GREEN, C. 2024. *Optimising for holistic VR interaction*. Robert Gordon University, PhD thesis. Hosted on OpenAIR [online]. Available from: https://doi.org/10.48526/rgu-wt-2565290

Optimising for holistic VR interaction.

GREEN, C.

2024

Optimising for Holistic VR Interaction

Corrie Green

A thesis submitted in partial fulfillment of the requirements of

Robert Gordon University

for the degree of Doctor of Philosophy

May 2024

Supervisors Dr. Yang Jiang, Dr. John Isaacs

“*We live in a physical world whose properties we have come to know well through long familiarity. We sense an involvement with this physical world which gives us the ability to predict its properties well. For example, we can predict where objects will fall, how well-known shapes look from other angles, and how much force is required to push objects against friction.*”

“The Ultimate Display”, Ivan Sutherland 1965

Abstract

This thesis aims to understand the limitations of human perception to develop egalitarian, accessible interface solutions designed specifically for virtual reality (VR) environments.

The added dimension provided by VR has presented unique challenges for developers during interaction and interface design. Existing 2D interface design philosophies do not necessarily translate directly to VR, such as the use of a VR laser pointer for ray interaction which is designed to that of a traditional cursor.

It was discovered by integrating the 1-Euro Filter for ray-based interaction in VR, we can reduce task completion time by mitigating the effects of benign essential tremors. In this, we place a low-pass filter between the user's real-world motion and their virtual input. The study highlighted the necessity for a filter to accommodate Fitts' Law, demonstrating that as interaction distance increases, so does completion time.

By exploiting our proprioception ability, the creation of the modular 3D interface was then developed and studied through an iterative, accessible first design process. Informed by human ergonomics and anthropometrics, the interface is designed from the ground up to support integration of current and future input modalities such as eye-tracking, voice, and brain-computer interfaces. Striving to empower user preference over developer discretion.

Review of optimisation stages in the VR render pipeline reveals the constraints imposed by current head mounted display (HMD) and lens technologies. Highlighting the necessity for developers to exploit flaws in the human optical system to produce a seamless visual experience in VR while being bound by processing hardware constraints. From this, the introduction of a novel addition to the pipeline is made. 'Frameless Eye-tracked Foveated Rendering,' where we leverage eye-tracking technology further with frameless rendering.

Keywords: Virtual Reality, Human Computer Interaction, Accessibility, 3D Interface

Impact Statement

The research presented in this thesis has strived to advance interaction and optimisation in immersive environments. It has resulted in 1 peer-reviewed publication [1] presented at HCI International 2023, 1 paper pending journal feedback [2] and unique 2 published pre-prints [3] [4] pending submission at an appropriate journal as of thesis submission. Chapter 5 is also in development for journal publication as a review piece.

The primary outcomes of this research have addressed gaps in accessible VR interaction, visualisation, and optimisation, focusing on human factors in the VR domain. Striving to enhance the user experience and accessibility. Demos and source code for the discussed solutions in the presented chapters are available on GitHub which can be found in the associated repositories section on the next page.

Summary of Publications

C. Green, D. Y. Jiang, D. J. Isaacs, and D. M. Heron, “Tremor Reduction for Accessible Ray Based Interaction in VR Applications,” May 2024, Accessed: May 16, 2024. [Online]. Available: https://arxiv.org/abs/2405.07335v1

C. Green, Y. Jiang, and J. Isaacs, “Modular 3D Interface Design for Accessible VR Applications,” pp. 15–32, 2023, doi: 10.1007/978-3-031-35634-6_2/TABLES/2.

C. Green, C. Farr, Y. Jiang, J. Isaacs, “An Accessible Toolkit for 360 VR Studies,” Apr. 2023, doi: 10.1016/B978-0-12-818697-8.00006-6.

C. Green, “Augmenting Heritage: An Open-Source Multiplatform AR Application,” Sep. 2023, doi: 10.3390/APP12199859.

Associated Repositories

“GitHub - corriedotdev/VR-Tremor-Reduction.” Accessed: Nov. 01, 2023. [Online]. Available: https://github.com/corriedotdev/vr-tremor-reduction

“GitHub - corriedotdev/vr-modular-3d-gui.” Accessed: Nov. 01, 2023. [Online]. Available: https://github.com/corriedotdev/vr-modular-3d-gui

“GitHub - corriedotdev/vr-360-player.” Accessed: Nov. 01, 2023. [Online]. Available: https://github.com/corriedotdev/vr-360-player

“GitHub - corriedotdev/3D-Museum-Library.” Accessed: Nov. 01, 2023. [Online]. Available: https://github.com/corriedotdev/3D-Museum-Library

Acknowledgments

Such positive reactions from friends, family, and colleagues exploring my VR demos over the years showed the potential for VR's impact as a tool for sharing a collective experience, and for this, I cannot express my gratitude enough. Watching you explore digital worlds provided me with the motivation needed for this path less traveled. Thank you for your willingness to step into my often flawed, but always impassioned virtual experiments.

I would like to express my heartfelt gratitude to my supervision team, Yang and John, for their invaluable insights, expertise, and motivation throughout this journey. The support received from starting higher education has meant more than I can articulate. I extend my appreciation to the wider computing team at Robert Gordon University. Your engagement in my work and our discussions over the years have significantly shaped my academic mindset and research philosophy.

I hope the presented thesis provides more context for an accessible vision of the future and showcases why I've been so passionate about the work and hope it's a step in the right direction for heightening the human experience.

I'd like to express my gratitude to my parents and brother. To my grandma Shona Green. To my late grandparents, Papa John Dunlop and Granda Eddie Green whom I hope are proud of the work. My Nan Marilyn Dunlop whose blindness inspired so much of my work in VR accessibility, whom challenged conventional belief from a young age. Lastly a heartfelt appreciation to my incredible fiancé for all the support and encouragement throughout this journey.

Declaration

I confirm that the work contained in this PhD thesis has been composed solely by myself – Corrie Green - and has not been accepted in any previous application for a degree. All sources of information have been specifically acknowledged and all verbatim extracts are distinguished by quotation marks.

Contents

List of Figures

List of Tables

Table of Equations

Authors Figure Attribution Table

All figures listed below were created by the author Corrie Green

Figure No.	Page No.	Date Created
1.1	11	01/01/23
2.1	18	11/05/22
2.2	23	11/05/22
2.5	26	30/06/22
2.6	29	21/02/23
2.7	30	06/09/22
2.8	33	06/12/22
2.9	35	21/09/22
2.10	36	04/04/23
2.11	37	04/04/23
3.1	42	24/10/22
3.2	44	17/05/23
3.3	45	27/12/22
3.4	45	06/02/23
3.5	47	02/01/23
3.6	50	07/05/22
3.7	51	06/09/22
4.1	63	15/09/23
4.2	68	30/05/23
4.5	77	28/02/23
4.8	83	18/06/21
5.1	93	06/01/22
5.2	95	05/01/22
5.3	101	19/06/23
5.4	102	17/03/23
5.5	104	19/06/23
5.6	105	17/05/23
5.7	105	19/06/23
5.8	113	26/01/22
5.9	113	26/01/22
5.10	113	26/01/22
7.1	138	22/08/23
7.2	138	01/09/22
7.3	139	21/10/23

Abbreviations

HMD	Head Mounted Display
VR	Virtual Reality
AR	Augmented Reality
XR	Extended Reality
MR	Mixed Reality
6-DoF	6 Degrees of Freedom
ETFR	Eye-tracked foveated rendering
AFR	Adaptive Frameless Rendering
FETFR	Frameless Eye-tracked foveated rendering
VBI	Vertical blanking interval
VAC	Vergence Accommodation Conflict
LBS	Laser beam scanning
FoV	Field of View
SS	Simulator Sickness

This page is intentionally left blank

Chapter 1

Introduction

The structure of this thesis begins with an overview of the history of VR, aiming to identify the problems faced from early adoption to present day, specifically VR user interfaces, interaction then leading into the rendering pipeline. The literature review explores both writing and existing visualisation applications in VR immersive environments, with the goal of identifying research gaps and discussing trends for existing solutions. The findings from the review lead to discovered insights in 4 key areas: accessibility, interaction, interface, and optimisation.

1.1 Research Questions

The core aim of this thesis is to explore issues inherited from legacy VR systems that have been overlooked during the advancement of VR technology. Specifically focusing on interaction and graphical optimisations. The central research questions guiding this exploration include:

1. What are the historical design decisions for VR rendering and interaction?
2. How can the design of VR interface and interaction be improved, to support both current and future input modalities, particularly, how can ray-based interaction be improved?
3. Is there a better means of general interaction that prioritises user ability, and how should such an interface be designed and developed?
4. Can new input modalities, such as eye-tracking, enhance VR rendering fidelity and how can they be integrated into or modify the existing VR render pipeline?

Chapter 2 explores the problem space further, by identifying existing design constraints when developing VR experiences. Overviewing core accessibility philosophies that should be considered during development, such as an accessible first design approach. Exploration into existing accessibility constraints found in VR interfaces are made, with a discovered enhancement for the most common input medium used in VR interaction, ray-based input. Here it is demonstrated how an algorithm can provide an opportunity for a more accurate and consequently less frustrating experience by filtering and reducing involuntary hand tremors. From this, ethnographic study approaches were used and developed through focus groups and task-based evaluations. The first conducted user study required participants to complete time-based tasks using ray-based interaction with the 1-Euro low pass filter being used at set intervals. The discovered insight shows how interaction could develop in the near and distant future with the use in all VR accessibility menus and multimodal input mediums.

Chapter 3 covers interface approaches for exploiting humans' proprioception ability in 3D space. Striving for a more natural interaction experience when using a 3D graphical user interface over traditional 2D ray-based interfaces. Holistic approaches for interacting with spatial interfaces are discussed, with the development of a novel 3D modular graphical user interface (GUI) where concepts can be used for both VR and augmented reality (AR) applications. A focus group study was conducted to understand users' high and low usability satisfaction, highlighting where the interface could be enhanced and improved upon.

In Chapter 4, we explore critical optimisation techniques for VR head-mounted displays (HMD), discussing both current and prospective VR render pipelines. Highlighting the extensive processing devoted to optical feedback, there's a notable lack of robust control algorithms designed to enhance user input. Furthermore, emerging challenges in display and optics are examined with the introduction to the concept of frameless eye-tracked foveated rendering (FETFR). This chapter adopts a comprehensive stance to identify avenues for improving rendering as we address hardware constraints bound by technical progress. We also discuss the influence of human factors, both in refining the render pipeline and physics-based interactions, drawing connections to prior chapters.

Chapter 5 delves into developed visualisation case studies from curated experiences, emphasizing the storage and processing of extensive point cloud and photogrammetry scans. The increase of sensor arrays for scanning real-world data has led to a surge in heterogeneous datasets. Although this data can be presented in stereoscopic 3D within VR or superimposed using AR, the technical skills required by end users create a barrier to access. The developed applications aim to provide tools for researchers and non-technical users alike to digitally twin and experience scanned content. Playback of 360-degree video also can facilitate this, with rapid access to digital twins of environments by using 360-degree video cameras. The chapter introduces open-source tools for video playback and augmented reality visualisation through mobile AR hosted on a website. It emphasizes the variety of multidimensional data types and storage, sharing their cross-compatibility using case studies on LiDAR and photogrammetry pipelines.

The conclusion draws upon each chapter and aims to link each section together to provide a better understanding of the future in immersive interaction, where designs focusing on user ability in an immersive 3D environment can provide users with the opportunity to draw conclusions with real-world data through its digital counterpart.

1.2 Virtual Reality State of Play

The impact of VR on future generations may not be fully understood until a clear understanding of the present-day challenges are explored. Problem cases that cause barriers to entry for the adoption of VR vary from consumer accessibility to technical limitations in the hardware itself. Key among these challenges include the lack of user familiarity with spatial interface design in immersive environments, resulting in a cognitive accessibility constraint. Immersive environments themselves are digitally created spaces where users can experience a sense of presence and interact as if they are physically within the environment. Headset ergonomics is another challenge, where the current design of most VR headsets are still bulky, heavy on the forehead, and requires fine adjustments to prevent distortion. Low graphics processing also induces motion sickness due to constraints bound by two high-resolution displays with a high refresh rate while still striving for low motion-to-photon latency. Due to VR hardware from manufactures also being fragmented, there is also a toll on developer exposure to proprietary software development kits (SDK) to fully exploit the capabilities offered by headsets. Headset capabilities may not be unanimous across headsets, such as eye tracking support. Following hardware fragmentation, this also leads to content limitations on storefronts where developers cater to specific head-mounted display (HMD) hardware. This contrasts with traditional 2D monitor storefronts where consumer hardware is more predictable. This thesis aims to explore these issues and provide solutions with an exploration into user testing and discussion from use case scenarios. There then leaves us with the question to explore:

How can we improve insight and engagement in a virtual environment?

In 2010 the virtual reality industry moved at pace towards an affordable consumer HMD that provides a fully immersive experience with 3-degree-of-freedom (DoF, 3 degrees of placement, and 3 degrees of rotation) supporting the tracking of head movement and rotation. The Oculus DK1 released in 2013 did not, however, provide support for a tracked control scheme and instead required the use of existing game controllers to interact and engage with the virtual space presented [5]. The two joysticks on Xbox or PlayStation controllers can provide 6-DoF of motion in 3D space by using four input signals. One joystick provides z-axis translation for forward and backward movements with the x-axis supporting side translations.

Another joystick provides pitch and yaw input supporting rotation. However, one joystick is not required with a 3-DoF HMD to get 6-DoF, we instead place that responsibility on the observer's perspective allowing for control of yaw and pitch via head rotation, resulting in support for moving in the environment to just one joystick. With this input mechanism, we are now able to navigate around the presented environment in a full 6-DoF experience. However, the user is not necessarily interacting with the environment, but merely passively viewing the world as a digital observer. Described as an "on-rails" experience, much of the initial technical demonstrations in 2013 for VR headsets used 3-DoF which resulted in many rollercoaster simulated experiences, where the term "on-rails" gets its definition from. Users were forced by the nature of the control scheme provided to view an ever-changing environment rendered from the perspective of an animated camera, without ever having direct interaction with the environment, presenting itself as a passive 360-degree experience.

From this, development challenges arose from how humans can naturally interact and engage with the virtual world's array of rendered objects in 6-DoF. In relation to VR interaction, it refers to the software architecture of the human-machine interface, being translated into a virtual environment system. With research and optimisation approaches to render a 3D environment onto an HMD gaining traction, the next phase of development appeared to be interaction. Oculus shortly released the Oculus touch controllers which provided 6-DoF tracking per controller but with the addition of positional tracking, allowing users to physically walk around their physical play space and have their head and hand positions directly translated from real-world space into virtual space. It should be noted that 9-DoF, where the additional degrees are added are from accelerometers or inertial measurement unit (IMU) devices, are typically discussed in the context of robotics and drones and are not particularly relevant to VR tracking at this time.

Joysticks are still included however on positionally tracked controllers for a variety of reasons. One being the play space available by users may not have enough space to physically walk around and interact with the digital environment on a 1:1 scale due to the finite physical space available [6]. Instead, developers must design locomotion strategies to allow the user to travel distances in the virtual world without having a 1:1 mapping to the physical play space.

As one of the most researched areas in VR academia, locomotion strategies are required specifically for VR, requiring a variety of input controls such as a hybrid movement system, where physical tracking and input with controllers are used together allowing users with positional tracking HMDs to physically walk and interact with the game world, whilst also being able to use joysticks to travel distances in the virtual play space. Although not exclusively using controller joystick input for traveling in virtual spaces, devices such as omnidirectional treadmills, are available that allow users to run in situ.

Much of the input techniques for VR can therefore be translated directly from the traditional gaming industry with similar challenges such as rendering large-scale digital environments and designing the associated interaction mechanics. It is therefore clear that during the early adoption of VR, many design decisions have been directly translated from the traditional 2D gaming industry to support 6-DoF interaction in VR. This raises a key research question that will be expanded on in Chapter 4.

What scenarios can algorithms be placed between the user and their digital input to provide increased usability satisfaction during VR tasks?

1.3 Holistic Approach

The term Holistic is derived from Holism, which is the idea that human behavior should be viewed as a whole integrated experience and not as separate parts, in that systems are interconnected [7]. The definition of a system can be said to be a series of sub-systems that work together to fulfill a purpose. To consider the human system integrating with a VR system is to relate and view the whole of human engagement with the VR hardware and not each in isolation, it is discussed in this section that we need to consider the constraints from both systems – the human and the VR hardware together – to include in its collective design moving forward as the technology evolves. It is from this holistic view of the human VR interface that we can identify emergent properties in its design.

A single definition for VR has been described as “the sum of the hardware and software systems that seek to perfect an all-inclusive, sensory illusion of being present in another environment” [8]. Although not exclusively striving for an all-inclusive sensory illusion, specific sensory stimulus parameters can be exploited to enhance presence in a space. In addition to stimulating the senses, we also have the recent ability to use our body as an input medium itself, with the advancement of hand tracking, eye-tracking, and brain-computer interfaces (BCI) which should now be considered throughout this discussion.

Input parameters have predominantly been controller wand based where users navigate interfaces with buttons and joysticks. With passthrough technology supporting hand tracking now, there is a strive for natural interaction techniques. Furthermore, in a BCI study comparing VR with 2D displays [9], it was shown that BCI showed more promise for accuracy in VR than the 2D display counterpart. The researchers noted that the inaccurate noisy inputs, training, and command learning pipeline needs improvement before integration into real-world systems. In VR development we must consider all technology limitations, such as using fixed focal distance lenses. Further discussion on these issues this causes can be found in Section 2.2.2 and Section 4.3.

To discuss, it could be assumed that noisy electroencephalogram (EEG) signals trained on real-world data would not necessarily translate into the same EEG signals in VR for object recognition, as the clarity of the trained BCI model is not the same as reality. For example, a cup 2m away from a user in reality will result in the recording of high-fidelity EEG signals for object recognition. To then view the same cup's digital twin in VR, we need to consider the range at which the cup is in focus due to the fixed focal lenses found in current VR headsets, resulting in – what we can assume – more noise in the signal as the object is not as clear as the real-world counterpart. This is an example of a fundamental problem in EEG use in VR until adaptive lenses are integrated and general noise from EEG devices can be mitigated. However, the above is only an example to help describe the importance of an holistic view, including current hardware. This theoretical study could be completed in future work.

This scenario highlights the importance of jointly examining human capabilities and VR input hardware to identify limitations about how humans perceive and interact with virtual environments. Exclusively focusing on one input medium appears to be counterproductive to the result we want if limitations are not considered. Currently, not all VR interfaces incorporate multiple input modalities, as technical limitations associated with the various sensor systems still need to be discovered.

Only then can we start to rediscover how the human can interface with new sensor hardware. Therefore, a multimodal outlook should be considered as technology progresses. This outlook requires redefining how we currently interact with VR systems and reconsider what interface we can use to support this multimodal outlook which will be discussed throughout this thesis.

1.4 VR Data Visualisation Review

Current information visualisation tools are predominantly lacking in utilizing VR or AR headsets, where a headset that does both can be referenced to as an extended reality (XR) or mixed reality device (MR). This is partially due to the usability issues still to overcome related to interaction and the relatively unexplored research territory of data visualisation within a digital spatial environment itself. Traditional 2D visualisation practices do not necessarily translate to a stereoscopic 3D environment; however, studies have attempted to bridge this gap by introducing new and creative perspectives for the purpose of data analysis. One such example has the user being immersed within a three-dimensional scatter plot, allowing for a 360-degree perspective of the data relative to other points in VR [10]. Further examples include digital twins of real world environments that allow users from around the world to experience a scanned object or location of interest digitally in 3D, such as historical castles, clothing and museum artifacts [3].

The opportunity to analyse data through an XR headset could support establishing new relationships between multidimensional datasets, whilst simultaneously improving user clarity and understanding within the data itself. VR also offers a natural interface for human-computer interaction when supported by gesture controls and immersive properties that do not exist in a traditional 2D desktop environment. Kinect systems used with VR have shown that a gesture-based system can decrease interactivity times by 27% when compared to traditional keyboard and mouse input mechanisms [11]. Challenges involved include the human ability to extract the information and create a basis for meaningful conclusions in the big data era [12]. Due to the overwhelming amount of data that can be presented to a user at one time, it's relevant to consider the human perceptual and cognition issues that can arise during data discovery.

The 3D spatial ability of users should be taken into consideration concerning this, and subsequent studies discussed. Understanding spatial imagery and recalling relationships with objects in 3D space requires representation in short-lived memory which resides in working memory [13]. This can place cognitive strain on users when visualizing the interaction of components if task-relevant perceptual information is no longer present. Such as architects viewing a building when the technical drawing is not available. Traditionally, utility companies working with underground pipework use 2D maps to locate and inspect underground work. Provocatively, literature on map definitions from the 1990s [14] suggest that the development of map visualisation should be challenged and progressed to higher dimensions of visualisation, by the very description of the map being a "graphical representation of the milieu." A testament to this, and the advancement of map visualisation include tools like Google Maps [15] and its use of processed photogrammetry scans for detailed 3D representation of buildings and landscape features such as hills and foliage. The Google Maps solution has also progressed into a standalone VR application allowing users to move in real time between digital twinned cities and photogrammetry assets. Due to the increased field of view and pixel density of such a system in VR, load times are however significantly longer than a typical 2D web browser experience.

For the period from 2013 to 2020, global data increased from 4.4 to 44 zettabytes [12]. This rapid expansion of data can be attributed to the growth in digital data sources, such as mobile hardware's ability to record and process more data associated with geographic information systems (GIS). Technologies supporting the recording of large digital twinned datasets such as LiDAR and high-resolution imagery to reconstruct are also a contributing factor. Much of this data is of a relatively new genre, with LiDAR scans and high framerate slow motion videography taking gigabytes of data a second. Cultural heritage sites are a prime example of a digital twinning scenario, which promises new modes of analysis and access to the scientific community worldwide. Physical access to sites of interest are no longer required for researchers to analyze an artifact or conceptualize an environment [16]. The associated metadata that comes with 3D scans of cultural heritage sites is to be considered as well. Clustering of metadata from datasets can enhance access to contextual information and its associated historical knowledge. Fast access to large datasets for Visualisation in a more immersive environment such as VR, could lead to the discovery of more relationships between cultures in a generous user experience.

