersive photogrammetry in 3d modelling*, *Geomatics and Environmental Engineering* **9** (2015), no. 2.

[49] R. Anderson, D. Gallup, J. T. Barron, J. Kontkanen, N. Snavely, C. Hernández, S. Agarwal, and S. M. Seitz, *Jump: virtual reality video*, *ACM Transactions on Graphics (TOG)* **35** (2016), no. 6 198.

[50] Y. Zhang, S. Song, P. Tan, and J. Xiao, *PanoContext: A Whole-Room 3D Context Model for Panoramic Scene Understanding*, in *ECCV: European Conference on Computer Vision*, pp. 668–686, 2014.

[51] "Immersive 3D for the Real World." `https://matterport.com/`, 2017. Accessed: 2017-07-01.

[52] Y. Horry, K.-I. Anjyo, and K. Arai, *Tour into the picture: using a spidery mesh interface to make animation from a single image*, in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 225–232, ACM Press/Addison-Wesley Publishing Co., 1997.

[53] A. Criminisi, I. Reid, and A. Zisserman, *Single view metrology*, in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, pp. 434–441, IEEE, 1999.

[54] L. Zhang, G. Dugas-Phocion, J.-S. Samson, and S. M. Seitz, *Single-view modelling of free-form scenes*, *Computer Animation and Virtual Worlds* **13** (2002), no. 4 225–235.

[55] A. van den Hengel, A. Dick, T. Thormählen, B. Ward, and P. H. S. Torr, *Videotrace: Rapid interactive scene modelling from video*, *ACM Trans. Graph.* **26** (July, 2007).

[56] J. Whyte, N. Bouchlaghem, A. Thorpe, and R. McCaffer, *From cad to virtual reality: modelling approaches, data exchange and interactive 3d building design tools*, *Automation in construction* **10** (2000), no. 1 43–55.

[57] J. Liang and M. Green, *Jdcad: A highly interactive 3d modeling system*, *Computers & graphics* **18** (1994), no. 4 499–506.

[58] J. Butterworth, A. Davidson, S. Hench, and M. T. Olano, *3dm: A three dimensional modeler using a head-mounted display*, in *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, I3D '92, (New York, NY, USA), pp. 135–138, ACM, 1992.

[59] M. F. Deering, *Holosketch: A virtual reality sketching/animation tool*, *ACM Trans. Comput.-Hum. Interact.* **2** (Sept., 1995) 220–238.

[60] D. P. Mapes and J. M. Moshell, *A two-handed interface for object manipulation in virtual environments*, *Presence: Teleoper. Virtual Environ.* **4** (Jan., 1995) 403–416.

[61] J. LaViola, E. Kruijff, D. Bowman, R. McMahan, and I. Poupyrev, *3D User Interfaces: Theory and Practice.* Usability Series. Pearson Education, Limited, 2017.

[62] B. Jackson and D. F. Keefe, *Lift-off: Using reference imagery and freehand sketching to create 3d models in vr*, *IEEE Transactions on Visualization and Computer Graphics* **22** (Apr., 2016) 1442–1451.

[63] "Oculus Medium v1.1.2, now Adobe Medium." `https://www.oculus.com/experiences/rift/1336762299669605/`, 2017. Accessed: 2017-07-01.

[64] "Create 3D models in VR - Google VR." `https://vr.google.com/blocks/`, 2017. Accessed: 2017-07-01.

[65] R. Stoakley, M. J. Conway, and R. Pausch, *Virtual reality on a wim: interactive worlds in miniature*, in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 265–272, ACM Press/Addison-Wesley Publishing Co., 1995.

[66] M. Gleicher, *Image snapping*, in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, (New York, NY, USA), pp. 183–190, ACM, 1995.

[67] C. Cadena, Y. Latif, and I. D. Reid, *Measuring the performance of single image depth estimation methods*, in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4150–4157, IEEE, oct, 2016.

[68] A. Saxena, Min Sun, and A. Ng, *Make3D: Learning 3D Scene Structure from a Single Still Image*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31** (may, 2009) 824–840.

[69] C. Li, A. Kowdle, A. Saxena, and T. Chen, *Towards holistic scene understanding: feedback enabled cascaded classification models*, in *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, pp. 1351–1359, Curran Associates Inc., 2010.