A study discussed in "Visualizing Big Data with VR/AR" [12] discovered key design principles that should be taken into consideration in representing volumetric datasets with large amounts of metadata. It was described that serendipity can be influenced by the implementation of features such as narration or task-based analysis. Which in itself can engage the user in the data discovery process in a generous and pleasurable way. This was also reaffirmed in contrast to 2D Visualisation with VR [10] where user's productivity was subjectively rated by participants as higher, potentially due to the novelty factor of using a VR headset to explore data in a new medium. Applying a narrative to analysis has also been heavily explored in the data Visualisation community [17] with Visualisation tools such as Tableau [18] allowing for real-time data insights in a dashboard layout that dynamically updates temporal data. Facets of uncertainty is another indicator that data classification is an issue in the Visualisation space, to which VR could be a solution to the initial discovery during clustering without AI intervention by visualizing large datasets at one time. Examples of dataset clustering include text mining, shape similarity, image analysis for photogrammetry or computer vision with facial recognition, average colour abstraction, artist classification, and clustering manually by the author.

Research into advancing geospatial data Visualisation has stemmed from cognitive accessibility concerns. Within the civil engineering industry, it has resulted in new representations of volumetric data sets. "Advances in mobile augmented reality for interactive Visualisation of underground infrastructure" [19] attempts to develop a system to solve communication issues involving quantity surveyors and conveying ideas to those without formal map training. Augmented reality visualisation's that overlay 3D graphical content onto the view of the physical world was developed, which allow for the surveyor to create, edit, and update geospatial data representing real-world artifacts in real-time in an all-in-one solution. This provided immediate feedback to related parties and conveyed the information onsite at the time they required it.

Tadpole VR [20] attempts to display an anatomical model of the neural network from a tadpole spinal cord in VR. The end experience allows the user to observe the physiological model and the spatiotemporal interplay between neurons, axons, and their firing in real-time. A task-based analysis was conducted to support spontaneity by visualizing the data from a different perspective, presenting an opportunity to establish new data relationships, resulting in more questions about neuron activity. Whenever possible, researchers have used 3D visualisation tools to assist them in analysing complicated data [17].

Visualisation of neural networks are mostly completed on conventional 2D displays. A comparable example of such a tool is The Neuron Navigator [21]. The study used 3D neuron imagery in the form of large density maps from a fruit fly brain to understand how information is processed and transmitted along the neural pathways. The neural navigator was then developed to represent these volumetric neuron representations in 3D space to visualise their connective relationships. Similarly, a neuron tracing [22] application demonstrated that a visual representation of neural mapping in 3D would be more efficient for researchers to trace neural connections than plotting on numerous 2D maps. Allowing them to solve complex cases and understand the links faster. This study focuses on consumer-grade hardware where performance metrics do not appear to be discussed in detail, a common practice discovered during the literature review in interdisciplinary VR studies. The Tadpole VR system was developed with the graphical fidelity expectations of 75fps on an Oculus DK2 headset [20]. This approach allowed the study to render 1,500 neurons and 80,000 axons, a significant amount of data to visualise at one time but still a limitation as it is not a full representation of the dataset. The neuron navigator uses a voxel-based approach to segment neuron activity from a top level, allowing the user to search for and query specific brain regions.

It should be taken into consideration that many studies involving the development of a VR experience have been conducted with now legacy hardware such as the standalone Oculus Quest and the desktop-supported Oculus DK2. External system requirements and HMD specifications such as field of view, pixels-per-inch, refresh rate, resolution, graphics processor, and memory should be taken into consideration when evaluating end-user experience and will be further discussed with a detailed overview of HMD limitations found in Chapter 4.

The discussed hardware features give important indicators to the fidelity of the end-user experience and can contribute towards a positive or comfortable user experience. For example, it is recommended that the HMD frame rate stays consistent with the refresh rate to prevent motion sickness but also limits wasted frames rendered to not overuse system resources unnecessarily [23]. Therefore, the results of these user studies should be evaluated with the user in context to, *and* with the technical specification of the headset with associated hardware. Comfort, readability, interaction mechanisms, and immersion of the user’s virtual reality experience should be taken into consideration when analysing user studies. Without this, it opens the risk to bad user research data to iterate upon where users' subjective experiences may be impacted negatively through poorly optimised implementations not hitting performance metrics required for a pleasant experience.

Previous studies have used a variety of input mechanisms while wearing a VR headset. A study on immersive and collaborative data Visualisation [24] used a keyboard and mouse whereas a comparable study [25] used point-and-click with a mouse, gesture-based interaction, and controller-based direct manipulation for comparison. These different mechanisms of input can lead to varied user experiences during interdisciplinary studies. Where the user's input modalities are being explored, but not the user interface itself.

Initial pilot studies in 2014 [24] for VR data Visualisation research used off-the-shelf tools for data exploration, where "graphical rendering, geometry, and interactivity issues are already solved". The 3D platforms used in the study were second life and open simulator, both traditionally used to host online virtual worlds. Arguably, interactivity or interaction techniques were not an important focal point of the study if it is said that "interactivity issues are already solved", but the focus was placed instead on how users *infer* information from the Visualisation medium being presented.

A thorough analysis of visualizing data types in XR platforms have been completed however, identified issues relate to VR data representation [12]. Visualisation methods can be classified by data type, Visualisation technique, and interoperability. Each of these methods may require different interaction techniques based on the information dataset being presented, such as a 3D scatter graph compared to a photogrammetry asset. Navigating and interacting with these different datasets will require different interface approaches as well. Due to the increasing interest across disciplines in volumetric Visualisation from CT scans to photogrammetry and the use of vector fields / scalar fields, it is important to understand the limitations when displaying these datasets in a spatial VR environment.

Seven notable issues were identified when developing a VR Visualisation tool [12]. Interaction with the system should allow for the following functionality: enable data object scaling and navigating in the spatial environment to allow for a variety of perspectives. Selection of sub-spaces, objects, and groups of visual elements which enhance a subjective narrative throughout the analysis period. Views should allow the user to manipulate the location of data objects in 3D space enhancing comparative analysis. Planning routes of view and extracting the data analysed should also be a key feature. This would be interpreted as allowing the export and sharing of analysis results. The opportunity to share data relationships would likely be easier to demonstrate in the same environment it was discovered in, leading to collaborative methods. A multimodal control mechanism by gesture or voice control was also suggested to make it more intuitive for users.

Input mechanisms for interacting with the virtual environment were not explicitly discussed beyond gesture control [12]. It should instead be discussed, where input modalities change based, and consider the cognitive abilities required to complete a task and an interaction system that feels natural and intuitive for fast recall in unfamiliar interfaces. Input systems can also include eye tracking, utilizing user gaze, and facial gesture recognition [26] with BCI research not integrated into a commercial HMD yet. Hand tracking with a device like Leap Motion has limitations regarding haptic feedback as there are no haptics which is another key area of research to enhance immersive properties.

Haptic controllers such as the Valve Index do support natural touch-based interaction. They do however offer haptics through the use of the use of dual-stage triggers which can simulate initial touch and then increase haptic intensity with further interaction. On the contrary, removing controllers and using haptic gloves is a developing technology that bridges the gap between hand tracking and physical haptic feedback. 3D-printed prototypes of tracked gloves using drawcord systems to emulate touch or feedback have been developed. Haptic feedback in this scenario has proven to improve accuracy in interacting with volumetric medical images [27].

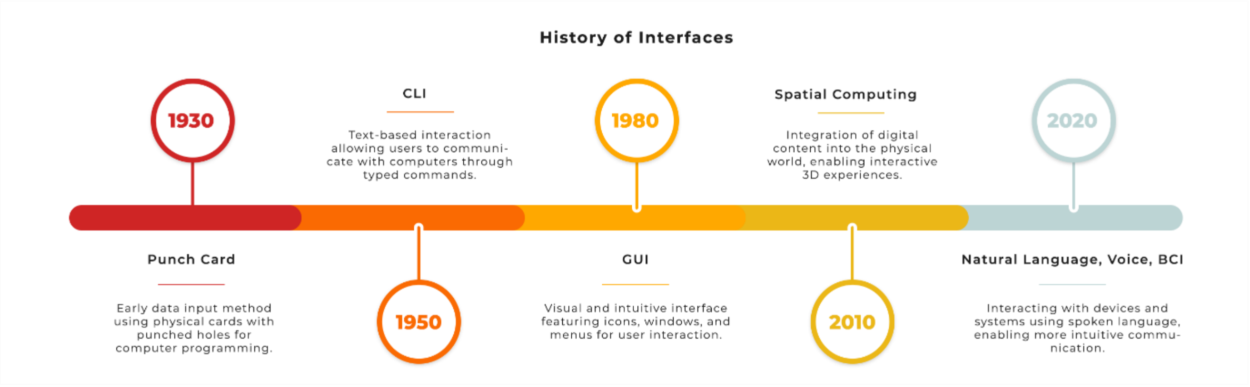


Figure 1.1 *History and current phase of UI development*

The advancement of brain-computer interface research has opened the discussion for its use with VR. OpenBCI provides EEG devices that developers can already begin to integrate and VR HMD vendors such as Valve patenting a device that attaches to the back of existing headsets, which are indications of its development stage. It is foreseen that there are still further levels of interaction capabilities that have not been fully developed beyond the 6-axis controller and tracking technologies. These BCI approaches are novel, and therefore limited to practical uses currently due to the signal processing time; however, it should be noted this is another form of interaction with a virtual space, which could enable accessible functionality to those with movement impairments. It's not clear how the integration of BCI can be used with existing interfaces beyond pattern recognition, but this thesis will discuss exploiting multimodal development approaches, allowing for innovations such as BCI to be considered at this design stage and not as an afterthought.

It is important to consider these upcoming HCI technologies when developing a framework for multimodal interaction. Figure 1.1 shows the history of the interface and the direction it is moving into the present and future. For interface development, it should be considered to focus on iterative development as it opens users to the interface at early stages for feedback, thereby improving support to the developed interface and improving adoption by providing a familiar feel for spatial VR experiences to users moving forward. Practical examples of this iterative UI development process for VR are discussed in Chapter 3.

Accessibility constraints found in interactive experiences may be exclusive to VR. An example is provided in a paper that describes mitigation strategies for those with visual impairments [28]. Users who can use a 2D tablet and have a range of motion to point and tap on the screen may have input issues related to a display that is fixed in digital space and can't be rotated freely. 3D Audio with distance perception is an example of removing boundaries using VR technology; Those with visual impairments can use 3D audio cues and play alongside other users who are receiving visual stimuli allowing for games and experiences to be played alongside able-visioned gamers.

The development of an accessible XR system should review existing cross-industry standards already in place highlighting common accessibility solutions. From this, consideration of the Visualisation medium itself should be made where an understanding of ergonomics in the working environment will provide information on how analysts currently work efficiently for sustained periods. An accessible first approach to designing VR interfaces could therefore reverse the traditional afterthought of accessibility implementations and could improve cognitive interaction with intuitive, dynamic interfaces from the start of development and support new sensor hardware – such as eye tracking and BCI – into the future with a familiar interface system. This would reduce the cognitive barrier to relearning interface elements as new sensor input modalities are made available.

VR applications often hamper able-bodied user's ability to interact with the virtual space as well however, and are therefore seen in general as a less accessible medium than traditional 2D desktop interfaces. A more inclusive approach from the start of the development cycle will benefit all users and prevent complex redesign for new sensor hardware later in the development cycle. Heuristic guidelines of human movement could be the start of a new usability evaluation for VR interaction. In the last five years, there have been increasing developments by researchers on VR head comfort range of movement [28][29]. This includes a heuristic evaluation of the comfort range for the user's head movement in horizontal head rotation and vertical head movement when looking at user interface positioning in VR. Further exploration into ergonomics can be found in Section 2.2.

The clarity of objects at the range has also been explored, Google's Distance-independent millimeter [30] allows for angular scaling of text. For example, if a text object is designed to be viewed from 2.5 meters away then scaling other objects to the same ratio of 2.5x should ensure readability at the desired distance. Note this does not take into consideration headset hardware but is a scaling ratio that can also be used in the real world as real-world objects are designed to be viewed from various distances such as street signs.

Olsen's framework [31] is designed for innovative interface design, in which a set of criteria for evaluating new UI systems is presented. Olsen's evaluation of interface systems was written near the birth of the coloured interface-driven operating systems in 2002. The desktop computer allowed design decisions to be consolidated with the creation of common interface workflows such as mouse pointers, context menus, drop menu systems, sliders, and buttons – sometimes known as windows, icons, menus, and pointers (WIMP). Relatedly, three assumptions in the criteria that previously had been outdated for over a decade could now come back into relevance with the extra dimension now available for XR interaction.

Under the section titled "Forces of change," Olsen's framework notes that the time criticality of dispatching an input event to the correct window location was no longer an important consideration due to advancements in system performance. Following this statement, an argument could be made that the assumptions that were said to be no longer relevant, could now once again be relevant regarding VR interface design, such as system performance. "10-million-pixel space is very different from interaction in a 250k-pixel space". This alludes to the idea that an increase in pixel space, such as a 360-degree, 6-axis motion HMD headset, may be a considerable enough leap in system requirements to explore optimisation requirements once again in interface design with such high resolutions being used in an HMD. High-resolution headsets require optimised memory management strategies to process high-resolution texture locations in 3D space from the back buffer on the GPU, but also when loading or streaming graphics from resource memory into positions in 3D space. Even more so, when considering interfaces in VR can now be 3D by design, this requires more resources than the 2D UI counterpart.

The relationship between photo-elicitation and volumetric environments in VR could impact our understanding of 3D environments with a user's emotion and their comments on a particular visual. Traditionally, using a 2D image to comment on, photo-elicitation is the process where an interview is conducted with participants responding to images displayed. They would describe the social, and personal meaning and values of the image. Meanings conveyed by the participant from the images supplement the verbal discussion in such a way that without images, different emotions may be described [32]. From this visual processing in the human brain is considered more developed from an evolutionary perspective in comparison to verbal processing of information. This again could be used as an argument that the next approach to understanding cultural and meaningful understanding with the world, VR could enhance understanding by using spatial 3D environments as a control method.

An experience *theme* is argued in the "Virtual Reality Applications" book as being a requirement for a VR experience. Also referred to as a metaphor, it is described as an introduction to the environment allowing the user to adjust [33] to the experience when it starts. The study “Interface Design and Usability for Immersive VR References” [25] uses empirical analysis of interfaces using three different types of interaction modes in the same application. Interestingly, trying to correlate a working environment that can be established in VR, it’s believed looking into traditional heuristics on desktop PCs using these types of arrangements [34] could provide a comfortable virtual workspace. Therefore, by taking a holistic view to representing data in VR we can potentially understand how the human operates in working memory and improve perceived performance when striving to understand multidimensional datasets.

Chapter 2

Accessible VR Interaction Model

Compared to conventional 2D interaction methods, virtual reality (VR) demonstrates an opportunity for unique interface and interaction design decisions. Currently, this poses a challenge when developing an accessible VR experience as existing interaction techniques may not be usable by all users [2]. Through direct observation with VR experiences and literature review, it was discovered that many traditional 2D interface interaction methods have been directly converted to work in a VR space with little alteration to the input mechanism, such as the use of a laser pointer designed to that of a traditional cursor. It is recognized that distance-independent millimeters can support designers in developing interfaces that scale in virtual worlds. Relevantly, Fitts' law [35] states that as distance increases, user movements are increasingly slower and performed less accurately. In this Chapter, it is proposed that the use of a low pass filter, to normalize user input noise, can alleviate fine motor requirements during ray-based interaction thereby accommodating for Fitts' law. Accessible first design methods are made, which are then exploited in further research in Chapter 3. A development study was conducted to understand the feasibility of implementing such a filter and explore its effects on end users’ experience. It demonstrates how an algorithm can provide an opportunity for a more accurate and consequently less frustrating experience by filtering and reducing involuntary hand tremors. Further discussion on existing VR design philosophies is also conducted, analyzing evidence that supports psychological models and multisensory feedback.

New technologies can result in more features; tooling; hardware; and capabilities which in turn can quickly turn into an amalgamation of complex design practices with conflicting goals during the development process. This results in critical evaluations for interaction left to the developer’s discretion without rigorous end-user testing. Opportunities created by technological progress can lead to excess choice during the implementation stage of the development process. This is reflected in the current state of the virtual reality (VR) market today with no defined set of accessibility design philosophies to follow. An intelligent reduction of design choices collectively can support a more efficient interaction system for all. For this reason, amongst others, it is crucial to consider an accessible-first approach that prioritizes users' ability. VR has great potential to enhance digital accessibility with the world health organization stating that 15% of the global population have some form of disability [36]. It is therefore important to address the potential challenges that may arise during users' interaction with VR devices.

Cognitive strain can arise when increased accuracy is required to complete a task, both mentally and physically. Fine motor skills are a requirement to interact with windows, icons, menus, and pointer (WIMP) based interfaces on traditional systems today using input devices such as a mouse or stylus, which themselves are prone to temporary accessibility problems. The same fine motor skills are required for 3D virtual environments using a controller with 6 degrees of freedom (DoF), however, there are no physical surfaces to support objects in free-floating space when using VR controllers. Whereas a mouse and keyboard have tangible feedback and supported by a surface. Moving one's perception to a 3D virtual environment for interaction with traditional 2D interfaces appears to be counterproductive as the significant increase in potential interaction space is not used. Rather than adapting the graphical user interface (GUI) many VR applications are instead adapting the input mechanism to work with the traditional 2D interfaces. This is due to issues related to precise selection and manipulation of objects in 3D environments [37]. In a comparable study, augmented reality hand-tracking input was compared with a controller laser pointer variant for manipulating 3D objects using 6-DoF. The result was on average a 39.4% faster task completion time compared to other AR manipulation techniques such as hand tracking where user hands are required to be elevated in the field of view from the Augmented Reality (AR) headset sensors. Notably, the resulting score of 8.51 on a fatigue test conducted using NASA's task load index (NASA-TLX) questionnaire showed a 15% average improvement over hand-tracking approaches [38].

Advancement in this field may require defying most traditional 2D interaction assumptions and creating new ones specifically for 3D spatial interaction. Canvas-based interfaces are not necessarily optimal for all VR displays. Previous mediums such as the desktop monitor require 2D canvas layouts to display GUI components due to the nature of the desktop monitor being 2D. However traditional GUI primitives such as buttons; sliders; dropdowns; context menus; are still almost directly being translated to VR experiences without any adjustment or modification to their 2D-designed counterparts. These systems have relied on a laser pointer to interact with the interface the same way a mouse uses a cursor for a 2D GUI. There are over 30 of these UI primitives in 2D UI design that once combined on a canvas make up most of the interfaces used today, such as sliders and buttons [31]. This has created an opportunity when migrating to technologies with new interfaces, allowing for elements to be redesigned for VR with an accessibility-first approach using lessons learned from 2D GUI mediums [39].

Prior to the adoption of the computer mouse, interfaces were command-line-based and interactable with a keyboard exclusively. Text-based adventure games require users to type or use the arrow keys to move around on the screen interface. Users were reluctant to switch from their familiar command line interface with DOS to a GUI-based pointer approach, but as graphics capability improved alongside the adoption of the mouse, interface design started to optimise tasks by developing these primitive UI tooling. Just like we tailor tools in the real world to complete a task for a specific environment, *we inevitably improve; automate, and substitute parts of its design with more technology because of progress.* Accessibility, therefore, is an important hurdle to overcome when developing new interaction techniques for extended reality (XR) devices. It is important that understanding new UI components is quick and supported by intuitive design while also considering alternative more accessible approaches to design.

The design environment with 360 degrees of freedom and 6-axis motion control in the spatial context of a head-mounted display (HMD) is considered a new medium for interface design with gesture-based interfaces. Where previously a 2D window with a single pointer was the design environment. Further exploration with 3D interfaces should be considered with an inclusive first approach. There are no comprehensive interface design frameworks for VR yet or direct translations of the top 30 UI primitives used today and how their equivalent should be implemented in XR. VR adds depth and 360 degrees of space to the environment, but we have thus far failed to agree upon the interaction standards that should be followed to make it inclusive for all. Guidelines are presented by Meta for accessibility considerations but don't necessarily provide solutions for interaction challenges [40]. 3D interaction can allow for a more natural interaction method to explore the virtual world as one might the real world. The physics that is anticipated in the real world when picking up an item can be accurately recreated in the virtual space to add immersive qualities such as gravity.

Advancements in passive input mediums for selection such as eye tracking allow for more interaction strategies to now be made available, Professor Stephen Hawking was able to interact with a digital interface designed specifically to his ability. In this, exploiting fine jaw movement allowed for confirmation of selection, and was able to achieve feats that of an able-bodied person. VR could further reduce the physical boundaries of users. An example of this is a preliminary study where users' Parkinson's tremors were reduced using a VR headset and tracked controllers [41]. Thereby supporting interaction with precision at a reduced cognitive strain [42].

Given that many systems have already been developed and released to public markets without necessarily providing an accessible form of input, what solutions can be provided to make these systems accessible *after* their release?

2.1 Core Considerations

Olsen's framework [31] was developed to evaluate innovative interaction systems and to understand if new interface interaction techniques have resulted in true progress. It notes that for a new feature to be considered in an interface, it should be evaluated if it adds a 100% increase to user satisfaction and improves their workflow using the new tool over the legacy interface element. This philosophy could lead to implementing numerous accessible features for a specific virtual reality experience, allowing the user to adjust their interaction preferences to match their specific needs. Challenges of implementing such a volume of accessibility options in an interaction system include feasibility and understanding what user physical constraints may be. A VR user with neck mobility restrictions may struggle to use a locomotion system where the virtual direction is bound to head movement. Using 6-axis controller-based locomotion to move in VR spaces often uses a joystick to move in the player's head direction. If users have restricted head movement it would lead to an "on-rails" experience, limited to one direction of travel or axis. An example of head movement being an advantage is the use of a non-diegetic fixed crosshair, bound to head movement for the selection of UI components. Allowing users who may have limited hand mobility an alternative means of navigation and selection if eye tracking is not supported by the HMD.

It is not fully explored in research the best locomotion technique to use due to the nuances that can be implemented into each approach. It is theorized in [43] that continuous VR locomotion induces motion sickness more than a teleport technique and was a key reason for the study teleportation in comparison with other movement approaches, such as grab and pull. The resulting study concluded that continuous locomotion techniques could be added as a supplementary method to teleportation. Teleportation locomotion will teleport users to another position in the environment, but due to the number of modifications that can be made to a locomotion system, further study is needed to evaluate the approaches demonstrated in Figure 2.1. Specifically, in how adding directional UI of the resulting teleport influences motion sickness or improves subjective movement efficiency in comparison to other techniques.

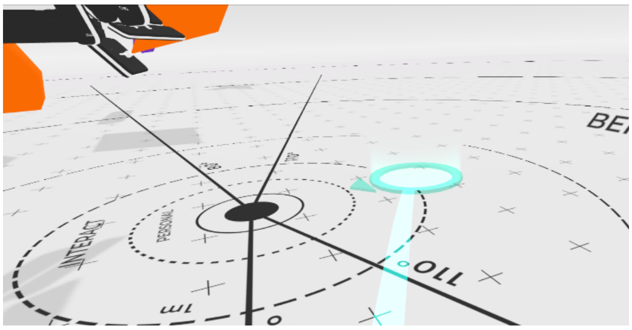


Figure 2.1 *Teleport locomotion with facing direction on release via joystick rotation*

With additional tracking hardware, full body tracking is possible, alleviating some locomotion nuances by allowing the user's hip direction to lead the direction of travel instead of the user's head. Thereby freeing the torso to rotate and allowing the user's neck movement to look around the environment if desired, which is more like natural real-world movement.

Supporting multiple locomotion variants allows users to pick what form of movement they wish to use to navigate. The authors in [44] discuss that designing for interdependence can facilitate communication with the user and those supporting or helping them during the experience if they have physical impairments.

In virtual social applications, supporting inclusive representation of users' avatars gives the power to the user to choose to embody characteristics of themselves such as the use of a wheelchair, prosthetics, or hearing aids [39]. Considering this, the user's height and width, avatar colour, and custom special effects should be reviewed to understand what virtual situations can induce the likes of seizures or increase neck strain from using heavy HMDs with various avatar heights. Standardization can provide an opportunity for users to still express and identify with an avatar but provide a less physically strenuous and inclusive medium to represent themselves. Such an example is allowing users to play a seated VR experience, but their avatar is still being represented as standing in social environments.

An examination of VR technologies in the context of the minority body concluded that the relationship between VR and disability is leading to a "body-centric technology that poses barriers for disabled users" [45]. The five points below are meant to spark conversation about some crucial topics to consider relating to VR accessibility. These themes could benefit from further discussion and inquiry in the context of VR, to help create a more accessible and egalitarian framework for improving all users' VR experiences.

2.1.1 Physiological and Physical Personas

Accessible computing still poses a challenge to overcome and a central focus on disability could prevent users from using their abilities. Interaction systems are designed and controlled methodologically by the developers therefore if the abilities of the current user do not match the assumptions made during development, it leads to an accessible gap. As described, there are many accessible solutions to leverage when creating an interactive experience which is why Oculus have introduced psychological modeling known as personas [40] into their accessibility documentation which describes techniques that the industry uses if there are limitations related to testing or accessibility to trial studies. The development of psychological models [35] is not a new concept for VR exclusively, it allows designers to gain reason and understand how the end user is perceiving the experience. Taking on the perspective of a user with an accessibility ailment and playing through scenarios in the application to understand what features can be introduced to enhance the end-user experience. By taking on user personas, this process would support identifying mechanics or interactions that prevent the user from advancing.

2.1.2 Dependency Reduction

To prevent overwhelming users with a plethora of new interface options for a given virtual environment, we can instead focus on the minimum skillset required to interact with the presented interface element. Users should be able to predict before interaction what they are going to do and how they can get their desired output just from the interface element. We can decrease the number of new mechanics the user must learn by introducing a *centralization of interface design* consistently, using similar UI elements such as buttons and sliders with predictable outcomes. With the goal to make operations easier, by focusing just on the core abilities the user already has from prior experience, rather than learning new skills. This would therefore deter cognitive overload and reduce the barrier to entry for novice VR users. To summarize, we can ask the following question to reduce dependency on previous experiences.

How can we reduce the amount of prior VR experience required, to allow a user to get to their desired outcome?

This shouldn't be taken as a contradiction to Section 2.1.3 below, where we can introduce more than one way to interact with elements. By effectively reducing the number of skill sets required for interaction, we can open other assistive approaches to interaction.

2.1.3 Availability of Input Methods

By taking inspiration from *established* colour-blind accessibility design such as the use of patterns and colour in the same material to support consistency in object purpose independent of colour. By supporting inductive thinking the user is more able to identify what will happen next should we interact with that object. Using similar design practices, we can support interaction techniques with interfaces, by using more than one way to interact with the interface presented.

Shortcomings of VR technologies include their barrier to entry. Exploring barriers to entries for VR, Leighton Evans [46] identified five barriers to the adoption of VR in higher education without accessibility as a central theme but touches on financial accessibility and the interactive expectations a VR application will provide. These were the materiality of the headset; interfaces; VR-induced sickness; hardware cost; and the language of VR.

A barrier to access includes the different forms of input, from interacting with a user interface to a 3D object. VR developers are often developing a mechanic to work with 6-axis motion controllers and mechanics will require serious development modification to work with another input device such as a joystick controller without 6-axis support. It was discovered that using a “multi-modality” approach allowing numerous input approaches, was a common solution to solving input-related accessibility issues [47].

Hardware locking input controls to a compatible controller may therefore be required to interact with an object in 3D space such as picking up an item from the floor. However, this constraint could still be mitigated by an alternative means of picking items up. Steam VR Input does allow for button remapping, where any controller button can be remapped. However, if an interaction requires a motion controller, then the developed mechanic may not work due to the requirement for positional tracking. It is in a scenario like this where the option to skip a task or mechanic that requires motion controllers would allow other aspects of the application to be experienced. Further discussion on this is found in Section 2.1.4.

Using hand tracking is an alternative means of interaction or selection without the need for motion controllers. By removing the requirement for controllers, we are reducing the requirement for learning physical button bindings but also limiting the possible interaction approaches and locomotion techniques beyond room-scale tracking. Using virtual joysticks or buttons in replacement of physical controllers can result in the loss of button remapping support.

The Meta Quest 2, 3 and Pro allow for hand tracking using inside-out cameras which detect hand movement when a user holds their hands within a specific field of view in front of the headset. Valve index headsets also have the option to attach the leap motion accessory to allow for hand tracking, again when the user’s hands are in front of them. Such features are not without their limitations. Tracking issues related to occlusion are common, and recently Meta [48] has developed a tracking solution that can track users’ fingers when they occlude each other. This does require baking hand movements to be processed by their developed algorithm, however, as technology advances - such as the inclusion of depth cameras - this may be an issue of the past and a solution is already available in the Quest 3 HMD. Tactile feedback is also a limitation of hand tracking without additional haptic gloves, therefore requires audio and visual feedback cues to show the user that their input was registered. While gesture-based input offers an alternative method of interaction, it can pose challenges for users with limited mobility or those who lack the necessary physical capabilities, such as having all fingers or the use of both hands. In such cases, the gesture-based system becomes a bottleneck, hindering the user's ability to effectively engage with the interface due to its specific requirements.

Notably with body tracking interfaces a clinical trial involving participants with Duchenne Muscular Dystrophy (DMD) [49] were tasked with interacting with interfaces with and without physical contact. Input mediums were therefore a touch screen interface, a Kinect sensor interface, and a leap motion interface. The transfer of skills learned from using the interfaces into a real environment was surprisingly effective for some tasks. Users with DMD "benefitted from the use of virtual technologies as compared to touch interfaces in acquisition and retention of performance during tasks [49]." Some further upper limb support due the being raised for an extended period. The implications of this were a reduction in task performance. Such results can show that different interfaces can support different demographics of accessibility.

As highlighted in [44] after long periods of use, keeping the user's hands in front of them can be fatiguing after a sustained period due to the tracking sensors being in front of them. To counter this issue, tracking of the user's hands must happen even if the user's hands are not in the limited field of view in front of the HMD. Implementing such a solution would require additional sensors, potentially on the headset's sides or room-scale tracking of users' head and hand movements. Such a feature would enable continuous tracking of the user's hands, even when their gaze is not directed towards them, ensuring seamless hand tracking support regardless of where the user's HMD is facing.

2.1.4 Forcing Mechanics

Positional requirements for interacting with the virtual environment should also be considered. Much like hardware locking, experiences can also have hard movement requirements. An example of this is where room-scale tracking locomotion is required to interact with a spatial UI element. This requires users to move in their real world space to get to the spatial UI in question. Comparably, requiring a user to crouch produces a similar problem. In this scenario, users may be forced to navigate into a physical position that is not optimal or comfortable for the user during interaction. If users cannot interact with an interface from any space in the virtual room, then the requirement for a room-scale experience is a hard requirement and therefore has many accessibility constraints associated with it. Having interactable components available at optimal ergonomic range and location for users could help alleviate issues that may be experienced with physically based anchored spatial menu systems. By forcing the location of world interactable elements (unless designed to be a one-to-one mapping of the real world) some users may find it difficult or impossible to interact with the component and would be classed as a restrictive interaction approach. Due to the nature of VR and the 360-space available, we instead have the opportunity to design modular interfaces and allow the user to adjust the interactable to where they desire or have *smart assistive features such as menu height adjustment* based on user proximity.

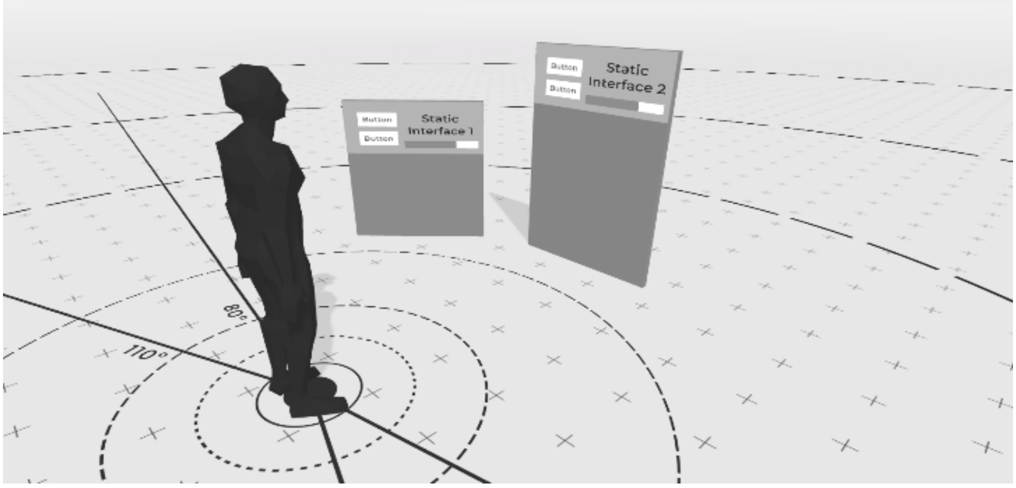


Figure 2.2 *Example of a fixed spatial interface resulting in user strain*

Figure 2.2 above is a representation of two static spatial UI menus where room-scale tracking is a requirement to interact. Such spatial UI menus can be found in many if not most VR experiences, Valves Half Life Alyx [50] and community worlds in VR Chat [51] frequently make use of static interfaces. The user would have to physically move in real-world space and adjust their height to interact with elements as intended. Even with laser pointers at range, the interface elements will likely be difficult to use or read due to the range from the default user position.

With much of the VR market tailored towards the gaming industry, there is an opportunity for those with physical limitations to experience another reality for a period and explore what otherwise would not be explored in the real world. Examples include digital reconstructions using photogrammetry scans of historical sites. Cognitive impairments can restrict a user's progression in traditional games with examples of low literacy or understanding of the task to be completed but interacting with 3D objects like the real world has the potential to remove the literacy barrier with intuition due to interacting with objects and physics just like the real world. An example of forceful mechanics is in Valve's Half-life Alyx, where users are presented with tasks such as holding onto a sphere with one hand and pointing to areas on the sphere with the other hand without colliding with moving obstacles. Similarly, the wire loop game was ported to VR and users had to pass this to unlock a feature or to advance to the next stage. Users could avoid some of these interactions, however, some are mandatory to progress through the experience. Those with wrist movement limitations or Ataxia would struggle to complete tasks and advance through the experience and therefore miss out on further content.

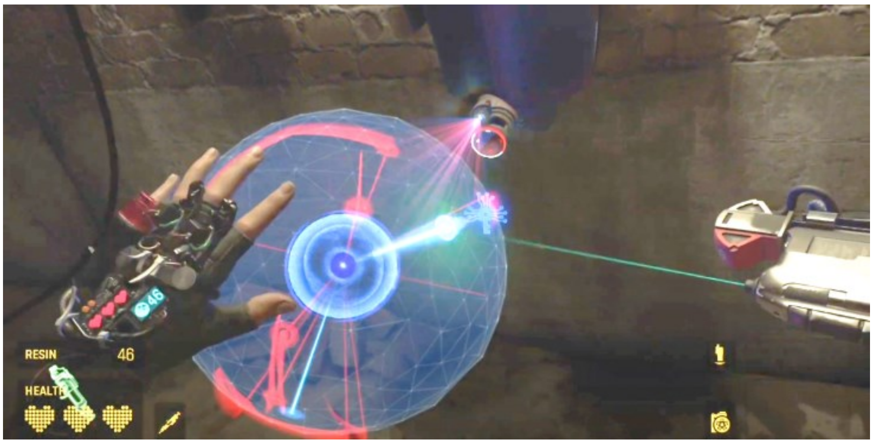


Figure 2.3 *Interactive scenario where users are required to use controllers* [50]

A solution to this is to prompt the user at the start of the experience if they want to enable an accessible mode, which allows for certain sections of experiences to be passed without negatively intruding or impacting their experience. Should the interaction be mandatory, creative approaches such as using alternative user-defined input mechanics could be made. Such as using the joystick to rotate a sphere shown in Figure 2.3, rather than rotating using shoulder and wrist movements. By using ability-based design as a central focus during the design process of interaction, the full human potential can be leveraged.

2.1.5 Text Focus and Rendering

Meta VR check requirements list objectives that encourage options for scaling UI elements. This allows users to adjust the reading size and contract according to their preferences. Considerations should be taken into the movements involved while reading text in VR including focusing user attention on subtitles during a 360-degree experience. Similarly, the source of audio or UI components may prove problematic during 360-degree exploration without additional guidance, such as reference frames. Spatial sound will provide an omnidirectional acoustic audio experience, which can assist in guiding the user's perception of the information being displayed in the environment. However, consideration is needed for those with hearing impairments who may not benefit from directional-based audio. For this reason, W3 [52] recommends that a mono audio sound option be available to users, similarly, Metas accessibility menu supports audio output from one side of the headset. Spatial arrows that point the user to where relevant information may be displayed is another reference frame approach using spatial UI elements. An alternative way to get user attention to subtitles or to view an area can also be accomplished by using the haptic rumble in their motion controller to direct them to an area of interest. This approach was used in [53] where visually impaired users have access to an advanced haptic system using a three-axis brake mechanism to simulate interaction with objects in virtual space.

For text interfaces to be friendly in VR, Oculus recommends a dyslexia-friendly font such as a sans-serif family with 35% kerning of character width and spacing between words should be roughly 3.5x the character width. Further study is planned in this area to understand how users perceive different font types in virtual reality, highlighting if readability practices outside VR coincide with text in VR. However, this would also need to consider text distance from the user in the virtual space and how this may influence readability. The British dyslexia association released a style guide in 2018 for best practices for all written communication both printed and screen media [54]. Character spacing is recommended at a value of 35% of the average width with word spacing also recommended at 3.5x the character width sharing similarities with Oculus recommendations.

Displaying text at angles can additionally cause strain and result in text being illegible when read at an angle. As text angles increase from the user's camera on the x-axis, as text tailors into the distance it becomes distorted and increasingly pixelated. This effect, also observed in 3D displays [55] showed that for 3D text angled at more than 60°, readability decreases. However, it was discovered that anisotropic filtering can improve text legibility, a factor to consider when applying anti-aliasing methods to text specifically [56]. To enforce text so that it is perpendicular to the user's camera, a billboard effect can be introduced to force text to always rotate and face the player's camera. Although potentially breaking immersive qualities due to the unusual effect compared to the real world, it will enforce text to always be facing perpendicular to the user's display. Another contributing factor to displaying text is the spatial mosaic of text strokes and their resolution size which was defined as "critical for legibility" in [57]. Furthermore, although not entirely avoidable for large passages of text, some characters were found to be easier to read than others. Where B, G, H, R, E, and N characters have a difficulty index of ≤ 1 and M, W, T, L, A, and I have difficulty index values above 1.13. With remaining characters ranging from 1.01 to 1.08.

2.1.6 Interface Scaling

Scaling of user interface components needs consideration in a VR space to ensure UI components are readable and elements are interactive at the intended distance. Canvases in VR allow for UI elements such as images, text, and iconography to be displayed much like a virtual screen. With numerous canvas sizes present and at a variety of ranges, consistent scaling of elements needs to be calculated to ensure content on virtual screens are usable for their intended purpose. This allows for content to be designed once and then scaled as appropriate for the distance it will be represented to the user. Font size for text at a distance can be calculated using distance-independent millimeters (DMM) to support legibility at their intended viewing distance. This term was coined by the Google development team as an investigation on how to appropriately scale items for visibility at various distances. Angular distances such as diameter, size, and apparent diameter or apparent sizes are used to understand object size at various distances. Angular distances are more commonly used in astronomy for celestial object size, however, can also be used to understand how large text or other objects will be displayed on objects such as billboards.

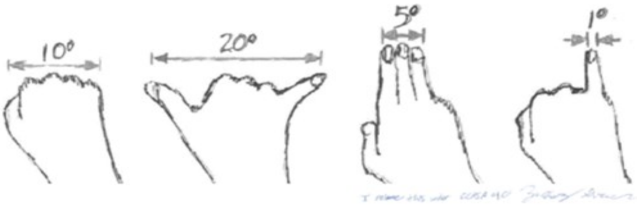


Figure 2.4 *Angular Diameter Approximation Visualisation in real-world space* [58]

DMM is described as one millimeter viewed one meter away, therefore scaling a unit of size to its distance. This allows for UI designs or objects to be created and then translated to the appropriate size based on the distance from the user's perspective, and camera. This approach would work well with traditional canvas-based rendering where ray casting is used to intersect the canvas and get user input based on the ray. Google has recommended that text be scaled based on its ray-casting requirements. 64 DMM is recommended for ray-based text scaling, where a user will be intersecting a ray for the text selection. This is considerably larger than bodies of text where 24 DMM is recommended with 16 DMM padding [30].

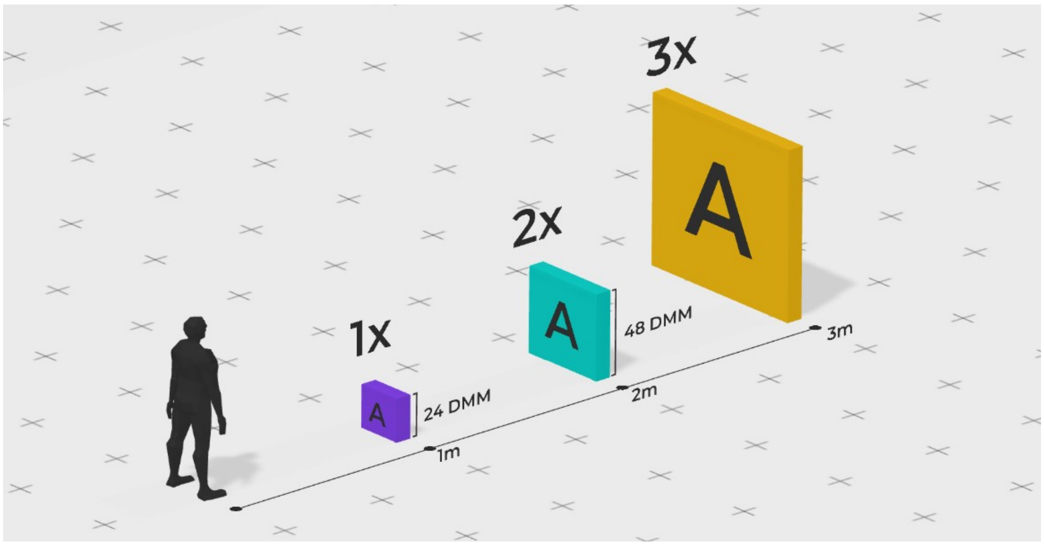


Figure 2.5 *Developed DMM Graphic demonstrating scaling for components at a distance*

Using this scaling approach, the development of any text in the interface can be run through a design process where text is sized at the dimensions desired by the interface designer and scaled up in the engine. An example would be if a sign with text on it is to be read at 2.5 meters away then the text needs to be scaled by 2.5 to match the design created by the interface designer where they expect the text to be read from.

Interaction with text at a distance will require fine motor skills due to the increased range, this is the reason for the increased DMM sizing for interactable objects over text. In the real world, a combination of senses is used to provide an actionable movement to our limbs. As vision is the sense used for distance, movements are increasingly slower and less accurately performed as distance increases [35]. This is summarized in Fitts' law. Taking Fitts' law into consideration alongside DMM, hit sizes for interactive elements are recommended to use 64 DMM as a minimum hit box scale and 96 DMM for a comfortable experience. To further enhance the interactive support offered and to deter user movement time from increasing as components are further away, a normalization algorithm could be used to improve the end user’s accuracy with elements at a distance. The below Equation 2.1 is the Fitts' law equation where user movement time is increased the further the target is.

$$movement\ time = a + b \cdot log^2 (\frac{distance}{size + 1})$$

Equation 2.1 *Fitts' Law*

2.2 3D Interface Ergonomics

As described in the previous section, areas of interaction beyond the user's arm's reach will require physical movement towards the anchored interface's location when using physical locomotion tracking. During prolonged periods this can not only result in fatigue but also prevent those with mobility restrictions from reaching the interface to interact. It is therefore important to consider constraints during development to support users’ ergonomic ability.

Considering user depth perception and headset field of view, locating the optimal spatial area for VR interaction with a motion controller can start to be conceptualized. Section 2.2.1 further explores work completed in this space with the development of a graphic to support VR researchers in future VR interface decision-making. Whereby overlaying all the figures, we can establish a baseline for where content should be placed in VR for a comfortable experience.

2.2.1 Accessible Rotation

To understand how head rotation can influence user comfort, Samsung conducted studies[29] that measured the range of comfort levels from users while turning their heads to the side while wearing a VR headset.