[70] "Unity engine v. 2017.1." `https://unity3d.com/`, 2017. Accessed: 2017-07-01.

[71] D. Hoiem, A. A. Efros, and M. Hebert, *Geometric context from a single image*, in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 1, pp. 654–661 Vol. 1, Oct, 2005.

[72] A. Saxena, S. H. Chung, and A. Y. Ng, *Learning depth from single monocular images*, in *NIPS*, vol. 18, pp. 1–8, 2005.

[73] A. Cherian, V. Morellas, and N. Papanikolopoulos, *Accurate 3d ground plane estimation from a single image*, in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 2243–2249, IEEE, 2009.

[74] E. Sayyad, M. Sra, and T. Höllerer, *Walking and teleportation in wide-area virtual reality experiences*, in *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 608–617, IEEE, 2020.

[75] Wikipedia, *Virtual Reality Roller Coaster*, 2020 (accessed May 16, 2020).

[76] S. Hayden, *Giant $500M Chinese VR Theme Park Opens Doors to the Public*, 2018 (accessed May 17, 2020).

[77] A. Nierenberg, *Flavorful Bites in a Virtual Reality*, 2020 (accessed May 16, 2020).

[78] Void, *The VOID, A Virtual Reality Experience*, 2020 (accessed May 12, 2020).

[79] Dreamscape Immersive, *Dreamscape*, 2020 (accessed May 12, 2020).

[80] D. Waller, E. Bachmann, E. Hodgson, and A. C. Beall, *The HIVE: A huge immersive virtual environment for research in spatial cognition*, *Behavior Research Methods* **39** (2007), no. 4 835–843.

[81] L.-P. Cheng, E. Ofek, C. Holz, and A. D. Wilson, *VRoamer: generating on-the-fly VR experiences while walking inside large, unknown real-world building environments*, in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 359–366, 2019.

[82] J. J. Yang, C. Holz, E. Ofek, and A. D. Wilson, *Dreamwalker: Substituting real-world walking experiences with a virtual reality*, in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pp. 1093–1107, ACM, 2019.

[83] E. Langbehn, P. Lubos, and F. Steinicke, *Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking*, in *Proceedings of the Virtual Reality International Conference-Laval Virtual*, p. 4, ACM, 2018.

[84] F. Buttussi and L. Chittaro, *Locomotion in place in virtual reality: A comparative evaluation of joystick, teleport, and leaning*, *IEEE transactions on visualization and computer graphics* (2019).

[85] C. Zanbaka, S. Babu, D. Xiao, A. Ulinski, L. Hodges, and B. Lok, *Effects of travel technique on cognition in virtual environments*, in *IEEE Virtual Reality 2004*, pp. 149–286, 2004.

[86] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks Jr, *Walking> walking-in-place> flying, in virtual environments*, in *Proceedings of the 26th annual conference on computer graphics and interactive techniques*, pp. 359–364, ACM Press/Addison-Wesley Publishing Co., 1999.

[87] R. A. Ruddle and S. Lessels, *The benefits of using a walking interface to navigate virtual environments*, *ACM Transactions on Computer-Human Interaction (TOCHI)* **16** (2009), no. 1 1–18.

[88] S. S. Chance, F. Gaunet, P. M. Berthelot, A. C. Beall, *et. al.*, *Locomotion mode affects the updating of objects...*, in *Presence*, Citeseer, 1998.

[89] E. A. Suma, S. L. Finkelstein, S. Clark, P. Goolkasian, and L. F. Hodges, *Effects of travel technique and gender on a divided attention task in a virtual environment*, in *2010 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 27–34, IEEE, 2010.

[90] E. A. Suma, Z. Lipps, S. Finkelstein, D. M. Krum, and M. Bolas, *Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture*, *IEEE Transactions on Visualization and Computer Graphics* **18** (2012), no. 4 555–564.

[91] C. A. Zanbaka, B. C. Lok, S. V. Babu, A. C. Ulinski, and L. F. Hodges, *Comparison of path visualizations and cognitive measures relative to travel technique in a virtual environment*, *IEEE Transactions on Visualization and Computer Graphics* **11** (2005), no. 6 694–705.