The formula below is using figures found in the study. It shows the subjective perception of comfort from the participant’s head rotation with 30° being a comfortable rotation range and 55° being the maximum before neck discomfort. However, every HMD has a different horizontal and vertical field of view (FOV) therefore it should be considered an estimate of user comfort range during head rotation [59]. The formulas below can therefore be used based on the comfort ranges on subjective comfort head ranges. Furthermore, the formula is used in a novel graphic to support developers design decisions in Figure 2.6.

$$Min\ Comfort\ Angle = \left(\frac{FoV}{2}\right) + 30$$

Equation 2.2 *Minimum Comfort Angle*

$$Max\ Comfort\ Angle = \left(\frac{FoV}{2}\right) + 55$$

Equation 2.3 *Maximum Comfort Angle*

2.2.2 Accessible Reach

Studies into stereoscopic perception have shown that 20 meters is the maximum distance humans perceive depth [60]. This also coincides with a study [29] where users placed in a VR HMD could see strong 3D from 1m to 10m ranges. With depth perception falling off between 10m to 20m, ultimately resulting in a 2D visual with no depth. Using this data, you can start to understand where 3D-rich information can be placed for a user and develop areas of the virtual space for specific user engagement.

The immediate area surrounding the user should never contain any interactable content as most of the space is taken up by the user's physical torso, leaving digital elements in this area could result in digital elements not being interactable as the user's hands collide with their real world body. At 0.5m users would likely feel discomfort due to the high potential for being cross-eyed so is classified as personal space. When taking into consideration that VR headsets also are relatively large on the head, over prolonged periods there can be further neck discomfort when looking directly down or up, therefore personal space should be left with no interaction.

However, HMD hardware focal distances can also provide insight into existing interaction areas. The Meta Quest 2 HMD has a fixed focal distance of 1.3m. Known as the vergence-accommodation conflict, users' eyes must therefore try accommodating the fixed focal distance by converging or diverging to a point in screen space. This design decision by Meta further supports the placement of objects at around the 1m range due to being in the optimal focal range for the user's hand movements. Until VR displays support varifocal lenses, we should continue to consider the clarity of objects in this range.

It is then suggested that the main hand interaction is between 0.5m to 1m from the user's center, where 3D depth perception is at its clearest, without cross-eye discomfort, and with existing HMD lenses supporting clear focus. Furthermore, a user's arm span is on average the same as a user's height. It would therefore be possible to scale this template algorithmically down or up accordingly based on user height while wearing the headset, however, there would need to be an additional study that takes comfortable user arm reach into consideration for a more accurate result. Arguably this zone is optimal for interactable components due to the comfortable legibility of presented text, with an evaluation of cockpit display interactions discussing values where it was said that the minimum comfortable viewing is at 35-50cm [61]. With supporting values in office ergonomics [34] monitors are about 70cm away from the user's eye gaze. Given the Valve Index HMD has a FoV of 107° and using the formula found in Equation 2.2 and 2.3, we are left with a minimum comfort angle of 82° and a maximum comfort angle of 107° which have been rounded up for the below Figure 2.6.

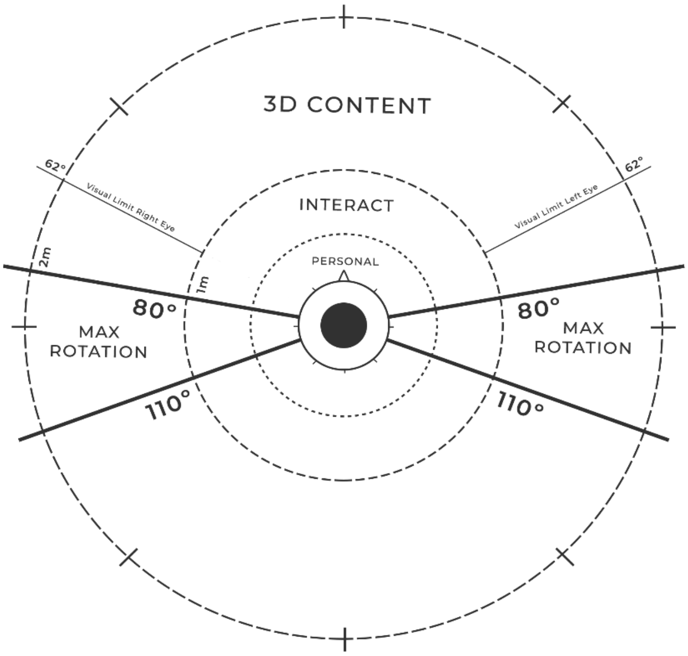


Figure 2.6 *Developed graphic to support developers in 3D UI placement*

The main 3D content zone is therefore between 1m and 2m as depth perception is still perceived and current fixed focal distance headset displays support this area. Objects placed in this area are too far away from the user to interact without moving in virtual or physical space.

The Graphic shown above in Figure 2.6 takes the methods described in this section and combines them into a single Visualisation that can be placed on the floor surface of the virtual environment being developed. Outlining the areas that developers should use for guidance when placing static objects in the scene, or dynamic objects where the user's gaze will be predetermined based on range. Figure 2.7 demonstrates a developed 3D GUI system using this graphic as a key part of its accessible first design.

Figure 2.7 *Developed Modular 3D GUI with spatial and diegetic interface elements*

2.3 Tremor Removal

The definition of a tremor from "Movement Disorders in Childhood" [62] paper states that a tremor is an oscillating, rhythmic movement about a fixed point that occurs when muscles contract alternately [63]. Traditionally managed through medical intervention via surgery or pharmacotherapy, recent research has however demonstrated the feasibility of tremor suppression using wearable technology and in a VR experience [64], requiring a non-invasive procedure for tremor reduction. In a previous premedical trial, 10 participants were studied to explore the opportunities available for involuntary hand tremor reduction using the Oculus Rift VR HMD [41]. The study in question found that tremors could be significantly reduced with the use of an algorithm to filter out input noise, enabling the users to complete tasks in VR that otherwise were impossible or difficult to complete, such as signing tax documents. Noticeably the study showed the tremors also showed a reduction in real-life tremors when the patient was using the virtual reality headset with an equalizer algorithm.

The jitter or tremor removal technique used in the above-described study was to introduce an equalizer algorithm, which was originally developed and designed specifically for normalizing communication channel data to combat the intersymbol interference effect [65]. Traditionally this would require previous data recordings to learn from and various algorithm input values adjusted for the optimal normalization result.

Without a prior dataset, however, it is difficult to define the frequency of a user tremor that can then be used to normalize the input frequency via a low-pass filter, it is perhaps then beneficial to describe various tremors known, to support targeting tremor frequency via a filtering algorithm. Present in every healthy person are action tremors known as physiologic tremors. Appearing more prominent during periods of excitement, fatigue, stress, or fear, the frequency of a physiologic tremor in young adults can range from 9 - 12 Hz, where it decreases to around 5-7 Hz in persons over 60 years [66]. Although may not be noticeable to the naked eye due to its low amplitude and high frequency. Further examples of action tremors include postural tremors, kinetic tremors, intention tremors, task-specific tremors, and isometric tremors. The essential tremor (ET) has previously been defined as a benign essential tremor. A resting tremor, however, is defined as a tremor where a person is at rest with no muscle engagement it is most common in hypokinetic syndromes – such as Parkinson's disease - ranging from 6 – 12Hz [67].

It's foreseen that integrating and allowing for adjustment of low pass filter values - such as the 1-Euro filter [68] - via a VR accessibility menu will allow users to define their desired tremor frequency cut-off value. It's important to note that the developed study in Section 2.5 doesn't target participants with tremors and should be treated as a sample size of the general population. We therefore aim to explore benign ET such as physiologic tremors found in every healthy person. Further exploration into using the 1-Euro filter targeting a specific tremor needs to be conducted, but it is hoped that this study provides an entry point into this area of study within VR interaction.

2.4 1-Euro Filter for VR

The removal of tremors via an algorithm can also have design implications when creating interaction mechanics as noisy signals are common when tracking human motion, therefore alleviating tremors may already be considered in the design of the interface. It is in human-computer interaction where the 1-Euro Filter algorithm can be used to assist interface selection whilst avoiding using a moving average solution that does not adjust based on movement speed. See Appendix A for an example implementation in C# for integration into the Unity Engine [69].

The 1-Euro filter is a low pass filter supporting a cut-off frequency allowing for filtered adjustment varying to movement speeds. At slow speeds, a low cut-off reduces jitter at the expense of latency, but at higher speeds, the cutoff is increased to reduce latency rather than jitter. The filter's official paper notes [68] that introducing time latency to signal processing naturally will reduce system responsiveness, although this isn't so important for artificial domains like decision-making, it can impact responsiveness and therefore perceived performance which has high usability implications.

In its foundation, the 1-Euro Filter was designed specifically for normalizing input related to interface interaction. The following study is a novel contribution, designed to explore how such an algorithm would be implemented for use in a VR environment and highlight the benefits of implementing such an algorithm using the Unity game engine, where this has not been studied prior and shows its potential in VR applications, from games through to quality-of-life improvements such as document signing for users with ET. After implementing the 1-

Euro filter, the application was then progressed into a user study using ray-based input in VR by adjusting user input noise during interaction with button and slider UI elements.

The 1-Euro filter normalization parameters are Frequency, Minimum Cutoff, Beta, and D-cutoff values, which can be adjusted in real-time to allow a tailored experience for the user at the desired frequency. This could open the opportunity for tremor reduction where the exact tremor Hz is not known, but the user can adjust the experience in real time and tailor it to their preference. The implementation of equalizer algorithms [41] for tremor reduction appears effective, however, the 1-Euro Filter is a more appropriate solution due to the low cut-off value allowing for reduced latency for fast movements. In addition, its simplicity in its implementation allows for fast integration by developers.

2.5 Implementation

An objective is to understand the feasibility of introducing a normalization algorithm into existing VR applications and potentially highlight issues associated with its implementation. Once complete, the next objective is to place users into a VR application and analyze their input times during UI interaction with the 1-Euro filter enabled and disabled respectively. The Unity Game engine was used with integration tests on a Valve Index HMD. The Unity XR plug-in framework was used to interface with the OpenXR standard which is now becoming a widely adopted standard supporting a one-build solution for compatibility with numerous HMD headsets. The hardware used for the development and running of the study meets the specifications required for the experience to run at 144hz at 144fps with hardware as follows; i9 9th generation CPU, 32GB of 3200Hz RAM; GTX 3070 GPU. This hardware is assumed to produce high graphical fidelity support for this study, introducing no jitters or unintended processing spikes for this developed experience.

The study first simulated vector noise on two 3D objects; A and B. Noise was simulated using a constant multiplied by the Random.Math() function. The returned value was then applied to the objects X and Y transforms. Additionally, the Z value during data recording was set to 0 as the objects had a cosine function applied to the controllers, simulating controlled movement across the X and Y axis where the Z axis was static. Note that normalization on a transform is different from normalizing a rotation value. For example, if normalizing the value 359, it may loop back to 0 causing a large jitter to be recorded, therefore conversion to Euler angles is required for rotational normalization using quaternions. The two-game objects were being manipulated with the same random vector values, where object A had the filter enabled at runtime and object B disabled. The output vector values for each object were recorded resulting in vector3 coordinates with their frame in the session. The movement and filter were applied in a FixedUpdate() method which restricts the frame rate to 50fps. This was to prevent fluctuations caused by various machine background processes and add a control element to the simulation. Both object's X position and Y position were stored for 500 frames or 10 seconds in a CSV file where Visualisation was completed using Tableau shown in Figure 2.8.

Implementation of the 1-Euro filter algorithm was forked from [70] utility with further modifications to support the VR environment. As described, the filter uses low resources and works by completing an initial first-order low pass with a frequency cutoff. During low movement speeds, the cutoff value stabilizes the signal by *reducing jitter* and as the speed increases the cut-off value is increased to *reduce lag* on the applied object.

Issues associated with the implementation were related to design decisions of integrating the algorithm with a VR controller for the user study. The decision to either directly filter the user's controller positions or filter an object set to a child of the controllers. As VR controllers can have 6-DoF, storing the user's desired position in a buffer and then applying the filtering algorithm can lead to increased latency, furthermore, filtering the user's hand or controller movement directly will cause the user tracking to not feel responsive and can therefore cause frustration. Not having 1:1 tracking can also contribute to breaking immersion. The solution was to implement the filter onto the pointer which was set as a child in the hierarchy to the tracked controller. The functionality of the filter is then limited to interaction systems on this object, such as the user's laser pointer. This allows the user's hand tracking to be 1:1 accurate and representative of real-world positional tracking, while the laser pointer and interaction system has the smoothing filter applied.

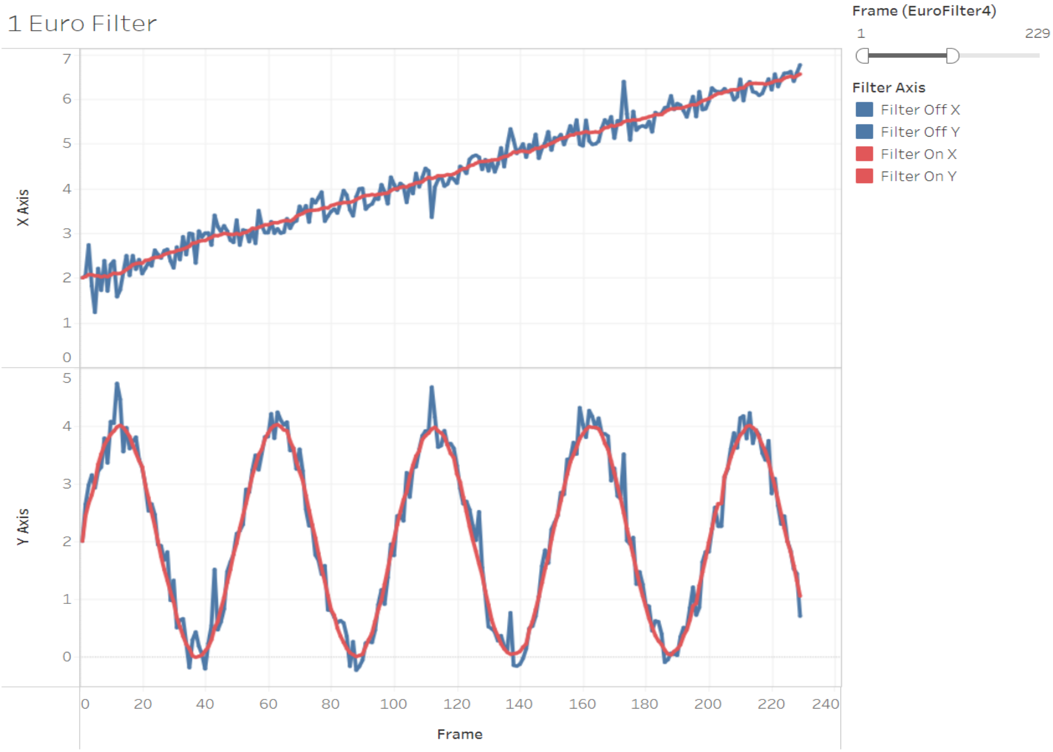


Figure 2.8 *Experimental results using noise generation and with filtered values*

Values for the 1-Euro filter were adjusted in real-time to experimentally explore what values are relevant for reduction using a ray pointer, taking into consideration ET at the frequency range of 9-12Hz. The study was completed using the following filter parameters: Frequency was set at 120, the minimum cutoff value was 1, the Beta coefficient was 0, and the delta cutoff was also set at 1.

2.6 Evaluation Procedure

A fully supervised in-person study was conducted with 36 participants aged 18-55 who were recruited from the public, faculty, and students. These participants identified having no movement disabilities and as far as they were aware free from neuromuscular disorders. Data from 6 participants were disregarded due to being unable to complete the tasks due to being unable to wear their glasses with the headset on and noted they have eye stigmatisms (N=4) with the remainder of participants (N=2) being unable to complete the tasks due to unable to see the interface elements clearly at various ranges.

Two UI elements were selected to be tested with 30 users, using a laser pointer input from the Unity XR framework. A 2D numpad with values 0-9 and a 2D slider with input from 0-100 were implemented into a Unity scene created from the previous simulation discussed in Section 2.5 as the filters proof of concept was already validated.

As tremors increase during fine motor requirements and with Fitts' law being taken into consideration discussed in Section 2.1.6, users would be presented with these generated UI elements for interaction at increased ranges of 1m, 5m, and 10m. Users are tasked with inputting 4 randomly generated integers into the numpad at each range in their own time after pressing the start button. Once they have correctly entered the value, they will return to the now stop button to stop the timer. The same process is carried out for the slider; however, the generated value was between 20-100 to prevent users from clustering input at lower values of the slider, where its handle is initialized at zero. If the number 5 was generated the slider anchor only needs to be moved 5cm. The slider is 1m in length – left to right width - and represents 1cm per input value throughout the tests at different ranges, the numpad sizing is also the same throughout to focus control measures on the efficacy of the filter.

At first, the participant was introduced to the environment after adjusting the headset to a comfortable position and appropriate IPD value. Mistakes made by the user during integer selection using the numpad were rectifiable by the user using a backspace button. Users were shown the slider and numpad at 1m initially and were able to practice input once in their own time before the study started.

To control task order effects, a counterbalance is used in which participants have the filter on or off at the start of their task, randomly assigned 50% of the time. Only when each task at the 3 ranges was completed with the numpad and slider, they would complete the same tasks again but with the filters toggle value inverted.

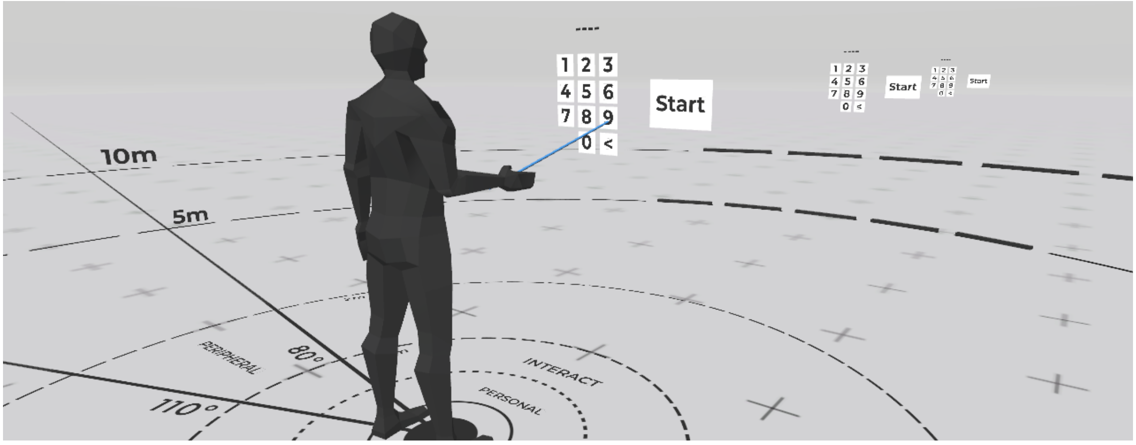


Figure 2.9 *Users test environment with the numpad and slider display at various ranges*

2.6.1 Results

In the following study, we investigated the null hypothesis that task completion times show no improvement with the 1-Euro filter enabled. A paired t-test was conducted to assess the null hypothesis that there is no significant difference in task completion times with the filter enabled, only where the p-value is less than the significance level which was set at $\alpha = 0.05$.

2.6.2 Slider

Results indicate that at 10m ranged slider tasks there was a significant improvement in task completion time with the filter enabled t(29)= 3.1750, p=0.0035. Therefore, we reject the null hypothesis that task completion times show no improvement with the filter enabled and find task completion time was on average 10.57 seconds faster with the filter enabled at 10m ranged tasks. This correlates to an average of 42.83% improvement in task completion time for users during 10m slider tasks, as shown in Table 2.2.

However we found minimal difference in completion time for 5m ranged tasks showing t(29)= 1.8518, p=0.0742 where we therefore accept the null hypothesis. This resulted in an average completion time of only 4.11 seconds faster with the filter enabled.

Similarly at 1m ranged tasks, we find no significance in completion time t(29)= 0.7833, p=0.4397 where we again accept the null hypothesis with the average task completion time only 0.55 seconds faster.

	1m	**5m**	**10m**
p-value	0.4397	0.0742	0.0035

Table 2.1 *Slider task completion p-value at given ranges, with a significance level at 0.05*

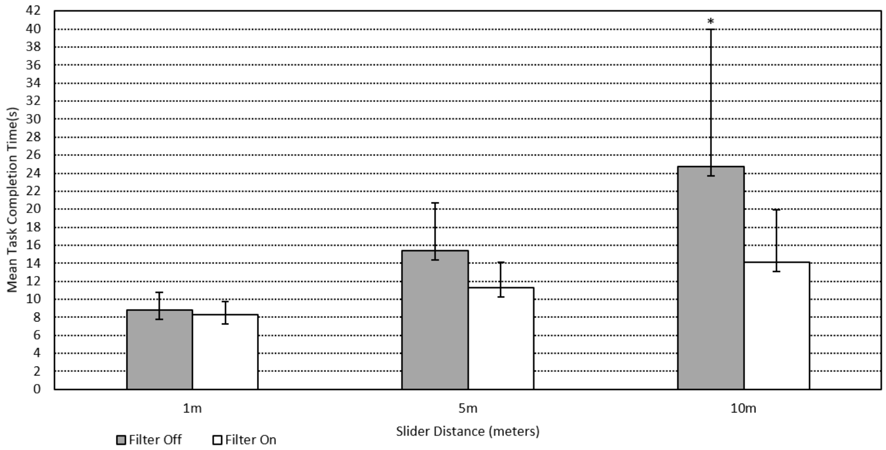


Figure 2.10 *Slider task completion mean and standard deviation*

	Filter Off	Filter On	Difference	Reduction (%)
1m	8.806	8.254	0.552	6.27%
5m	15.387	11.271	4.116	26.76%
10m	24.700	14.125	10.573	42.80%

Table 2.2 *Mean slider completion times with mean difference and mean % difference*

2.6.3 Button

Results for button selection-based tasks from a virtual numpad indicate that 10m ranged tasks showed highly significant improvements to task completion time with the filter enabled t(29)= 4.5569, $p < 0.01$. Therefore, we strongly reject the null hypothesis and find task completion time was on average 9.58 seconds faster with the filter enabled. Which was on average a 44.59% improvement in task completion time for users during 10m button tasks.

We again find significant improvement to 5m ranged tasks showing t(29)= 2.1426, p=0.0406 where we therefore reject the null hypothesis. This resulted in an average completion time of 1.72 seconds faster with the filter enabled.

However we found minimal difference in completion time for 1m ranged tasks, showing t(29)= 1.0608, p=0.2975 where we therefore accept the null hypothesis. 1m Ranged tasks resulted in an average completion time of only 0.434 seconds faster with the filter enabled.

	1m	5m	10m
p-value	0.2975	0. 0406	0.00008

Table 2.3 *Button task completion p-value at given ranges with a significance level of 0.05*

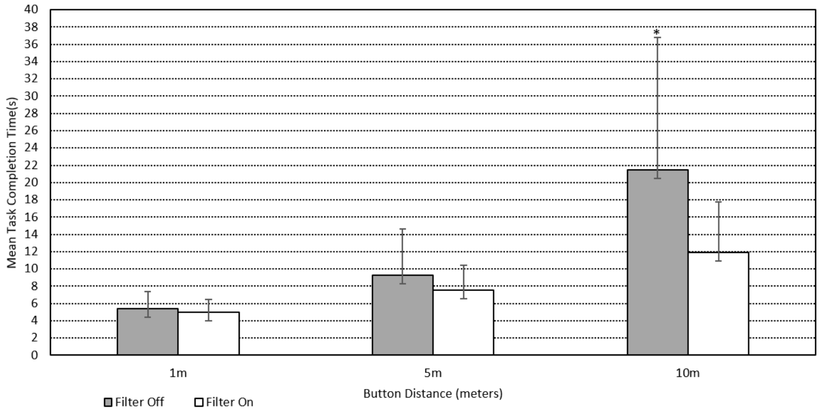


Figure 2.11 *Button task completion mean and standard deviation*

	Filter Off	Filter On	Difference	Reduction (%)
1m	5.413	4.978	0.434	8.02%
5m	9.242	7.523	1.719	18.60%
10m	21.507	11.924	9.583	44.56%

Table 2.4 *Mean button completion times with mean difference and mean % difference*

2.6.4 Discussion

With the filter enabled, metrics trended towards a reduction in task completion time from 1m to 10m range tasks and on average showed a reduction in overall task completion time with the filter enabled. Showing minor fluctuations in performances at close ranges - 1m to 5m - during both slider and button input. The mean time differences between 1m and 5m with the filter enabled showed minor average task completion times, however, it is still important to discuss the benefits it had on some of the population, just because a small population doesn't exhibit statistically significant improvement in some tests doesn't mean there aren't significant advantages for a demographic.

During interactions with the filter enabled, we observe a reduction in variance among participants which is correlated with the improvement in task completion time. It is noteworthy that while there was an average improvement in task completion time, the presence of the filter ensured that task completion never exceeded the maximum observed during interactions with the filter disabled.

The correlation between the timing of the slider and button tasks was unexpected, as we see similar average task completion time improvements despite utilizing different interface elements. The slider showed average times trending very similar to the button's average completion times, in which the 10m ranged tasks 42.83% was the average improvement with the filter enabled for the slider and 44.59% improvement for the button. It should be considered however that the button interface required 4 integers to be selected over the sliders 2. This unanticipated finding raises interesting questions regarding the effectiveness of ray-based interaction for button input over a slider counterpart. It prompts the exploration into whether using a slider for text input as opposed to traditional buttons could potentially enhance the speed of text input – a known constraint in VR interfaces.

As discussed in Section 2.5, the implementation of such a filter is low on system resources and can be enabled or disabled in real-time with a toggle at the user's discretion. Users who may not have increased times with the filter enabled could disable the filter in real-time or adjust the values to suit. Further study could be completed on the filters used with 3D GUI interfaces and how interaction could be eased with the use of ray-tracking eye focal points.

2.6.5 Conclusion

In this chapter, we have discussed techniques and psychological models to consider when developing an accessible virtual reality application. Ensuring we do not take for granted that existing 2D design paradigms will directly translate to VR, and instead provides us an opportunity for new solutions to be designed specifically for VR interaction. 2D WIMP Interfaces that are currently adopted in VR frequently introduce laser pointers to interact with 2D forms and menu systems containing buttons, sliders, and context windows. Due to the fine motor skills required for selection - at range specifically as discussed in Fitts' law – it can be frustrating, tedious, and an inaccessible design decision for many.

Our solution is normalizing ray-based user input using the 1-Euro filter algorithm, demonstrating how existing VR applications can be made more accessible by an inexpensive low pass filter, reducing fine motor requirements seeking to facilitate VR interaction of typical users. Our evidence supports significant improvement in task completion time specifically at 5m to 10m ranges. Further evidence from previous studies has also suggested Parkinson's disease involuntary hand tremors can be asymptomatic in VR with the use of a normalization algorithm [41] and our study demonstrates task completion times can be reduced for typical users without movement disabilities. With this, it is now hoped that further exploration can be given to those with persistent tremors as found in diseases such as Parkinson's. We propose such a filter being introduced into existing VR headset's accessibility menus via the toggle, providing support to users who may greatly benefit from interaction that relies on precise fine motor skills.

It should therefore be acknowledged, that there are benefits of ray-based interaction at range and how this can be an assistive input mechanism, and support users who otherwise may not be able to interact with spatial UI elements at range or with an alternative gesture control scheme. The likes of ray-based interaction mechanisms are important for eye-tracked input.

The use of a laser pointer for interacting with 2D UI elements in VR space could be considered as an interim solution for most VR interactions as many UI elements are similar to that of traditional UI and don’t take advantage of 3D space. By combining the accessibility approaches discussed, an outline of key considerations has been discovered to support developing an ability-first design approach. Further research has led to the exploration of an accessible first design framework for the creation of a novel VR interaction interface, presented in the proceeding chapter - the development of the Modular 3D GUI. It is aspired that the use of normalization algorithms can benefit users in care settings by mitigating their tremors in a virtual space. This, in turn, reduces barriers to accessing content and provides them with an opportunity to find relief in exploring digital worlds.

Chapter 2 established a significant correlation after study, between the increase in task distance and the increase in task completion time for both slider and button-based tasks. This aligns with Fitts' law. This finding suggests that ray-based interaction, commonly used in VR interfaces, may not be optimally designed for tasks that require completion from a distance relative to the user's fixed position. Chapter 3 will build on this insight by critically examining the input mechanisms currently employed in VR applications and games. It will explore how these mechanisms can be refined to support future modalities, enhancing user experience without necessarily discarding the ray-based approach, especially considering its potential benefits for accessibility. This chapter aims to question existing paradigms and propose directions for more inclusive and effective VR interfaces by introducing a novel contribution of this thesis, the creation of the Modular 3D Interface.

Chapter 3

Modular 3D Interface Design

Designed with an accessible first design approach, the presented chapter describes how exploiting humans' proprioception ability in 3D space can result in a more natural interaction experience when using a 3D graphical user interface in a virtual environment. The modularity of the designed interface empowers the user to decide where they want to place interface elements in 3D space allowing for a highly customizable experience, both in the context of the player and the virtual space. Drawing inspiration from today's tangible interfaces used, such as those in aircraft cockpits, a modular interface is presented taking advantage of our natural understanding of interacting with 3D objects and exploiting capabilities that otherwise have not been used in 2D interaction. Additionally, the designed interface supports multimodal input mechanisms which also demonstrates the opportunity for the design to cross over to augmented reality applications. A focus group study was completed to better understand the usability and constraints of the designed 3D GUI [1].

The progression of the graphical user interface (GUI) has reached a level of ubiquitous understanding, allowing users who are accustomed to desktop or mobile device interfaces to interact through familiar interface elements such as windows, icons, menus, and pointers (WIMP).

However, with the adoption of virtual reality, these design decisions have for the most part been directly translated to work in a 360-degree experience supporting, 6 degrees of freedom (6-DoF). Resulting in the same 2D elements being used in virtual reality (VR) with user selection primarily being a ray cast in the form of a laser. This allows designers to use existing design concepts by similarly placing elements to a traditional desktop environment.

If we depend just on conventional interface design choices, we are not necessarily taking advantage of the extra dimension made available. Existing interaction techniques for VR have resulted in a novel and diverse set of designs that have primarily been developed for the VR games market [6].

3D interfaces provide the opportunity to utilize our sense of proprioception as we physically surround ourselves with static interfaces in the form of 3D objects akin to that of the real world. Proprioception can also be referred to as kinesthesia and is the ability to know where your body is in space and has been referred to as our "sixth sense" [71]. It gives the user an idea of where their body parts are in relation to their environment and how they may be able to interact with it. Drawing closer to incorporating the human body fully can increase our understanding of the 3D digital environment presented, by using our natural understanding of the world [72].

As VR has the potential to provide more presence than a traditional interface, the idea requires more interface design evolution to become more flexible and human oriented. An example study simulated wind; the breeze enhanced presence among users [73]. Showing that somewhat unconventional methods can enhance presence, to which, natural interaction for the user may furthermore provide fewer constraints leading to increased presence. The developed interface outlined in this chapter provides a multimodal approach to interaction design and fully supports existing ray-based input controls. Thereby reducing the adoption effort required by users who may have accessibility constraints, requiring them to use ray-based interaction. Using a new medium of fully 3D panels over traditional 2D windows or forms, the system in addition to supporting novel 3D interaction also allows for the migration of ray-based interaction techniques typical for existing VR applications. Gesture-based interaction approaches have been used for input, allowing the users' hands to be tracked with or without a controller during selection and navigation. The system is modular allowing users to move the location of all UI components in 3D space to suit their preference, with the addition of contextual-based UI being explored where UI elements appear only when relevant to the task being completed.

Predicting what an interactive element will achieve is largely responsible for its placement, context, and iconography. A skeuomorph is a design concept that is used to represent objects with their real-world counterparts, primarily used in user interface design - such as associating a floppy disk with a save icon. In Ivan Sutherlands' famous essay "The Ultimate Display" [74] he opens with "We sense an involvement with this physical world which gives us the ability to predict its properties well. For example, we can predict where objects will fall, how well-known shapes look from other angles, and how much force is required to push objects against friction." Using the real world as a guide for the GUI we can cross over our ability for pattern recognition in context using inductive thinking, with the potential to enhance memory by association with their real-world counterparts. An important aspect to consider before exploring the opportunities made available by a 3D GUI was the vehicle in which to showcase the interface itself. Usability discussions were completed in focus groups which allowed participants to share their experience using the developed interface alongside peers and provide insight into the satisfying and frustrating aspects of the interaction. Establishing a familiar metaphor was important for this reason, as recruitment targeted users' previous exposure to VR applications, where novice to intermediate users were selected to participate. The selected demographic was used, as the study's focus was to understand the usability of the designed interface and highlight issues and struggles with a 3D modular GUI without significant context to current design philosophies.

A conceptual model was therefore established to understand the problem space for a new medium of interaction. To present the interface in an application that a 3D GUI may apply to, the development of a 3D Visualisation platform for geospatial positions was developed. Allowing for Visualisation of various forms of 3D spatial data including point cloud scans, video playback, and geospatial data which was represented by electric vehicle (EV) chargers' latitude and longitude positions. Querying of EV charger locations was completed based on their charging type and availability. This allowed for a selection of chargers to be scanned and linked to their real-world position visualised by their point cloud scan using the Microsoft Azure Kinect. The map was selected as the primary Visualisation tool as an homage to the map being the "graphic representation of the milieu" [14] but primarily due to it being a familiar Visualisation medium for many people.

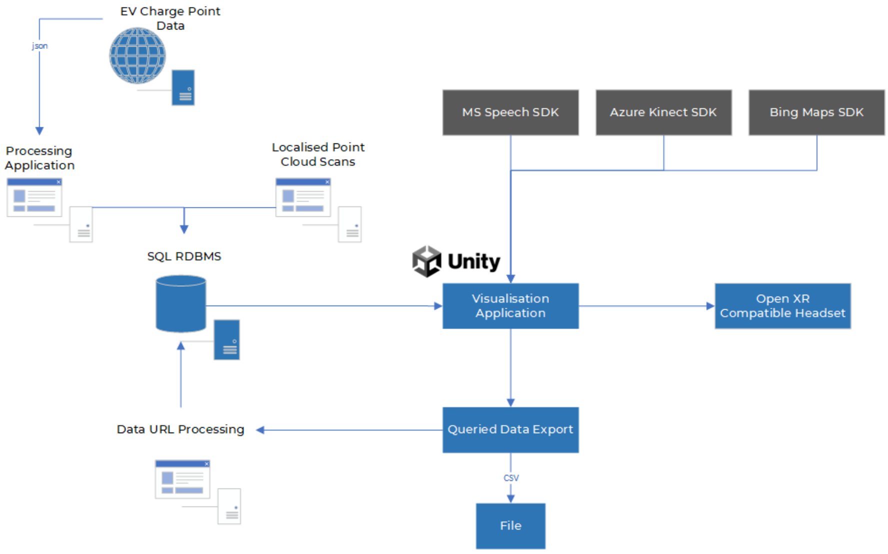


Figure 3.1 *Dataflow diagram of the developed VR querying system*

3.1 Modular Interface Concepts

The definition of modular by the Oxford Dictionary states, "employing or involving a module or modules as the basis of design or construction." In programming design practices, modular code is to divide the functionality of the system into separate modules that are independent of one another. The concept also stretches into UI design where interface elements are created for a variety of environments. An example of this is during website development, where interface elements are isolated with their own set of constraints, allowing them to be independent and dynamically adjust based on the user's device. This approach to a responsive layout allows for support across various platforms; mobile, desktop, tablet, and even VR. In contrast, defining constraints for 3D UI in immersive environments have been explored [72] but have also yet to be fully integrated into real-world applications. UI Elements without any constraint are problematic as the increased design space leaves many design choices at the discretion of the developer. The key areas in the presented 3D interface include the following and will be discussed further.

- Free-floating allows for adjustment in 6-DoF for users with specific viewing angle requirements and allows for anchored placement in 3D space.
- Contextual, allowing elements to only be displayed when relevant to the location in the environment resulting in;
- Predictable task-focused interactions, using skeuomorphs and colours around 3D interface trims in designated context zones.

Key to the naming conventions definition, it aims to provide the "basis of design or construction." The presented 3D interface allows users to move the UI to a location in 3D space, that isn't dictated by the application being presented. Rather than employing rigorous design methodologies to focus on the layout of interactable element positions such as the location of buttons and sliders, we can instead empower the user to place interactive elements best suited to their needs.

By interacting with VR interface elements repeatedly, the user can learn the spatial location of objects and understand the interaction mechanism employed by them [75]. As a user completes their desired task, they can adjust the placement of immediate controls in location to suit their preference. This allows users who may have accessibility constraints to adjust the UI to a location that is suitable for their interaction.

***Figure 3.2** 3D GUI Map Visualisation*

The concept of consistently visible UI controls stems from traditional fixed UI elements. An example is the toolbar on most WIMP interfaces where the close, expand, and minimize buttons are always available to the user. The developed interface has included this fixed UI concept through an iterative development process, ensuring users can show and hide 3D panels in a single action. The current interface solution is part of an iterative development process, incorporating feedback from users through focus groups as discussed in Section 3.4. This feedback is critical to focus on when aspiring to enhance and improve upon its design.

Panels containing UI elements such as buttons can be moved in 3D space with no gravity influencing them, resulting in panels being *free-floating* with collision properties for user interaction. The user interacting to move the placement of panels uses natural gestures such as grabbing and releasing. These panels can be fixed to the user's virtual body for movement around the environment but also when not fixed to the user's movement (controlled via button press) they are instead anchored in place in relation to the desired task being carried out in the environment. The completed study in Chapter 2 also shows results that task completion time is improved when interface elements are close to the user. A paper exploring new directions and perspectives of 3D GUI [75] notes that "standard 3D UI techniques can and should be based on object attachment" when regarding interfaces floating in space. In addition, the author describes this as an exception, not a rule when relating real-world interaction to virtual interaction. The designed interface presented in this chapter includes panels that are free-floating based on the user's exposure to existing interfaces in the real world that may have free-floating connotations. Primarily being, the computer tablet. Although the paper's release was in 2008 [75], many users would not have been familiar with the concept of the digital tablet as the original iPad was released in 2010, two years after the initial paper's release.

The panel is a key component to the design of the modular interface, allowing for relocation in free-floating space and anchored to user movement or left static in a location desired by the user. This is consistent across all panel and UI elements, keeping interaction between elements predictable. Using panel colour and shape to direct the user to these interfaces with predictable interaction areas. This was created by adding trim around all panels in a colour that can be selected by the user. In Figure 3.3, the trim is white around the grey panel, with various transparency properties on the back panel to provide contrast to the surrounding environment.

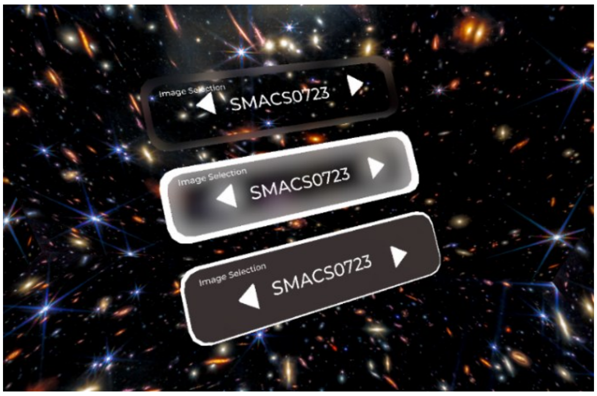


Figure 3.3 *Three design examples of the Modular 3D UI Panels*

The key to natural interaction is the use of gestures and physical movement to move panels, to support accessibility, it is intended that the interface can be customized via a configuration file that has values of key elements pre-placed in the environment allowing them to interact with the interface. For example, if a user is unable to stand and lie down, they may not be able to reach for the interface components at first, instead, another user or a configuration file can be used to allow the user to interact with the elements from their position. Further exploration into the availability of interaction techniques is discussed in Section 3.3.

The redesign of the traditional 2D slider into its 3D counterpart was centered around ensuring it doesn't require a complex interaction, being a variation of small and large movements in a single action. Instead, the slider allows for scrolling via small movements at a fixed distance, not requiring movement across the entirety of the slider. Prioritizing shoulder and elbow movement over wrist movement. Due to the spring physics associated with the anchor point in the center, the slider's anchor always returns to the center position upon release. Scroll speed scales as it is moved further from the origin. This design is like that of the middle mouse click auto-scroll on existing 2D scroll-bar elements.

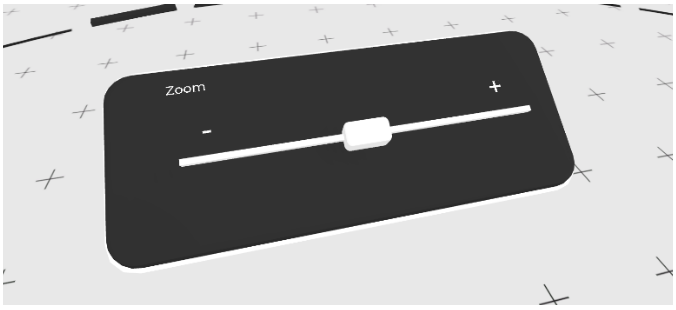


Figure 3.4 *Reimagined 3D Slider using Physic Spring Joint*

With the addition of the third dimension being added to interface design, there becomes more opportunity for additional confusion in establishing a mental model of the system for end users, due to the context space in which the interface may be presented. With more freedom of space, there becomes an increased availability for the placement of GUI components. Existing 3D applications such as CAD software on 2D interfaces are designed for professionals and as such, provide a plethora of options in tabs and sub-menus. With the extra dimensionality of Visualisation VR provides, there is an opportunity for an excess of choice, consequently, an intelligent reduction of choice needs to be integrated into its design. It is therefore important to consider constraining the system to allow interface elements to be displayed only when necessary.

The panels developed are a key area to which add constraints to. The button and slider elements that are traditionally found in forms or menu systems can be located via panels in the designed interface. These panels act as grabbable containers allowing the interface elements to be arranged inside an area that can be moved in 3D space to suit the user's preference. The borders of these panels are consistent in colour and style to highlight their interactivity, a consistent interaction colour theme was established to convey to the user that elements in this colour can be interacted with. For the modular 3D panel for buttons and sliders these elements include the white button, white panel border, and white slider anchor.

Contextual UI location allows for interactable features to only appear when in proximity to the task at hand. Much like the real world, computer users can physically walk to a monitor to select its controls. If the monitor is fixed into world space in a virtual world, the question must be asked, is it appropriate for the monitor to be fixed? The two contexts described below show how local user space and world space can be used effectively to provide UI context to the task being carried out.

3.1.1 Player Context

Consistent user interface elements are concepts that can be found in the WIMP design structure. A common feature of window controls allows for minimize, maximize, and float controls of windows or form containers. These are always available to the user and are persistent throughout the navigation of an application. In a virtual reality environment, the player's space can also be divided into fixed player UI and environment UI. The developed system shown in Figure 3.2 shows specific UI elements that can be locked to the player's position while navigating each zone. Controls available to them may be consistent throughout environment contexts, such as loading a settings menu, resetting player position, or closing the application. The specifics of what events the UI elements complete are at the discretion of the developer, however the separation between environment-specific controls and player controls is the context in which they are displayed.

For example, closing the application would likely be best suited as *player context* UI as it is always linked to the player regardless of *environmental context*. Entering an environment focused on video playback in Figure 3.5 annotated as “zone 2”, can allow for a play button to be locked and added to the player’s context while navigating in this zone, but not necessarily. The player may choose to keep UI in a fixed location in world space close to the interface’s *environment context*, which is the virtual screen in this example.

3.1.2 Environment Context

Based on the player’s location in the virtual environment, it may be appropriate to hide all UI and only display relevant UI elements based on the task in the environment. For example, if the user is interacting with the 3D map in Figure 3.5 “zone 1", then the controls for this map should be located near the map itself, as the user would be unable to see details of the map if controlled at a distance. The context here is the environment which is map control. In the other example described above, video playback, the screen is the environment context and the UI elements that are in proximity of this will be displayed.