[92] E. A. Suma, S. L. Finkelstein, M. Reid, A. Ulinski, and L. F. Hodges, *Real walking increases simulator sickness in navigationally complex virtual environments*, in *2009 IEEE Virtual Reality Conference*, pp. 245–246, 2009.

[93] D. Bowman, E. Kruijff, J. J. LaViola Jr, and I. P. Poupyrev, *3D User interfaces: theory and practice, CourseSmart eTextbook.* Addison-Wesley, 2004.

[94] Niantic Inc., "Pokémon Go." `https://www.pokemongo.com/en-us/`.

[95] A. D. Cheok, K. H. Goh, W. Liu, F. Farbiz, S. W. Fong, S. L. Teo, Y. Li, and X. Yang, *Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing*, *Personal and ubiquitous computing* **8** (2004), no. 2 71–81.

[96] M. Serino, K. Cordrey, L. McLaughlin, and R. L. Milanaik, *Pokémon Go and augmented virtual reality games: a cautionary commentary for parents and pediatricians*, *Current opinion in pediatrics* **28** (2016), no. 5 673–677.

[97] F. Steinicke, Y. Visell, J. Campos, and A. Lécuyer, *Human walking in virtual environments*, vol. 56. Springer, 2013.

[98] S. Razzaque, Z. Kohn, and M. C. Whitton, *Redirected walking.* PhD thesis, University of North Carolina at Chapel Hill, 2005.

[99] B. Williams, G. Narasimham, B. Rump, T. P. McNamara, T. H. Carr, J. Rieser, and B. Bodenheimer, *Exploring large virtual environments with an HMD when physical space is limited*, in *Proceedings of the 4th symposium on Applied perception in graphics and visualization*, pp. 41–48, ACM, 2007.

[100] M. Sra, X. Xu, A. Mottelson, and P. Maes, *VMotion: Designing a Seamless Walking Experience in VR*, in *Proceedings of the 2018 Designing Interactive Systems Conference*, pp. 59–70, ACM, 2018.

[101] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe, *Estimation of detection thresholds for redirected walking techniques*, *IEEE transactions on visualization and computer graphics* **16** (2009), no. 1 17–27.

[102] T. Grechkin, J. Thomas, M. Azmandian, M. Bolas, and E. Suma, *Revisiting detection thresholds for redirected walking: Combining translation and curvature gains*, in *Proceedings of the ACM Symposium on Applied Perception*, pp. 113–120, 2016.

[103] C. Boletsis, *The new era of virtual reality locomotion: A systematic literature review of techniques and a proposed typology*, *Multimodal Technologies and Interaction* **1** (2017), no. 4 24.

[104] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey, *Locomotion in virtual reality for individuals with autism spectrum disorder*, in *Proceedings of the 2016 Symposium on Spatial User Interaction*, pp. 33–42, ACM, 2016.

[105] N. Coomer, S. Bullard, W. Clinton, and B. Williams-Sanders, *Evaluating the effects of four vr locomotion methods: joystick, arm-cycling, point-tugging, and teleporting*, in *Proceedings of the 15th ACM symposium on applied perception*, pp. 1–8, 2018.

[106] C. G. Christou and P. Aristidou, *Steering versus teleport locomotion for head mounted displays*, in *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pp. 431–446, Springer, 2017.

[107] J. Frommel, S. Sonntag, and M. Weber, *Effects of controller-based locomotion on player experience in a virtual reality exploration game*, in *Proceedings of the 12th international conference on the foundations of digital games*, pp. 1–6, 2017.

[108] J. Bhandari, P. MacNeilage, and E. Folmer, *Teleportation without spatial disorientation using optical flow cues*, in *Proceedings of Graphics Interface*, vol. 2018, 2018.

[109] B. G. Witmer and M. J. Singer, *Measuring presence in virtual environments: A presence questionnaire*, *Presence* **7** (1998), no. 3 225–240.

[110] C. Games, *VR Navigation*, 2020 (accessed May 5, 2020).

[111] Aldin, *Introducing Next Generation Telepath VR Movement Features*, 2020. https://medium.com/aldin-dynamics/introducing-next-generation-telepath-vr-movement-features-6417a7e0ff49 (accessed May 5, 2020).

[112] B. Williams, G. Narasimham, T. P. McNamara, T. H. Carr, J. J. Rieser, and B. Bodenheimer, *Updating orientation in large virtual environments using scaled translational gain*, in *Proceedings of the 3rd symposium on Applied perception in graphics and visualization*, pp. 21–28, 2006.

[113] V. Interrante, B. Ries, and L. Anderson, *Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments*, in *2007 IEEE Symposium on 3D User Interfaces*, IEEE, 2007.

[114] C. T. Neth, J. L. Souman, D. Engel, U. Kloos, H. H. Bulthoff, and B. J. Mohler, *Velocity-dependent dynamic curvature gain for redirected walking*, *IEEE transactions on visualization and computer graphics* **18** (2012), no. 7 1041–1052.

[115] F. Steinicke, G. Bruder, T. Ropinski, and K. Hinrichs, *Moving towards generally applicable redirected walking*, in *Proceedings of the Virtual Reality International Conference (VRIC)*, pp. 15–24, IEEE Press, 2008.

[116] F. Steinicke, G. Bruder, K. Hinrichs, A. Steed, and A. L. Gerlach, *Does a gradual transition to the virtual world increase presence?*, in *2009 IEEE Virtual Reality Conference*, pp. 203–210, IEEE, 2009.

[117] Q. Sun, A. Patney, L.-Y. Wei, O. Shapira, J. Lu, P. Asente, S. Zhu, M. McGuire, D. Luebke, and A. Kaufman, *Towards virtual reality infinite walking: dynamic saccadic redirection*, *ACM Transactions on Graphics (TOG)* **37** (2018), no. 4 1–13.

[118] E. A. Suma, G. Bruder, F. Steinicke, D. M. Krum, and M. Bolas, *A taxonomy for deploying redirection techniques in immersive virtual environments*, in *2012 IEEE Virtual Reality Workshops (VRW)*, pp. 43–46, IEEE, 2012.

[119] D. R. Montello, *A new framework for understanding the acquisition of spatial knowledge in large-scale environments*, *Spatial and temporal reasoning in geographic information systems* (1998) 143–154.

[120] R. A. Ruddle, E. Volkova, and H. H. Bülthoff, *Walking improves your cognitive map in environments that are large-scale and large in extent*, *ACM Transactions on Computer-Human Interaction (TOCHI)* **18** (2011), no. 2 10.

[121] T. C. Peck, H. Fuchs, and M. C. Whitton, *An evaluation of navigational ability comparing redirected free exploration with distractors to walking-in-place and joystick locomotio interfaces*, in *2011 IEEE Virtual Reality Conference*, pp. 55–62, IEEE, 2011.

[122] S. Freitag, D. Rausch, and T. Kuhlen, *Reorientation in virtual environments using interactive portals*, in *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 119–122, IEEE, 2014.

[123] M. Hegarty, A. E. Richardson, D. R. Montello, K. Lovelace, and I. Subbiah, *Development of a self-report measure of environmental spatial ability*, *Intelligence* **30** (2002), no. 5 425–447.

[124] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, *Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness*, *The international journal of aviation psychology* **3** (1993), no. 3 203–220.

[125] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, *Matterport3D: Learning from RGB-D data in indoor environments*, *International Conference on 3D Vision [(3DV)]* (2017).

[126] M. Usoh, E. Catena, S. Arman, and M. Slater, *Using presence questionnaires in reality*, *Presence: Teleoperators & Virtual Environments* **9** (2000), no. 5 497–503.

[127] M. Billinghurst and S. Weghorst, *The use of sketch maps to measure cognitive maps of virtual environments*, in *Proceedings Virtual Reality Annual International Symposium'95*, pp. 40–47, IEEE, 1995.

[128] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[129] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417–424, ACM Press/Addison-Wesley Publishing Co., 2000.

[130] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari, *Image inpainting: A review*, *Neural Processing Letters* (2019) 1–22.

[131] A. A. Efros and T. K. Leung, *Texture synthesis by non-parametric sampling*, in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1033–1038, IEEE, 1999.

[132] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, *Summarizing visual data using bidirectional similarity*, in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[133] T. S. Cho, M. Butman, S. Avidan, and W. T. Freeman, *The patch transform and its applications to image editing*, in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[134] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, *Patchmatch: A randomized correspondence algorithm for structural image editing*, in *ACM Transactions on Graphics (ToG)*, vol. 28, p. 24, ACM, 2009.