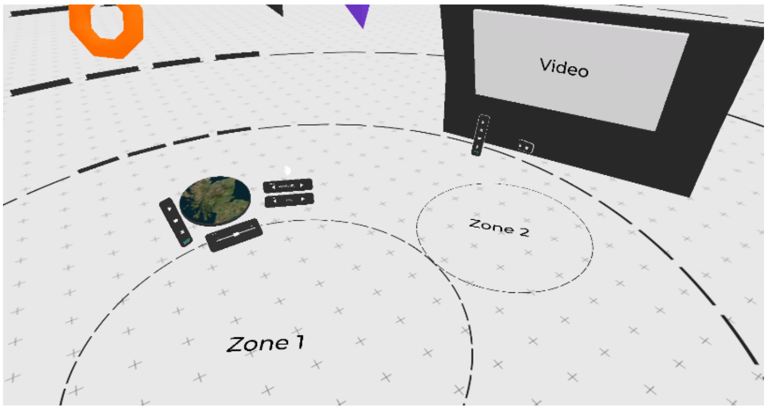


Figure 3.5 *Environment context showing zone 1 for map and zone 2 for video*

3.2 Natural Interaction Matrix

This section explores the requirements for a VR application, and how planning interaction mechanisms can support engagement with the task being completed. Evolving technologies and newer interaction techniques will likely result in the matrix changing for the requirements of an application over time. However, the developed matrix serves as a baseline for requirements gathering to support in understanding the input techniques available to complete an action for VR.

In questioning naturalism in 3D interfaces [76], two high-level mechanisms for 3D interaction design are discussed. A design technique with *interaction fidelity* and another with a focus on enhancing usability and performance by using “*magic techniques*”. The two techniques contrast in strategy, where interaction fidelity focuses on how natural the interaction is concerning a real-world environment, and “magic techniques” focus on task completion in an unnatural environment. Such as using the teleporting locomotion strategy due to lack of physical play space which would otherwise be a high-fidelity interaction. Each actionable task in a virtual space will need to be broken down to fully understand the potential options made available and select the best-suited technique for the interaction. The authors of [76] notes 3 universal UI tasks that can be viewed as high-level requirements for a 3D application, travel, selection, and manipulation.

Travel is movement in the virtual space, where both high interaction fidelity - physically walking in a tracked environment - and “magic” mechanisms - such as teleporting are options available to the user. In visualizing big data challenges [12], a set of requirements was provided that describe features a VR Visualisation platform should achieve in the context of 3D Visualisation of multidimensional datasets. This was used as a guide for the development of the conceptual model for an application with a 3D GUI focus presented in this section. The matrix shown in Table 3.1 categorizes these requirements into three interaction tasks, travel, selection, and manipulation. These are categorized further by a potential choice for their interaction mechanism, high fidelity, or a “magic technique”. Based on the requirement type, developers decide what category of interaction it falls under. The matrix is to encourage the exploration of High-fidelity and Magic interaction approaches in 3D UI development for defining universal 3D task interaction mechanisms.

Table 3.1 *Natural Interaction Matrix*

Requirement	**Scaling**	**Navigation**	**Subspace Selection**	**Object Selection**	**Object Move**
Travel		x			
Selection			x	x	
Manipulation	x				x
Mechanism	Magic	High-Fidelity	Magic	High-Fidelity	High-Fidelity

3.3 Multimodal Input

In the following sections, multimodal input and selection approaches are discussed. Providing a multimodal approach to user input has been shown to allow more users to interact with the system more effectively [77]. The availability of various input mediums is an important accessibility design consideration and is a critical component of the modular system.

Input modality is discussed in the web accessibility guidelines for “pointers” [78] where timed and complex gestures may be inaccessible for some users. Although not to be avoided in its entirety, the functional recommendation is to offer alternative methods of inputs to enable users with motor impairments to interact with the interface. The input control mechanism for the discussed 3D modular interface is supported by ray-based interaction. In addition to this popular method of input, by using gestures recorded with hand tracking - with or without controllers - we can support the user in deciding their preferred input medium.

As the user can freely move in 6-DoF, they are away from surfaces that can support fine motor skills for accurate selection of objects at range. Mouse and keyboard interaction approaches limit natural interaction grasp because of this and why for natural interaction approaches, hand tracking is a lucrative opportunity for improved immersion and intuition.

In comparison to 2D input approaches which are traditionally 4-DoF, 6-DoF interaction approaches have the caveat of requiring more physical movement which can be restrictive to some users. 2D input mediums are two orders of magnitude more precise and have less latency, both discussed in papers [79] [80]. Although attempts have been made to alleviate this, more study is required to see what other algorithmic approaches can be used.

It is for this reason that supporting numerous input mediums should be considered when developing a 3D GUI. Every panel in the designed interface has elements of buttons and or sliders.

3.3.1 Hand Tracking with Controllers

Ray-based UI interaction systems can support controller input with the use of a laser pointer. In augmented reality, the same concept of laser pointers is used as a metaphor for interacting with an augmented environment [81]. For 3D interfaces that can rely on gestures followed by the selection of interface elements, there is an opportunity for the support of controller and hand-tracking input. Supporting both allows for user preference to be exploited and the potential to use the same interface system as new input modalities are supported. At Meta reality labs, research into using electromyography (EMG) where electrical motor nerve signals are translated into actionable input for a virtual interface [82] has shown promising development although it is still in its infancy, it provides a new input medium not previously explored. With these new input modalities, interfaces currently developed for VR based on the WIMP concept will have to be refined or redefined to support the more robust input sensor hardware that is currently in development. Whereas the opportunity using the modular 3D GUI, a multimodal input mechanism is used enabling maximum exploitation of the variety of new input sensors as they are developed and released, such as BCI, Voice, Eye tracking and EMG sensors.

Not all VR headsets require controllers as they have inside-out hand tracking capability which at a minimum allows for navigation around the headsets OS via gestures – such as pointing. Specifically, in 2022 the Meta Quest 2 headset allows for navigation on the browser and loading of applications without the use of a controller. This has its advantages such as user convenience for a more passive experience such as video playback, however, not all VR applications support hand tracking. By doing so developers are limiting those with hand-tracking headsets to experiencing their product. As the future could contain more and more hand-tracking supported headsets, it was therefore deemed important to develop support for hand-tracking and controller-based input.

The Valve Index controllers can be seen in Figure 3.6 with input mappings highlighted. These controllers use infrared on the index fingers to track the user's finger at a distance without touching any of the controller itself. The grip around the controller supports tracking other fingers with the thumb having more infrared sensors available on each button at the top of the controller. This provides finger-tracking capability while also being able to support feedback via haptics.

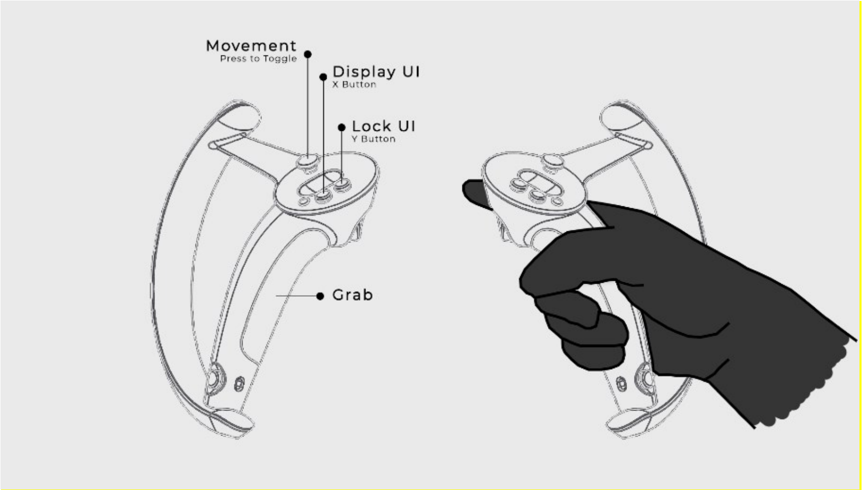


Figure 3.6 *Controls required for interacting with modular UI*

The environment is manipulated exclusively with hand tracking for this reason but selection/input with ray-based mechanisms can be used via toggle. The buttons on the controller do however provide additional functionality such as virtual space movement. Room-scale tracking is enabled by default; however, it is not always convenient, and designing for every room layout is not feasible. The joysticks are therefore used for locomotion techniques supporting teleporting and standing movement by clicking down the joystick. This keeps both mediums of movement selected by one control mechanism. In addition, this supports the same controls for each hand, allowing users to navigate using their preferred hand and further supporting temporary accessibility issues, such as users having a broken dominant hand.

The UI displayed around the user at any given time can be hidden and redisplayed at their own preference via a toggle on the controller's button. The UI can also be locked to the user's position as they navigate around the map. In Section 3.1.2, contextual UI placement is discussed where this feature opens a lot of opportunities for hidden control mechanisms in the virtual space where gestures or physical controller buttons are used to hide or show UI elements.

3.3.2 Eye tracking

With the adoption of variable rate shading and foveated rendering based on eye tracking, there presents an even larger opportunity than physical interaction design, to intelligently introduce a new input medium into virtual reality interfaces. Rather than selecting a UI component using physical hand tracking, the gaze-based selection is an alternative ray-based interaction technique, that can be used as an additional input mechanism [83] demonstrated in Figure 3.7. The user can select a UI element for interaction and confirm their selection with a button press or with a countdown while gazing at a specific element for a specified duration based on a ray directed by the user's eye gaze. By designing an interface that supports both gesture and ray-based interaction we can exploit eye tracking as it becomes more available in VR headsets, such as the Quest Pro and HTC Vive Pro.

Figure 3.7 *Designed to support eye tracking selection via gaze*

3.3.3 Voice input

Buttons on panels can be selected using voice input using the Azure Voice SDK. Each button has a required value to be said such as "left charger status." Each panel is named appropriately allowing any user who can view the panel to know its primary identifier. Every button that can be interacted with via voice commands requires a primary key to be said to activate the correct event. Alternatively, a hierarchy can be established if a button is the only one displayed on a panel. For example, the panel tag title followed by the button. Due to the significant amount of UI displayed, this was only implemented on a set few buttons for experimentation. An example of the keyword followed by a command is as follows broken down into the primary key and command associated with a button. "Charger panel, next."

Beyond the selection of buttons, such a mechanism can be used to expand menus into further groups without alternative interaction. Such a mechanism can enhance the usability and performance of the task being completed by the user by being presented with only relevant information to the task being carried out. This allows preparation for the next task while still immersed in one.

Emerging user interaction design principles could also have a place in a virtual environment. Sonic interaction design (SID) can play a key role in supporting the quality of an interactive experience. The authors of [84] describes how multimodal listening approaches such as "listening in, up, out, through and around" a play space can help form shape perception.

Although not directly implemented in this conceptual model, it does provide a notion for further expansion by exploiting audio cues for 3D objects beyond selection feedback.

3.4 Study Design & Methodology

The preliminary focus group study provided qualitative data about how 6 users reacted to a 3D GUI, allowing the ability to query a database and update visual cues on a 3D topographical map. The study aims to identify potential accessibility constraints and explore the use cases for producing high and low usability satisfaction.

Participants were invited for an initial interview to confirm they are the target demographic for this study which was targeted at users aged 18 to 55 and classified their experience between intermediate use of virtual reality and inexperienced. The participating users were aged 24-29.

3.4.1 Participant Interview

Preliminary interviews were carried out to establish a baseline user base of novice to intermediate VR users, with users between 1 and 3 on the Likert scale displayed in Table 3.2. User consent was required at this stage and was followed by an opportunity for participants to highlight any history of locomotion sickness in the context of VR.

Initial Interview Questions				
Do you have experience with motion sickness from VR or other?				
Are you at present comfortable to proceed?				
Scale from 1 to 5, where would you place your VR experience?				
1	2	3	4	5
Once or Never	Rarely	Occasionally	Frequently	Very Frequently

Table 3.2 *Initial interview questions to target users' VR exposure.*

3.4.2 Process

The in-person study was completed where all participants were in the same room around a screen which allowed the current user's VR headset perspective to be displayed, this was to encourage discussion about experienced interactions with the system. The hardware used was the same as the previous study described in Section 2.5.

Firstly, a tutorial demonstration was provided by the chair to the entire group, allowing them to view the external screen and ask questions about what they were seeing. This provided an overview of the interface to give some context and expectations on interactions.

Once an overview of the experience was presented by the chair of the focus group, the first user was placed into VR, this was selected based on their experience in VR. Those who are novices with little to no experience – Likert scale 1 – were asked to go first and users were then selected incrementing in VR exposure until everyone had completed the task.

The study demographic only had novice to intermediate experience with VR. Requiring headset adjustment and fine control over the IPD for each user which took 2-5 minutes per user. Participants only started the experience once their headset was comfortable and they could see text in the environment clearly. Users were asked the following.

1. Walk around the space and select a locomotion technique of their choice (if the default joystick wasn't comfortable, they could opt for teleport)
2. Interact with the 3D UI Panels that interface with the map and move the map to latitude-longitude locations queried.
3. Interact with the point cloud playback and view the cloud alongside the video counterpart.

During the interaction, the user in the experience was asked to share their unfiltered thoughts about their interactions. Other members of the focus group could spectate from the perspective of the VR user and participate in the conversation as well due to displaying on an external screen. After 10 minutes or so the next user would be placed into the same experience. Once half of the users had completed their experience further questions were added to the group, discussed in Section 3.4.3.

3.4.3 Assessment Criteria

Notes were carried out during each participant interaction in the focus group session. Questions were asked to support in identifying areas of interaction satisfaction partnered with the observation notes taken. **Thematic analysis** was used and will be outlined in the following section which allows for themes and patterns to be identified that can be further explored with further questioning. Key questions for discussion were asked throughout and open to anyone to respond, not just the user in the experience at the time. Two key research questions are focused on and are supported by three follow-up questions.

Research Objective 1: What aspects did you see as high usability satisfaction?

Research Objective 2: What aspects did you see as low usability satisfaction?

Question 1: How was the experience overall?

Question 2: Where do you see the application of such an experience?

Question 3: What do you think of using a 3D Interface?

3.5 Results

A total of 6 adults participated with initial interviews indicating their exposure to VR was 1-3 on a Likert scale of VR experience, with 4 users being inexperienced at 1 on the scale, and 2 being at scale 3. This defined a baseline demographic of novice VR users. No candidates were disqualified at the initial stage due to experience with motion sickness. One participant reported they do get travel sick, but due to their exposure to VR in the past were comfortable to proceed as previous VR motion sickness exposure was due to a sharp drop in HMD framerate.

Users who stated 1 on in the initial stage said they have only used VR for demonstrations at short intervals at shopping centers or a friend's house. Users who put their experience as a 3 on the Likert scale shared that they have previously owned headsets, however, it has been "a few years since using it." Key themes were consistently brought up by all users, these included, controls haptic feedback, multimodal input, task-specific usability, and UI functionality.

3.5.1 Haptics

One of the points of discussion was how users know that they have interacted with a game world object. Responses here ranged from desiring more vibration feedback to having unique sound prompts for every type of action completed. A frequent comment was that users didn't know when their finger had entered the actionable UI element such as a button. They suggested visual effects that could be added to let the user know that the button was ready to be interacted with. Due to the 3D depth requirement to trigger a button, this comment came up more than once. One comment discussed how haptic feedback wasn't fully recognized due to the sound that was played back, they said they heard the button press, and that indicated selection to them without vibrational haptics, which went unnoticed upon reflection.

3.5.2 Multimodal Input

Users could select their desired locomotion technique, either teleporting or using the joystick for a more natural walking experience, however both are considered magical interaction techniques. Real world tracking was also available providing high-fidelity movement should the participant want to adjust their position in real world space. Two users brought up that they instead felt comfortable physically walking in the place space and found that using the joystick to navigate was to move larger distances to interact with the floating interactable around the environment. Another user mentioned that a larger play space would have been preferred but also commented that if they were familiar with their physical surroundings, they would feel more comfortable. Ideas discussed regarding this included demonstrating the pass-through VR mode that allows the VR headset camera to be displayed to allow the user time to become more familiar with the physical environment. The joysticks however were not favored by one user who hasn't spent much time in VR apart from one previous demonstration, this made them "feel sick" and preferred to use the teleportation mode.

Two users once entered the environment instinctively looked for a laser pointer to interact with the UI, even after watching other participants interact. They shared that this was something they are used to in VR and haven't interacted with UI in this way before. One user stated they preferred to use their finger to interact with the slider using physics colliders rather than grabbing the slider's anchor handle.

3.5.3 Virtual Environment

One user shared that they are never in a cockpit-like environment with so many controls floating around them so learning how the elements interact with their hands was a learning curve. In addition, it wasn't always clear what elements are interactable with their virtual hands, once informed that everything in the environment is interactable participants ventured away from the menu to explore the floating 3D models in the distance. Visualisation of the point cloud scan of the geospatial location selected proved to be an exciting point of discussion. One user mentioned that they hadn't experienced anything like this before and had ideas for virtual meeting rooms to be presented like this, allowing for a fully collaborative 3D experience. Further to this discussion comments on collaborating in this environment were discussed as a benefit for sharing 3D data, with use cases being discussed in the tourism sector for viewing hotels and in the real estate sector for viewing the property.

Video playback was also a discussion once a geospatial location was selected, a user mentioned it felt unnatural to look at the video and said it was something they would have to get used to seeing a video "floating" in space. Furthermore, one user adjusted their headset went back into the experience, and instinctively tried to grab the interface without the controllers being placed on them.

3.5.4 UI Functionality

The users were tasked to interact with the menus and panels surrounding the 3D map which allowed for zooming and navigating around the world. One user shared a creative comment, that moving the UI system between locked and unlocked positions "felt like a Star Wars cockpit more than a menu" Another user praised its interaction as feeling more "natural" than a conventional UI. Users unanimously enjoyed being able to hide and show the UI based on what they were doing in the environment, allowing them to walk without the intrusion of the objects being displayed in the environment, allowing for less distraction when completing tasks. However, one user mentioned that there could be an annoyance if the UI panels are obfuscating a task being carried out at the time. In such a scenario the UI panels would be required to be displayed while also interacting with the 3D map movement.

Pertaining to extending functionality, some users felt there was a lot of freedom to move and place the UI in different locations but would have additionally liked to scale the UI elements presented allowing for complete customization of the interface elements. In addition, two users stated that they always customize the UI for applications they have and saw the benefit of real-time manipulation of the UI's location, allowing for specific UI priorities to be "kept close" to the user at relevant tasks.

3.5.5 High usability satisfaction highlights

One user who rated 1 as their experience on the Likert scale, admitted to not using their less dominant hand – left – for anything other than movement. They preferred to use their dominant hand for interacting with UI elements and having the freedom to select the hand they use for movement as both controllers supported movement was a benefit.

Another high usability feature that was discussed was how close the UI was to them after the initial setup, they found it helpful to have the controls move around with them as they explored the environment with the addition of the locking and unlocking feature. One of the more experienced users shared that they wanted to move the UI as soon as they were placed into the environment to locations that they prefer, and further discussion exposed that having UI not relevant to the current task at hand would be cumbersome.

Users thought highly of the UI being displayed only at relevant locations where tasks were completed showing a clear indication of what task the UI elements were associated with. The contextual UI in this scenario allowed users to explore without requiring them to manually disable the UI based on where they were in the world "The map UI disappears when I can't see the map which makes sense".

3.5.6 Low usability satisfaction highlights

One user stated while using the joystick locomotion technique that looking at their feet made them feel "wobbly". This was discussed where a virtual set of legs may benefit them. Novice users shared that they were concerned with how long it took them to set up their IPD and adjust the headset, and one comment suggested wireless headsets may provide less of a cumbersome experience. They shared that they were confused as to why distant objects were blurry.

Some users were concerned with the haptics vibrational frequency being set too low. It wasn't consciously noticed that it was being used, although it was admitted that they could hear the sound which may have distracted them from the vibration.

Although UI elements could be moved, one user desired to scale and snap the elements to ensure they were always straight. The current implementation allows full rotational freedom resulting in some UI panels not being "straight or parallel."

One user with a 3 on the experience scale shared that they prefer to have the traditional first-person joystick rotation movement over snapping rotation to 45°. This was described as a "smoother" transition for turning on the spot.

3.5.7 Discussion

The focus group study targeted users with little to intermediate exposure to VR applications, intending to support an unbiased view from existing VR interface design aiming to encourage discussion on novel ideas the interface presents. It provided insight into usability concerns when interacting with an exclusive 3D interface environment. Findings concluded high usability satisfaction when interacting using natural interaction approaches, such as the manipulation of the interface in 3D space via hand gestures. Low usability satisfaction was primarily focused on the haptic feedback of the interface, where depth perception using hand tracking was a concern for multiple users as they were unsure if the UI element had been interacted with.

User safety is of the highest concern when working with any population. Initially users were asked if they had experienced motion sickness from travel or VR in the past [85]. The study was designed to focus on users' movement being minimal in the real world and virtual world as motion sickness is primarily experienced by those who passively travel [86] ensuring they are as comfortable as possible while interacting with the presented interface. All users felt comfortable proceeding with the focus group as the one user who has experienced motion sickness in the past had intermediate exposure to virtual reality experiences. Users were also offered to sit down if it would make them more comfortable as the experience is also possible to do this way, but users wanted to physically walk between interfaces which drove discussion around how UI is presented in different areas of the environment. Topics such as object occlusion between interaction areas were raised because of this.

Participants expressed that they enjoyed being in the environment and interacting with the various objects and interfaces it presented. Users were encouraged to discuss what they experienced in an informal manner, what they enjoyed, and what made them uncomfortable to interact with. This generated a significant amount of relevant conversation exposing ideas on how the presented interface could be used in various applications along with their usability considerations. The 3D map of the world provided an element of familiarity to users, standing as a metaphor that all participants reported being exposed to in the past in a more traditional way via smartphone and desktop applications. Interacting with the map in a fully 3D immersive environment was reported to be an enjoyable and memorable experience due to the familiarity and comparison users could provide.

The study format was focused on the user discussion with the headset being rotated between each user leaving time for discussion while one user was interacting with the various panels containing buttons and sliders. This allowed for the formulation of thematic discussion, leading from one idea area to the next and exploring the usability considerations in various contexts.

3.6 Conclusion

This chapter explored the use of a newly developed modular 3D interface, focusing on supporting current and future input modalities. An important aspect to consider before exploring the opportunities made available by a 3D GUI was the vehicle in which to showcase the interface itself to participants with appropriate feedback. The developed application provided a conceptual model using a 3D topographical map, linked to geospatial locations allowing for navigation and viewing selected locations through a point cloud scan. The modular panels provide a novel interaction mechanism that leverages the strengths humans use in their day-to-day interactions and empowers the user to adjust real-time interface elements to suit their exact needs and preferences.

Overall, the focus group discussion provided key areas of future work, where notably low usability satisfaction was perceived with appropriate haptic feedback being a primary concern. The participant's sentiment overall however was positive during the experience, with many ideas being generated for how such an interface could be used in a practical application.

Limitations of the study included time allocation per user, where some users took longer to adjust to the virtual environment than others which was mainly due to headset adjustment and locomotion strategy. Some users naturally immersed themselves in the VR environment and were comfortable interacting with the various elements it provided with minimal direction. Whereas other users needed more guidance through the application flow, and to understand that the entire environment is interactable with their virtual hands.

With the rapid adoption of VR headsets and their sensor capabilities - such as eye-tracking, this growing area of study appears to lack a multimodal development approach beyond using typical ray-based interaction. It is therefore important to iterate 3D GUI design concepts with users to expose usability constraints with various input mediums. The presented study provides two key contributions. Firstly, a design framework for VR focus groups for use in iterative development. Additionally, the practical guidance on how to develop a modular GUI based on its core design principles and showcases its practical application with users.

Chapter 4

Holistic VR Optimisation

Building on previous Chapters discussions, we recognize that user interaction is critical for engagement in virtual environments. Therefore, it is essential to address and minimize digital artifacts that could detract from the user experience. Stages in the VR pipeline implement optimisation strategies that are exclusive to VR due to the nature of stereoscopic rendering. The following chapter seeks to highlight the importance of these processes in the VR render pipeline and explore these requirements which are predominantly designed based on the human. We must exploit flaws in the human optical system to produce a visually seamless experience as VR utilizes the same hardware as traditional 2D displays but requires higher resolution, display refresh rate and lower latency or risk locomotion sickness and an unusable experience.

With an holistic overview of the VR pipeline being made, the discovery of a novel, VR-exclusive renderer is discussed. Frameless eye-tracked foveated rendering (FETFR). In this approach, rather than waiting for a frame to be fully processed for display, we instead render pixels and display them to the user as soon as they are processed on the back-buffer. Striving to reduce latency and exploit the human's optical system further, we also integrate the use of eye-tracked foveated rendering, an eye-tracking solution that has only recently become integrated in commercial headsets. The following chapter seeks to highlight the stages of the render pipeline holistically and concludes with a discussion on implementing EAFR using the Vulkan graphics API, with an outlook on the core considerations that need to be made during its implementation.

Another key finding during the review of optimisation stages used exclusively for the HMD, is the lack of further interaction processing being completed. Where the HMD has significant research and implemented solutions for improving visual stimuli, there are still no native interaction solutions implemented by headset manufacturers. This raises the question, what optimisation approaches are available specifically for user interaction? In addition, what does the VR *interaction* pipeline look like?

The research delves into the requirements of HMD technology and investigates techniques that can be exploited for the interaction layer of VR development, aiming to enhance user experience in virtual space.

The scope extends beyond achieving high frame rate, as numerous algorithms are already known to bridge the gap between the 3D world and rendering frames. Instead, discussion aims to identify algorithms that can instead improve interaction systems between the user and the VR environment once implemented. Additionally, the study explores alternative approaches for handling processing complexity for human factors in computing, including investigation into the interoperability between the user's real-world movements and their interaction with the virtual physics engine. Here we leverage control system theory for improving real-time physics engine interaction. By striving for a holistic approach, this research aims to contribute to the advancement of VR technology and its usability.

4.1 Interaction Overview

Virtual reality has proven to be a cross-disciplinary technology allowing researchers and the entertainment industry an opportunity to redefine how we consume and interact with digital environments. With this, the race for high-fidelity output in a small form factor has resulted in numerous manufacturers releasing HMDs to the public market since the Oculus DK1 was released in 2012, each with different processing capabilities and interaction mediums. Such features include providing a consumer-friendly low-cost HMD, increased resolution per eye, increased field of view, and 6-DoF interaction controls all while decreasing form factor and latency. In addition, the VR render pipeline has recently introduced a variety of split processing techniques to support rendering high-fidelity images using a standalone HMD. The rendering pipeline for VR remains proprietary among headset manufacturers and therefore some details remain obscure for low-level processes - such as the specific complexities during the asynchronous re-projection formula used, keeping their competitive advantage whilst decreasing cost through software.

The development of shared entertainment has evolved from providing a repeatable experience to an audience in theater performances, to now evolving the *big screen* for cinema entertainment, where a 1:1 reproducibility of the experience can be provided. We are now, therefore, in the interaction era. Where we originally modified amphitheaters to accommodate hearing, we must now accommodate further human factors in an immersive digital format, such as preventing locomotion sickness.

A high-fidelity locomotion system that uses room-scale tracking can be enhanced by emulating users' legs and torso in the digital environment with inverse kinematics. This is an algorithm introduced between the user and their virtual space, enhancing the sense of embodiment with the digital self. Embodiment studies have already addressed the neglect of the sensorium and the human body where ocular centrism is instead focused on, being the prioritization of vision over other senses [87]. It is therefore unsurprising that the current VR render pipeline is almost exclusively focused on ocular techniques to immerse a user into a digital environment.

Are there more scenarios where algorithms can be placed between the user and their digital input to enhance presence whilst providing increased usability satisfaction?

Developers must integrate bespoke optimisation pipelines for each unique interactive experience during VR development, striving for low artifacts and latency. Although developers don't have access to specific HMD hardware changes in standalone headsets, they are instead left to focus on how they can optimise their developed experience to run on the hardware they have selected to support. These are typically 3D engine optimisation steps that the video game industry has been using to present 3D worlds onto 2D graphical displays.

The following section explores VR optimisation stages that are completed in the VR render pipeline. In computer graphics, we must acknowledge direct comparisons to non-VR pipelines as they are directly related. Specifically, the GPU buffer must undergo more alterations before displaying the frame buffer in VR due to the nature of stereoscopic displays, distortion, and 6-DoF tracking.

Conventional real-time 3D development pipelines utilize engine features such as occlusion culling and level of detail systems. However, VR pipelines integrate additional strategies that consider human factors in frame rendering to enhance application usability and minimize artifacts, as detailed in Section 4.2. This necessity arises from the critical need to avoid digital artifacts that degrade the user experience, potentially leading to issues like locomotion sickness caused by interaction to photon latency on the HMD. These VR-specific approaches are crucial for meeting HMD framerate metrics which ensures usability.

For current commercial standalone headsets such as the Meta Quest 2, Pico 3, and Quest Pro there is a baseline for minimum metrics that a standalone headset experience should hit such as a framerate above 72fps although, in ideal scenarios the frame rate should match as high a refresh rate as the HMD can produce (90Hz is the current trend for standalone). Meta Quest performance target guidelines suggest "interactive applications must target a minimum of 72 frames per second" [88]Although not explicitly described, the Oculus DK1 supported this refresh rate as a minimum with the option to run at 90Hz if processing allowed [89]. It should be considered that this is also a hardware limitation for the formfactor of the DK1 HMD at the time of study, and that if higher refresh rate displays were available and systems could run them at that frame rate then it would be exploited instead. A study validating perceived refresh rate during moving tasks however, found that 120fps is an important threshold for VR to prevent simulator sickness (SS) which is an interchangeable definition to locomotion sickness induced by VR usage. The study also found that users were able to adapt, thereby compensating for low refresh rates, such as 60fps, by predicting or “filling in the gaps to try to meet performance needs” when presented with a moving task.

Frame rate and refresh rate targets are guides to deter artifacts – such as input lag - from appearing to the end user, preventing locomotion sickness, and providing a more seamless experience. For standalone VR headsets running on mobile chipsets, there are processing constraints. Desktop VR applications allow for dedicated GPU and CPU hardware that support higher processing and memory capabilities that may not ever be possible to fit the form factor of a standalone headset. However, it should be noted that using technology such as WebRTC, standalone headsets can be used with a desktop machine enabling split processing techniques. Here, wirelessly streaming the processed VR frame is completed on the more resource available desktop device and streamed to the HMD which is exclusively responsible for headset tracking.

As HMD hardware evolves, screen refresh rates increase which requires higher GPU requirements, here lies a critical area for adoption that requires optimised application development. Foundation-level optimisations in the engine of choice however will branch across headsets and are not a one solution for every application, it requires optimisation for the specific application being developed, such as a large open world experience vs a linear guided experience with smaller scene size. Due to the variety of experiences that an HMD can render in 3D, there will be applications where optimisation will require extensive optimisation for compatibility with standalone and will be discussed.

4.2 Rendering Optimisation Processes

The following sections discuss how to improve system performance by exploring how the VR render pipeline creates a continuous and seamless visual experience, where the boundaries between frames are imperceptible to the user. Providing a fluid perception of the virtual environment striving towards an in-like real-world viewing experience, reducing motion sickness and therefore reducing accessibility barriers required. Due to computationally complex requirements for rendering a high-fidelity experience, we must exploit weaknesses in the human visual system to reduce the amount of computational processing required. A novel area of discussion to add to the VR render pipeline is also made in Section 4.2.4, Frameless Eye-tracked foveated rendering. Where a design is made combining frameless rendering which is designed for low latency frame outputs along with eye-tracked foveated rendering for use with physically based rendering approaches. Each technique has been proven in isolation, but due to the fundamental change in the fragment shader an end-to-end HMD solution has yet to be implemented. To support frameless rendering with eye-tracked positioning, the fragment shader must be rearchitected. This shader, which determines the color and position of each pixel, needs to accommodate the newly available eye-tracking technology now featured in commercial VR HMDs whilst not using current native graphics API for frame-based rendering.

Optimisation considerations based on the human need to be made during the design of interactive and scalable real-time systems. Current HMD devices require two displays to render output in each V-Sync cycle in 11ms for a smooth 90fps experience. To prevent locomotion sickness current HMD display and lens technology limitations such as the vergence accommodation conflict without a varifocal lens must be improved upon.

This section aims to discuss the VR rendering schedule with a discussion on techniques to optimise the render-to-photon pipeline, with a focus on human factors from the back buffer to the resulting viewed frame buffer. Multiple optimisation techniques are used together in a single V-Sync cycle to create a seamless visual user experience with techniques such as low persistence reprojected frames to *trick* the user's visual and vestibular system into being presented in a seamless VR environment. In addition, it is highlighted that many considerations in the VR pipeline have focused on the user-to-photon aspect. To enhance this design, discussions are also made on novel algorithms that lay between the user and their input, which is rarely discussed in existing literature. It therefore describes how we can consider integrating optimisation algorithms between the user and their input.

Unlike conventional 2D monitors where a camera frustum plane is used to render the frame buffer output to the user, VR requires multiple steps to effectively immerse a player using HMD with a lens. The following graphic is one such example of a typical VR render pipeline, showcasing CPU and GPU-bound tasks to be completed in a single V-Sync cycle. Each time period of the graphic is estimated based on a 144hz display, resulting in an estimated V-Sync cycle of ~6.94ms. Fluctuations have been considered based on VR rendering patterns identified through development studies completed in Chapters 1, 2, and 5.

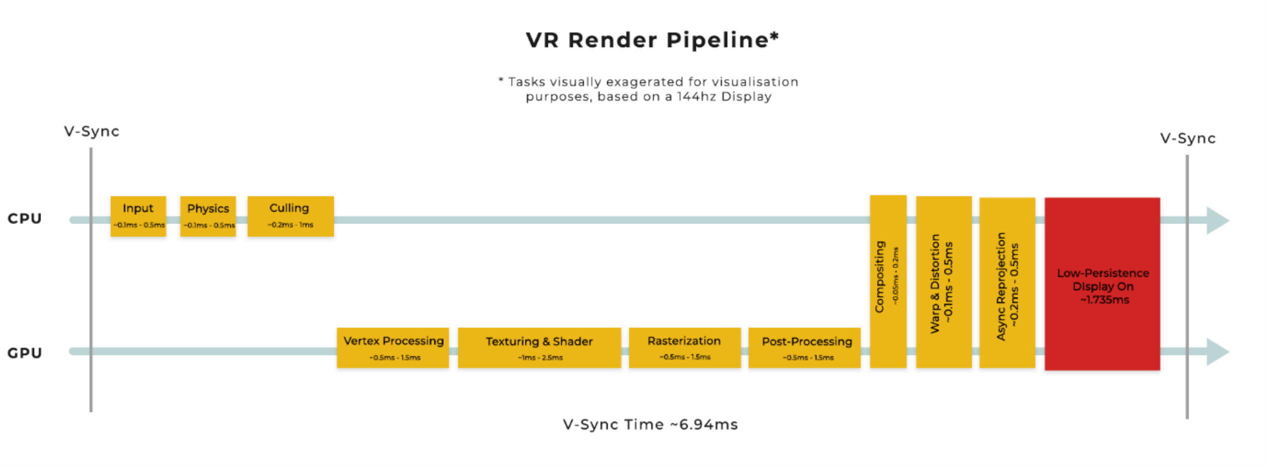


Figure 4.1 *CPU and GPU bound tasks to complete in a single VR V-Sync Cycle*

4.2.1 Motion to Photon Latency Prevention

In VR the act of turning your head to explore the surrounding environment whilst keeping your gaze centered within a given camera frustum is often referred to as “screen panning.” This results in motion blur as the user adjusts their head to get the expected perspective of the digital environment.

Motion-to-photon latency or delay (MPD) is considered in this movement, and can also be referred to as end-to-end latency, where the delay between the user’s head movement and the change of the display's frame output. As soon as the user's head moves, the VR scenery should match this movement. The more delay (latency) between these two actions the more unrealistic the VR world seems. To keep latency as real-time as possible, a latency between <7ms to 20ms is required [90]. For a 90fps experience, this allows for 11ms (0.00011 seconds) per rendered frame on a 90Hz display. Should rendering take longer than the allocated 11ms then reprojection or “time warping” can be used and is discussed in Section 4.2.5.

The refresh rate is therefore a critical component to consider during the VR render pipeline, it is the total amount of times a display can display a new frame from the GPU. It dictates the length of latency between each rendered frame, with a higher refresh rate resulting in less latency between frames. In a paper exploring “methods of verifying pursuit accuracy” a list of display-induced motion artifacts is discussed [91]. These are all specific artifacts as a result of low refresh rate.

- Blur: Resulting from slow pixel response when trying to generate smooth-pursuit eye tracking (SPET).
- Judder or edge flicker: Caused by a low sampling rate combined with high contrast.
- Ringing: Unwanted ripples in the image caused by excess overdrive with spatial high-pass filtering.
- Ghosting: Signal crosstalk and image retention result in faint duplicated images or image trails.
- Large-area flicker: Caused by asymmetric fall times with unbalanced voltage inversion and can be influenced by loose HMD cable connections in a desktop VR experience.
- Dynamic false contours: Caused by SPET or incorrect bit order.
- Pursuit and saccadic-related color breakup: Caused by low field rates.

Nvidia and ASUS have disclosed their research ambitions toward a 1kHz refresh rate display, this ambition would remove motion blur without introducing a low persistence requirement. Motion blur can be added with a post-processing effect in games, where introducing a blur effect between frames of moving objects supports users' eyes to track movement without introducing jitter. Persistence in this context, is known as “moving picture response time” (MPRT.) It is theorized that viewing 1000fps on a 1000hz display would eliminate motion blur. However current HMDs can support 144hz (Valve Index) therefore significant form factor development is required to result in such high refresh rates on a small HMD panel. With only one commercial desktop monitor released as of writing supporting 500Hz, the Alienware AW2524H. The minimum frame rate requirements for VR HMDs are said to be 85Hz [92] which is not standard for standalone headsets, with the Meta Quest 2 producing 72Hz and the Quest 3 just recently supporting 90Hz. It is therefore important to discuss how to reduce latency in the VR pipeline through software approaches until high refresh rate headsets are supported.

Low persistence HMDs are a requirement by design where displays use a refresh rate less than human eye perception. A low persistence display only displays the rendered frame buffer for a fraction of a millisecond before being turned off or “pulsed” until the next frame is ready. This blackout period displayed to the user significantly reduces the amount of time each frame is visible on screen, resulting in a low persistence frame which allows for the user’s eyes to stabilize their gaze on world-fixed objects through the vestibulocochlear reflex (VOR). In traditional displays, each frame is displayed continuously until the next frame is ready on the back buffer of the GPU which can lead to motion blur as the users move their head/perspective. By introducing this pulsed effect, we decrease the motion blur effect caused by head movement in fast-moving experiences thereby reducing motion sickness effects and providing higher framerate perception. The main artifact of introducing such an effect is strobing.

Taking into consideration pixel switching time is another factor when aiming to reduce latency. This is the time taken to update all the pixels on the display. Although primarily at the discretion of the display's hardware it should be noted that future development using OLED displays boasts up to 100x faster response time than LCD displays [93]. Software approaches do exist however to improve visual fidelity for cheaper, more available LCD displays over OLED, such as local dimming. Local dimming allows specific zones of the LCD backlight to be independently dimmed or brightened, enhancing contrast and providing deeper darker blacks for virtual environments in this context. It is not supported by all LCD displays however and is a hardware requirement to exploit.

4.2.2 Adaptive Frameless Rendering

The following two sections explore novel VR rendering approaches which when combined could provide significant compute availability for real-time ray-traced scenes. Recent discussions on 3D scene render pipelines have found a frameless rendering approach to be favored with low latency displays such as OLED. Scene complexity with dense pixel resolution displays and their currently double-buffered rendering approach can result in high compute power for frame output. With the double-buffered approach, an off-screen buffer is stored for the image being constructed, by the time a user moves their perspective in the rendered world it is out of date, resulting in outdated information being rendered. It must be noted that this is across single and multipass rendering techniques.

Adaptive frameless rendering [94] or alternative frame rendering (AFR) techniques aim to minimize latency to the user. Pixels in this approach are sent from the back buffer to the frame buffer as soon as information is available and typically sampled at random. This can result in artifacts such as "pixel dust" and the perception of motion blur without any post-processing. Only when each pixel has been correctly rendered for the current frame are the post-processing effects allocated. With lower sampling rates compared to framed rendering, adapting sampling rates and reconstruing pixels based on regions of spatial or temporal changes will require physically based rendering [95]. AFR uses sampling based on changed rendering output between frames in both spatial and temporal space. Spatial is the 3D environment being rendered, while temporal dimension takes the duration and sequence of frames as they are displayed.

An advantage of using physically based rendering (PBR) allows for selectively sampling the image plane in the scene compared to rasterizing the scene image to sample. Interestingly, authors have discussed [95] how adaptive sampling domains such as temporal anti-aliasing could also benefit from the introduction of frameless rendering. In addition, it is common for PBR to be used in conjunction with ray-tracing scenes as PBR defines how surfaces will respond to light whereas ray tracing defines how light moves through the scene.

The implementation of a frameless renderer was discussed in 2016 [96] and has not gained much traction in the industry since. It is possible to support this rendering variant without requiring all released VR projects to be updated. But there are concerns which may arise. When integrating eye-tracked foveated rendering, this is a proprietary SDK that needs to be integrated by developers for their developed HMD - although OpenXR has recently introduced it in its framework. As eye-tracked foveated rendering is being adopted the HMD manufacturer would have to develop for these changes. Although, as legacy VR experiences developed years ago likely require less processing power than today, it's possible that the introduction of frameless rendering would negate the optimisation benefits that eye-tracked foveated rendering could provide.

4.2.3 Eye-Tracked Foveated Rendering

Eye-tracked foveated rendering (ETFR), sometimes referred to as dynamic foveated rendering, may be one of the most influential adoptions to reduce computational complexity for VR rendering and is likely to become a consistent integration with all future HMDs. It is an example of how reverse engineering perception can result in a viewing system designed for the human. All objects being rendered – unless masked or occluded – inside a camera's frustum will be projected onto the viewport and displayed. The viewport is defined by the near and far clipping planes which can be adjusted to suit the render distance and field-of-view desired. In ETFR, we use the same camera frustum but only render the area where the user's gaze is focused in full fidelity, with the surrounding periphery objects being at a lower fidelity.

Fovea's are small regions of the retina where visual acuity is highest [97], any light hitting this area of the eye will allow for the most detail to be captured whereas peripheral vision (areas around the fovea) captures less detail. Given that this is the case, it is sensible to suggest that we should only render areas that are in focus to the fovea, leaving the peripheral vision at a reduced quality allowing for a significant reduction in computation complexity without visual loss perceived by the end user. By enhancing the detail only to where the user is looking, we can improve user experience by deploying to higher resolution displays without requiring an increase in hardware specification to run at native resolution.

However, eye-tracking integration requires a low-latency and high-accuracy representation of the user's gaze, especially during saccadic eye movements when the user moves their eyes quickly. Maintaining accurate tracking during fast movement is critical to prevent the user from experiencing visual latency. During smooth pursuit, when the eyes follow a moving object, saccades are minimized, leading to smoother tracking. When the user or the environment is in motion, saccadic movements can still occur. Due to saccadic omission, where the brain suppresses visual stimuli during saccadic eye movements [98], the requirements for precise tracking can be slightly relaxed during these brief periods.[103] There are four basic types of eye movements to consider: saccades, where the user moves their eyes quickly to the point of foveal fixation; smooth pursuit movements; vergence (or disjunctive) movements, where the eyes converge or diverge based on the distance of the object from each fovea; and vestibulo-ocular movements, which stabilize the visual field during head movement.

An Nvidia study focused on latency requirements found that eye tracking latency between 80-180ms caused a significant reduction in acceptable foveation, this is in comparison to a latency of 50-70ms where foveation threshold was tolerated [97]. In addition, the study found that the visual angle on retina projection with a VR display resolution of 1080 x 1200 per display was equivalent to 5 arc minutes per pixel. The human visual system can resolve gratings as fine as 0.4 - 0.5 arc-minute per pixel. An arcminute per pixel is a unit of measurement on how much detail users can perceive and resolve by the visual system. The resolution 1080x1200 is therefore 10 times lower than what the human eye can perceive which can be found on existing HMDs. In addition to arc-minute per pixel, pixels per degree (PPD) is another term for accounting for the distance from the display and its resolution. To translate, 0.4 arc minutes is around 150 PPD.

Increased display resolution would improve results, however, as discussed in this section, there are numerous metrics to consider. Just resolving arc minutes to human eye resolution, does not necessarily mean the rendered environment is believable or provides enhanced "realism". Considerations outlined in this section would need to be validated. Such a validation study should also include the mechanism for interaction with objects or the fidelity of the objects being rendered, hyperrealism vs. abstract concerning real-world viewing environments.