[135] K. H. Jin and J. C. Ye, *Annihilating Filter-Based Low-Rank Hankel Matrix Approach for Image Inpainting*, *IEEE Transactions on Image Processing* **24** (2015), no. 11 3498–3511.

[136] N. Kawai, T. Sato, and N. Yokoya, *Diminished Reality Based on Image Inpainting Considering Background Geometry*, *IEEE Transactions on Visualization and Computer Graphics* **22** (2016), no. 3 1236–1247.

[137] Q. Guo, S. Gao, X. Zhang, Y. Yin, and C. Zhang, *Patch-Based Image Inpainting via Two-Stage Low Rank Approximation*, *IEEE Transactions on Visualization and Computer Graphics* **24** (2018), no. 6 2023–2036.

[138] H. Xue, S. Zhang, and D. Cai, *Depth Image Inpainting: Improving Low Rank Matrix Completion with Low Gradient Regularization*, *IEEE Transactions on Image Processing* **26** (2017), no. 9 4311–4320, [arXiv:1604.0581].

[139] J. Liu, F. Yu, and T. Funkhouser, *Interactive 3D Modeling with a Generative Adversarial Network*, 2018.

[140] D. Ding, S. Ram, and J. J. Rodriguez, *Image inpainting using nonlocal texture matching and nonlinear filtering*, *IEEE Transactions on Image Processing* **28** (2019), no. 4 1705–1719.

[141] J. Duan, Z. Pan, B. Zhang, W. Liu, and X. C. Tai, *Fast algorithm for color texture image inpainting using the non-local CTV model*, *Journal of Global Optimization* **62** (2015), no. 4 853–876.

[142] W. Jiang, *Rate-distortion optimized image compression based on image inpainting*, *Multimedia Tools and Applications* **75** (2016), no. 2 919–933.

[143] Y. Wang, Z. Luo, and P. M. Jodoin, *Interactive deep learning method for segmenting moving objects*, *Pattern Recognition Letters* **96** (2017), no. 4 66–75, [arXiv:1712.0153].

[144] X. Li, Y. Dong, P. Peers, and X. Tong, *Modeling Surface Appearance from a Single Photograph using Self-augmented Convolutional Neural Networks*, *ACM Transactions on Graphics* **36** (2017), no. 4 3050–3064, [arXiv:1809.0088].

[145] K. Li, Y. Wei, Z. Yang, and W. Wei, *Image inpainting algorithm based on TV model and evolutionary algorithm*, *Soft Computing* **20** (2016), no. 3 885–893.

[146] G. Sridevi and S. Srinivas Kumar, *Image Inpainting Based on Fractional-Order Nonlinear Diffusion for Image Reconstruction*, *Circuits, Systems, and Signal Processing* **38** (2019), no. 8 3802–3817.

[147] Z. Yan, X. Li, M. Li, W. Zuo, and S. Shan, *Shift-net: Image inpainting via deep feature rearrangement*, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11218 LNCS, pp. 3–19. Springer International Publishing, Cham, 2018. arXiv:1801.0939.

[148] J. Xie, L. Xu, and E. Chen, *Image denoising and inpainting with deep neural networks*, in *Advances in neural information processing systems*, pp. 341–349, 2012.

[149] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, *Image inpainting for irregular holes using partial convolutions*, in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 85–100, 2018.

[150] C. Hsu, F. Chen, and G. Wang, *High-resolution image inpainting through multiple deep networks*, in *2017 International Conference on Vision, Image and Signal Processing (ICVISP)*, pp. 76–81, IEEE, 2017.

[151] Y.-L. Chang, Z. Yu Liu, and W. Hsu, *Vornet: Spatio-temporally consistent video inpainting for object removal*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[152] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[153] C. Zheng, T.-J. Cham, and J. Cai, *Pluralistic image completion*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1438–1447, 2019.

[154] Y. Chen and H. Hu, *An improved method for semantic image inpainting with gans: Progressive inpainting*, *Neural Processing Letters* **49** (2019), no. 3 1355–1367.

[155] Y.-G. Shin, M.-C. Sagong, Y.-J. Yeo, S.-W. Kim, and S.-J. Ko, *Pepsi++: Fast and lightweight network for image inpainting*, *arXiv preprint arXiv:1905.09010* (2019).

[156] H. Wang, L. Jiao, H. Wu, and R. Bie, *New inpainting algorithm based on simplified context encoders and multi-scale adversarial network*, *Procedia computer science* **147** (2019) 254–263.

[157] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-image translation with conditional adversarial networks*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[158] J. B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, *Image completion using planar structure guidance*, 2014.

[159] K. He and J. Sun, *Image completion approaches using the statistics of similar patches*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36** (2014), no. 12 2423–2435.

[160] D. Lee, S. Yun, S. Choi, H. Yoo, M. H. Yang, and S. Oh, *Unsupervised Holistic Image Generation from Key Local Patches*, 2018.

[161] Y. Zhang, J. Xiao, J. Hays, and P. Tan, *Framebreak: Dramatic image extrapolation by guided shift-maps*, 2013.

[162] M. Wang, Y. K. Lai, Y. Liang, R. R. Martin, and S. M. Hu, *Biggerpicture: Data-driven image extrapolation using graph matching*, 2014.

[163] H.-Y. F. Tung, R. Cheng, and K. Fragkiadaki, *Learning spatial common sense with geometry-aware recurrent networks*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2595–2603, 2019.

[164] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, *Scene representation networks: Continuous 3D-structure-aware neural scene representations*, in *Advances in Neural Information Processing Systems*, pp. 1119–1130, 2019.

[165] M. Meshry, D. B. Goldman, S. Khamis, H. Hoppe, R. Pandey, N. Snavely, and R. Martin-Brualla, *Neural Rerendering in the Wild*, 2019.

[166] J. Thies, M. Zollhöfer, and M. Nießner, *Deferred neural rendering: Image Synthesis using Neural Textures*, *ACM Transactions on Graphics* **38** (2019), no. 4 [arXiv:1904.1235].

[167] L. Yi, H. Su, X. Guo, and L. Guibas, *SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation*, in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 6584–6592, 2017. arXiv:1612.0060.

[168] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, *MeshCNN: A network with an edge*, *ACM Transactions on Graphics* **38** (2019), no. 4 1–12, [arXiv:1809.0591].

[169] Y. Horry, K.-I. Anjyo, and K. Arai, *Tour into the picture: using a spidery mesh interface to make animation from a single image*, in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 225–232, 1997.

[170] D. Eigen, C. Puhrsch, and R. Fergus, *Depth map prediction from a single image using a multi-scale deep network*, 2014.

[171] L. Ladický, J. Shi, and M. Pollefeys, *Pulling things out of perspective*, 2014.

[172] F. Liu, C. Shen, and G. Lin, *Deep convolutional neural fields for depth estimation from a single image*, 2015.

[173] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille, *Towards unified depth and semantic prediction from a single image*, 2015.

[174] 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* **38** (2016), no. 10 2024–2039, [arXiv:1502.0741].

[175] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, *Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries*, 2019.

[176] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau, *Flexible SVBRDF Capture with a Multi-Image Deep Network*, *Computer Graphics Forum* **38** (2019), no. 4 1–13, [arXiv:1906.1155].

[177] Z. Li, Z. Xu, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, *Learning to reconstruct shape and spatially-varying reflectance from a single image*, *SIGGRAPH Asia 2018 Technical Papers, SIGGRAPH Asia 2018* **37** (2018), no. 6.

[178] Z. Li, K. Sunkavalli, and M. Chandraker, *Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image*, 2018.

[179] P. Tan, G. Zeng, J. Wang, S. B. Kang, and L. Quan, *Image-based tree modeling*, 2007.

[180] R. Smelik, K. Galka, K. J. De Kraker, F. Kuijper, and R. Bidarra, *Semantic constraints for procedural generation of virtual worlds*, 2011.

[181] S. Ramalingam and M. Brand, *Lifting 3d manhattan lines from a single image*, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 497–504, 2013.

[182] G. Nishida, A. Bousseau, and D. G. Aliaga, *Procedural modeling of a building from a single image*, *Computer Graphics Forum* **37** (2018), no. 2 415–429.

[183] A. X. Chang, M. Savva, and C. D. Manning, *Learning spatial knowledge for text to 3d scene generation*, in *EMNLP*, 2014.

[184] C. Zou, J.-W. Su, C.-H. Peng, A. Colburn, Q. Shan, P. Wonka, H.-K. Chu, and D. Hoiem, *3D Manhattan Room Layout Reconstruction from a Single 360 Image*, 2019.

[185] T. Kelly, P. Guerrero, A. Steed, P. Wonka, and N. J. Mitra, *Frankengan: Guided detail synthesis for building mass models using style-synchonized GANs*, *SIGGRAPH Asia 2018 Technical Papers, SIGGRAPH Asia 2018* **37** (2018), no. 6 [arXiv:1806.0717].

[186] S. Kim, D. Kim, and S. Choi, *CityCraft: 3D virtual city creation from a single image*, *Visual Computer* (2019) 1–14.

[187] C. Y. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich, *RoomNet: End-to-End Room Layout Estimation*, 2017.

[188] S. Huang, S. Qi, Y. Zhu, Y. Xiao, Y. Xu, and S. C. Zhu, *Holistic 3D scene parsing and reconstruction from a single RGB image*, 2018.

[189] C. Zou, A. Colburn, Q. Shan, and D. Hoiem, *LayoutNet: Reconstructing the 3D Room Layout from a Single RGB Image*, 2018.

[190] H. J. Lin, S. W. Huang, S. H. Lai, and C. K. Chiang, *Indoor Scene Layout Estimation from a Single Image*, 2018.

[191] C. Fernandez-Labrador, A. Perez-Yus, G. Lopez-Nicolas, and J. J. Guerrero, *Layouts from panoramic images with geometry and deep learning*, 2018.

[192] H. Howard-Jenkins, S. Li, and V. Prisacariu, *Thinking Outside the Box: Generation of Unconstrained 3D Room Layouts*, 2019.

[193] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et. al.*, *Photo-realistic single image super-resolution using a generative adversarial network*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.

[194] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[195] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, *Encoding in style: a stylegan encoder for image-to-image translation*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2287–2296, 2021.

[196] E. Ntavelis, A. Romero, I. Kastanis, L. V. Gool, and R. Timofte, *Sesame: Semantic editing of scenes by adding, manipulating or erasing objects*, in *European Conference on Computer Vision*, pp. 394–411, Springer, 2020.

[197] T. C. Wang, M. Y. Liu, J. Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, *High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs*, Tech. Rep. 8, 2018.

[198] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, *Semantic image synthesis with spatially-adaptive normalization*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2337–2346, 2019.

[199] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, *arXiv preprint arXiv:1312.6114* (2013).

[200] A. Karnewar and O. Wang, *Msg-gan: Multi-scale gradients for generative adversarial networks*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7799–7808, 2020.

[201] V. Sushko, E. Schönfeld, D. Zhang, J. Gall, B. Schiele, and A. Khoreva, *You only need adversarial supervision for semantic image synthesis*, *arXiv preprint arXiv:2012.04781* (2020).

[202] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, *Semantic understanding of scenes through the ade20k dataset*, *International Journal of Computer Vision* **127** (2019), no. 3 302–321.

[203] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, *Maskgan: Towards diverse and interactive facial image manipulation*, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[204] R. J. Sternberg and K. Sternberg, *Cognitive psychology.* Nelson Education, 2016.

[205] D. B. Willingham, M. J. Nissen, and P. Bullemer, *On the development of procedural knowledge.*, *Journal of experimental psychology: learning, memory, and cognition* **15** (1989), no. 6 1047.

[206] E. Tulving *et. al.*, *Episodic and semantic memory*, *Organization of memory* **1** (1972) 381–403.

[207] K. Patterson, P. J. Nestor, and T. T. Rogers, *Where do you know what you know? the representation of semantic knowledge in the human brain*, *Nature Reviews Neuroscience* **8** (2007), no. 12 976.

[208] D. L. Scarborough, *Memory for brief visual displays of symbols*, *Cognitive Psychology* **3** (1972), no. 3 408–429.

[209] E. K. Vogel, G. F. Woodman, and S. J. Luck, *Storage of features, conjunctions, and objects in visual working memory, Journal of Experimental Psychology: Human Perception and Performance* **27** (2001), no. 1 92–114.