It is relevant to note that the arc minute per pixel value depends on the distance from the display. In the study described above [97] a VR HMD and monitor were used. Where a VR HMD is typically 2.5-5cm from the user's eye, the exact distances for both devices were not specified.

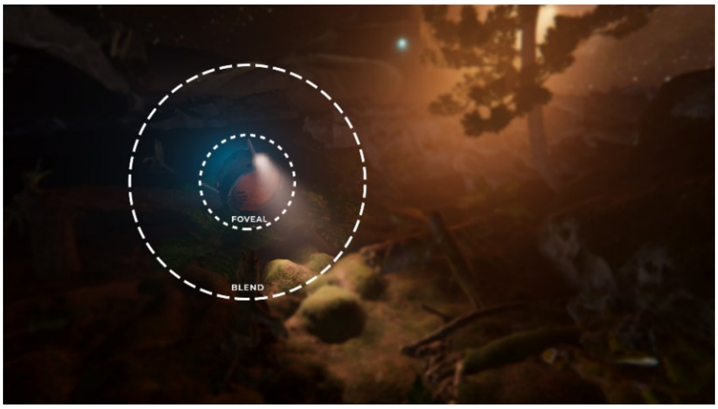


***Figure 4.2** Real-time ray-traced scene with overlay representing eye-tracked foveal areas*

In Figure 4.2 an exaggerated demonstration of a 100% focused fovea region is shown where the users' fovea focus would be centered, with gradually decreasing clarity away from this origin point. Various foveation techniques exist to blend these regions, such as Gaussian blur, subsampling, and foveated contrast-preserving subsampling (fCPS). In the paper [97] exploration into the peripheral eccentricity for the peripheral blur at varying degrees testing 5°, 10°, and 20°. This refers to the angular distance from the central point of vision. The study included synthetic eye tracker latency to understand the threshold which would degrade the user experience to noticeable effect amongst users. Results showed that more foveation in the periphery using specific blending approaches can allow for more "compression" or reduction of visual detail in the peripheral vision without sacrificing the end user's visual experience. The fCPS approach allowed for significantly more foveation than subsequent subsampling methods such as Gaussian blur which uses a pixel-weighted average method, even at 10° of eccentricity. The fCPS approach is unlike Gaussian where fCPS is focused on preserving contrast on the rendered output.

The discussion above explores the intricacies involved when developing a viable eye-tracked foveated rendering solution. With latency possibilities now as low as < 33ms we are seeing fewer drawbacks to implementing an eye-tracked solution with low form-factor sensors being used. These are primarily camera vision-based allowing for small form factors to fit inside an HMD, including standalone devices. Existing VR applications would require an update to support the additional tracking hardware made available and are not currently supported by the VR HMD vendor directly. As discussed in Section 4.2.2, the inclusion of eye tracking into the OpenXR standard may change this in a few years, providing a more linear pipeline for the integration of ETFR into future HMDs as they support eye-tracked hardware.

An area of interest would also take object saliency into consideration. In where an object's quality in a scene is designed to stand out more through properties such as contrast, colour, brightness and motion. These objects may by design attract more of an end users gaze due to the distinctive properties in context to the surrounding environment. With this taken into consideration, the selective rendering approach discussed in the paper [99] prioritised objects with a saliency map, over those in the peripheral.

In the context of VR, saliency also has the potential to enhance the VR rendering pipeline, particularly during the asynchronous reprojection stage which is outlined in Section 4.2.5. For instance, in video game environments, objects that require high fidelity and are crucial for interaction could benefit from a saliency priority flag. By marking these high-saliency objects we can ensure that important objects are given precedence during asynchronous reprojection.

4.2.4 Frameless Eye-Tracked Foveated Rendering

Both adaptive frameless rendering and eye-tracked foveated rendering (ETFR) approaches in the rendering pipeline aim to reduce latency to the end user. By combining these two new technologies we can introduce what appears to be a highly efficient, novel render solution that aims to only render and sample scene view data when it's being consumed by the user. Although research in this area is scarce, the following will outline the potential advantages of such an integration into the VR pipeline that will be referred to as Frameless Eye-tracked foveated rendering (FETFR).

AFR sends pixels to the frame buffer as soon as they are calculated based on random sampling. By reducing the resource allocation outside of the user's focus area/gaze we can render pixels only in the foveal region where the user is looking resulting in less detail in the periphery by design in the VR render pipeline. By introducing AFR alongside ETFR we can select pixels within the foveal to be sent from the back buffer to the frame buffer at the highest frequency possible. Sampling data outside the foveal – outside the blend space in Figure 4.2 – can be completed at a significantly lower rate as the user isn't directly looking at those objects.

An important aspect to consider for such a theorized pipeline is eye-tracking sensor latency. Providing low-latency data streams is an essential component as any lag between large eye movements (saccades) and display updates can limit the extent of foveation and corresponding performance gain. However, as discussed in previous sections 50-70ms is an acceptable sensor latency for tracking the user's eyes. This is already integrated into commercial HMDs and is within this tolerance limit. In addition, other processes exclusive to VR rendering over traditional render pipelines such as the described techniques, Asynchronous reprojection, and distortion correction need to be considered.

4.2.4.1 Implementation Challenges

For the integration of such a system into a game engine such as Unity, the graphics API would need to be modified, taking back buffer information and more importantly the timing of processed pixels. The Unity engine supports DirectX, Metal, OpenGL, and Vulcan graphics API's. Therefore, direct modification or extending functionality from these graphics APIs would be required. Considerations need to be taken for integrating into VR, every micro-frame processed needs distortion correction applied. It is unclear at which point in the pipeline this would be most effective. Whether latency would increase if each pixel is appropriately distorted or if micro-frame distortion, or even distortion after the complete processed frame would be suitable where post-processing can be applied.

A theoretical pseudocode implementation of this solution can be found in Appendix B. This pipeline has constraints that are not known until a practical implementation can be found. In particular the following are areas of concern. At what point is optimal to apply distortion correction techniques, at a per-pixel level, at a micro-frame (the fovea at 10 arc minutes), or at the end processed frame? It would appear logical to break the process down into manageable micro-frame chunks. The foveal micro-frame, peripheral micro-frame, and then the remaining surrounding pixels from the camera frustum. In addition, post-processing may be best suited at the end of the V-Sync cycle. Bounding frameless rendering to V-Sync cycles is still required as low persistence is still needed by injecting a black frame at the end of each V-Sync cycle to prevent motion blur and locomotion sickness. This again raises the question of what sample rate should be used for presenting pixels outside of the fovea region, and at what rates the pixels need to be updated. The sample rate determines how frequently the visual data for each pixel is refreshed to reflect the change in the scene and commonly can be associated with the fps. Is a typically high pixel sample rate required for the periphery as the user is not directly looking there? This would result in extending the lifespan of a periphery pixel in each V-Sync cycle vs the alternative of reducing the pixel sampling rate for the periphery. Sample rates could be decreased along with the resolution of the periphery due to the blur effect being used on in ETFR. Thereby reducing the amount of pixels required to be sampled and updated.

Realtime adjustment of the periphery resolution using adaptive resolution or dynamic resolution scaling (DRS) techniques allows the rendering resolution to change in real-time and allows for the resolution in different areas of the screen to be dynamically adjusted.

Asynchronous reprojection would also have to be accounted for, the users' gaze may open up for a laxer latency interval during head movement as focal time is increased during saccadic eye movement in addition to the periphery clarity being reduced.

4.2.5 Asynchronous Reprojection, Prediction Techniques

To accommodate for motion-to-photon latency caused by continuous head motion, asynchronous reprojection (ARP) or time warping allows the GPU to render frames with updated positioning based on users' movement between frames. It distorts the last rendered frame and makes adjustments to the frame based on the user's new head position. By using the very latest head-tracking data through to frame output, we can minimize perceived latency even if there has been significant head movement between the last rendered frame and the current frame.

ARP is an implicit process to the VR render pipeline and doesn't require developers to enable or integrate SDKs for usage in VR. Its integration is fully completed by the VR software stack (for example, Steam VR).

Each HMD vendor uses terminology for producing this effect or has variations based on their prediction technique, for example, asynchronous reprojection is used by Google and Valve whereas Meta / Oculus headsets use asynchronous time warp and asynchronous space warp, where this also includes depth information.

Nvidia's approach uses pre-emption allowing for the GPU to be interrupted via the command push buffer. By using the command buffer override you can inject the latest frame for frame submission being the next stage in the pipeline. Pre-emption is an overhead to the asynchronous reprojection pipeline and can result in delays by 1-2ms for single displays, thereby causing up to 4ms for a VR HMD due to two screens being displayed. Single pass rendering however can support reducing processing time compared to multi-pass by rendering once, utilizing cache coherence between the camera's positions, speeding up the reusability of rendered pixels for each eyes screen vs. the multi-pass rendering approach which will require both displays to be rendered individually. It should be noted that there are limitations with single pass with regards to shader development and may require adjustment to ensure both screens on the HMD are being rendered in respect to two displays, and not one single camera.

4.2.5.1 Limitations

Spatial reprojection and temporal reprojection are two forms of reprojection techniques used in ARP which have limitations associated with them.

Spatial reprojection injects a single rendered image before the vertical blanking interval which is the moment the GPU starts rendering the new frame to display to the user. By transforming the frame to the approximate new camera position based on the users' new head position. This technique uses pixel data from the existing frame to estimate what rendered objects might look like based on the change in viewer perspective within the same time frame. This approach is best suited for perspectives that aren't changing dramatically between frames with minimal complexity in the player's viewport or camera perspective. Temporal reprojection, however, is more complex. It leverages data from previously rendered frames which are cached over time to generate a new frame. By comparing previous frames to the current frame, we can attempt to predict what the next frame will look like, including accommodating moving objects in a complex environment. It can be summarized as an interpolation technique that may be proprietary to the HMD vendor. Artifacts introduced by temporal reprojection include "ghosting" where trails of objects being rendered are displayed. This is primarily caused by low frame rate, which may be influenced by temporal reprojections processing requirements.

In addition, phase sync is a technique used by Meta / Oculus and proprietary to their VR render pipeline. As mentioned, different HMD manufacturers have terminology and approaches that may be proprietary to their render pipeline. Phase sync is the same as asynchronous reprojection techniques which can be compared to established literature, disclosing the technology to adjust reprojection as close as possible to the frame buffer would minimize the time interval between the last possible sensor reading and the output of the reprojected frame and is therefore highly desirable from HMD manufactures resulting in proprietary solutions which provide a commercial advantage.

4.3 VR Display and Lens Considerations

The vergence accommodation conflict (VAC) is a visual phenomenon occurring when VR headset users experience mismatching cues between vergence and accommodation of the eye [100]. Unless the user is focusing on an object at the perceived distance of the display, consideration for the lens focal distance needs to be acknowledged to allow the eye to accommodate a natural focal distance. This conflict of having 3D objects rendered at a distance whilst being forced by the nature of 2D displays (with a fixed focal lens), results in eyes trying to accommodate the focus of objects at various distances using stereoscopic displays. The following section aims to outline significant advances in the field of optics and display in the context of VR and AR displays. It has been identified that research interests are targeting small form factor displays capable of producing a wide field of view (FoV) while having strong image detail.

There are alternative display solutions also available that aim to solve the VAC problem but do have compromises such as tedious manufacturing processes and limited FoV. This section aims to examine recent developments in display technologies concerning VR and AR with a discussion on future potential integrations.

4.3.1 Vergence Accommodation Conflict

To provide the perception of depth, VR headsets project cameras for each eye display slightly offset from each other. This allows each eye to converge at a single focal point allowing for the object to be viewed in 3D with depth. By rendering the output of offset cameras, we can provide a stereoscopic solution, however, human eyes converge the closer the object is to the eyes whilst trying to “accommodate” for the distance of the object. These two processes are naturally integrated into real-world focus, as vergence and accommodation are working in unison to provide variable focus adjustments to objects at various ranges.

Given this, to reduce the vergence accommodation conflict (VAC), a **vari-focal** display can be used, which adjusts the depth of a single virtual plane to match the depth where virtual objects are located. This plane adjustment allows objects in the scene to have variable focal distances depending on their distance from the virtual camera, which is then output to the HMD. This solution requires eye tracking to calculate the depth of the focal plane based on the user’s gaze, enabling the vari-focal system to function effectively. However, this isn’t the only solution to the VAC problem.

A **multi-focal** display allows for multiple depth planes to be rendered, effectively slicing virtual objects in a 3D volume and allowing true depth to be calculated. In this system, every 3D object in the virtual environment is projected to a custom focal point, requiring a volumetric display. This form of display is significantly more complicated to implement due to the need for numerous focal planes. Further discussion on this display technology is made in Section 4.3.4.

Both varifocal and multi-focal displays require lens adjustments in real-time as the user’s gaze focuses on different objects resulting in various depths and convergence. Recent research in this domain has resulted in potential candidates for real-time focal length adjustment. One such example is a proof-of-concept AR display using an Alvarez Lens [101] which is a tunable lens made of two sub-elements with complimenting phase profiles allowing for focal length to be adjusted. Due to the lens support for a large tuning range with a thin form factor, it appears suitable for VR and AR devices. The research team used Pancha Ratnam-Berry optical elements with irregular phase profiles which is required to create an Alvarez lens. Phase profiles describe the frequency of grating distributed over the lens. If the period doesn’t change, then the phase profile will stay the same regardless of whether the gratings on the lens are blazed, step, or in binary format.

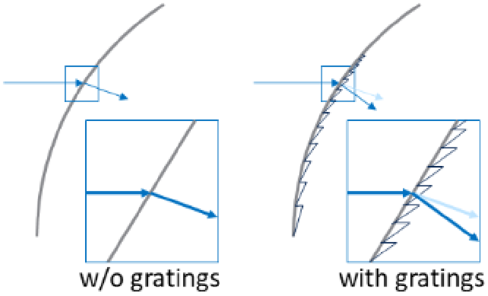


Figure 4.3 *Phase profile for diffractive lens* [102]

In the study [101], continuous diopter change from -1.4D to 1.4 at the wavelength of 532nm with a lateral shift ranging from -5 to 5mm was used. Limitations in the study include the approach to optimise the Alvarez lens for a single colour (green), but the researchers are exploring opportunities for a full-colour display. The lateral displacement in this study was also manually adjusted, therefore a future requirement would be to use electronics to control the lens displacement.

The Half-Dome 3 is a prototype varifocal HMD developed by Meta in 2018, relying on fixed liquid crystal lenses [103] with 64 focal planes that allow users to view depth information using eye tracking resulting in accurate focal point representation therefore accommodating for VAC issues. Fixed liquid crystal lenses allow for electronically controllable focal powers resulting in a small form factor for focal lens adjustment. They also eliminate noise, vibration, and any moving parts such as motors which improve ergonomics and reduce weight compared to a mechanical counterpart. Meta reality labs have advanced the prototype into a product referred to as Butterscotch. The Butterscotch prototype is limited to 50° field of view but provides a display system capable of 56PPD allowing for 20/20 visual acuity with a varifocal range from 0 to 4 diopters. More specific details regarding the design implications on human eye resolution can be found in Section 4.3.6. Further research into multifocal and lens technology has resulted in novel microlens arrays supporting multifocal capability with a high fill factor controllable via voltage [104]. With a high fill factor, distortion is prevented from the optical lens thereby preventing loss of resolution via regulating a single voltage signal. With such low power consumption, it's theorized that this is a viable solution allowing for integration into more portable headsets.

Figure 4.4 *Varifocal optics with mechanical and fixed liquid crystal lens on the right* [105]

4.3.2 Dynamic Distortion

All lenses introduce distortion, pre-warping algorithms, and static distortion compensation can help alleviate the issue however it still causes some users to experience motion sickness effects [106]. The "sweet spot" in virtual reality is found when the user adjusts the headset to allow their pupils to align with the lens and display. Lack of image sharpness, ghosting, and glare produced by the display occurs when the headset moves from the "sweet spot". Static compensation involves the user adjusting the headset to locate this placement of the HMD, software solutions are being explored into dynamic compensation to adjust frame rendering to "redress distortion in every frame." A white paper has described a potential method where high-resolution displays can use a dynamic distortion to better iterate on this phenomena in real-time based on the wearer's headset position [107].

A developed simulation system supports the prototyping of the distortion effect with high reproducibility and can improve iteration speed for lens design by presenting stereoscopic distortions on a high-speed screen with a shutter glasses, removing the need for physical optics to be required to simulate the effect. The paper [107] described a first-user study of perceptual requirements for an eye—tracked optical distortion correction solution.

4.3.3 Retinal Laser Scanning

There are alternatives to using a display to render environments. Advancements in near-eye-to-display technology have resulted in a variety of hardware progressions in recent years. One such advancement is retinal laser beams (RLB) also known as laser beam scanning (LBS) which don't use backlit flat pixel displays to render objects. Instead, it's more like that of a cathode ray tube where an electron beam scans across a phosphorate screen, in a retinal laser beam approach the laser beam scans rapidly across the retina to draw an image. Both techniques take advantage of the persistence of vision effects, producing the illusion of a continuous image. This temporal property of illumination [108] works in the real world where photons are continuously projected onto the retina, resulting in a "continuous real-world refresh rate" allowing the user to view moving objects and environments. The illumination period on traditional displays has hardware caveats such as being restricted to 144hz in the

Valve Index HMD – a high end VR system as of time of writing in 2023 - which also requires a low persistence display discussed in Section 4.2, which allows for natural motion blur to be exploited in the render pipeline during head motion in a HMD.

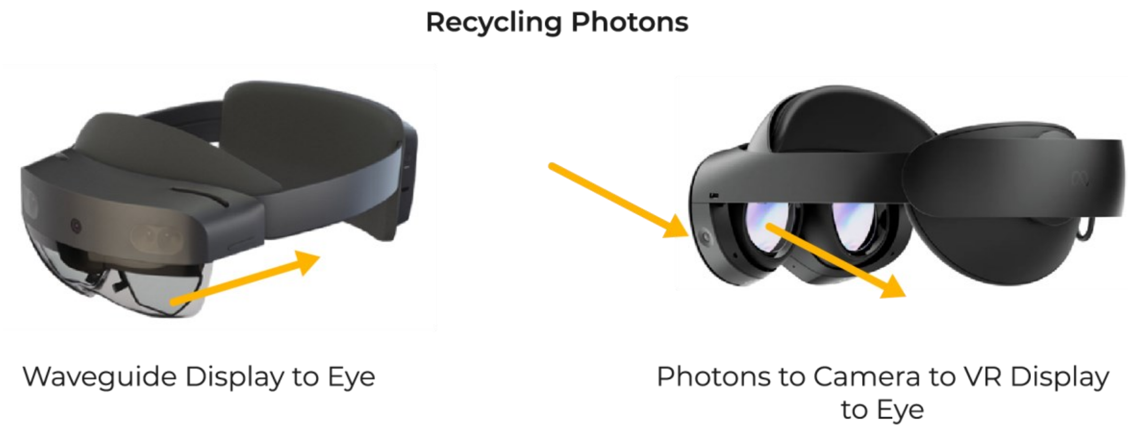
Retinal laser scanning, however, takes advantage of a continuous refresh rate to that of the real world or natural environments by using lasers as photon sources. The Avegant Glyph [109] released in 2016 was a commercially available product that used retinal laser scanning to provide a 65” 720p display for theatre entertainment, resulting in a screenless display. With only 40° of field of view (FoV) however, it demonstrates technical constraints with a RLB approach for displaying content. Limitations in the field of view arise due to the inherent design of the retina. The fovea has a small surface area inside the retina which is responsible for high-acuity vision. LBS therefore requires light photons to precisely target this light-sensitive area of the eye to provide the best visual experience in detail. This natural limitation of a small surface area of the fovea therefore presents a technical challenge that is constrained by nature, restricting high detail viewing to a narrow, small surface area.

Although the Avegant glyph provided a closed environment in the form of an HMD, it was more familiar to that of a VR HMD than an AR display, being that the real world is not visible when the headset is worn. The technology has since been directed towards AR use, providing a lightweight solution for AR glasses where a small FoV is not crucial for its use but perhaps a benefit. Due to compact RLB projectors developed, RLB has also been used in conjunction with waveguides to superimpose digital content onto the real-world environment. This problem space has recently resulted in the formation of the LaSAR alliance which is an IEEE initiative [110] to drive the manufacture of AR wearables allowing for increased collaboration amongst researchers interested in the compelling area that LBS provides. It is suggested in the presented discussion, that the field of LBS display technology is a highly desirable commercial problem resulting in many private institutions gaining significant funding for display technologies research.

Waveguides are designed to guide and manipulate light, allowing for light to be transported between points through a medium, known as optical folding. Fiber optic cables use waveguides to transmit information over large distances, for AR headsets the light is instead refracted across a glass or plastic screen to direct towards the user’s eye to provide a FoV for a digital image. The two main types of waveguides currently researched for AR displays are planar waveguides and volume holographic waveguides.

Planar waveguides are known as slab waveguides [111], where light is trapped by dielectric constants of the material being used resulting in light being confined to the middle layer by total internal reflection (TIR). When the optical component is used efficiently, light is confined within a flat surface. These constants use dielectric fibers to guide the light waves through the structure. Further research is still required in this area for RLB to be integrated into the VR HMD medium that we know today.

There does seem to be a potential integration for RLS into a VR HMD if 40° of view is used in conjunction with eye-tracked foveated rendering with a larger backlit LED screen to provide a peripheral view. Varjo HMDs use a small densely pixel-populated screen for the user’s center view with the surrounding peripheral vision taken by a lower-resolution display. This allows the user's main center cone of vision to be high resolution. A potential integration of RLS could use a similar hybrid approach to increase the clarity of the center image. However, due to the eye and gaze movement of the user, if they are not looking directly in the center where the RLS is being projected, then the back LCD panel of lower pixel density would be used and the RLS being disabled. It does therefore appear that this retinal projection technology when used with planar waveguides has more potential to support AR solutions in a glasses form factor vs a VR application. In a MR solution for projecting digital content onto the real world similar to AR, a MR HMD is used where photons are instead recycled from an external camera lens to a LCD or OLED display vs an RLS approach of being refracted onto a waveguide, as shown in Figure 4.5. Headsets such as the Quest Pro, Quest 3 and the Apple vision pro support MR solutions and VR applications at time of writing.



Waveguide Display to Eye

Photons to Camera to VR Display to Eye

***Figure 4.5** Difference between AR Hololens and MR Quest Pro in reference to photon recycling*

4.3.4 Holographic Display

As mentioned in Section 4.3.3 planar waveguides and holographic waveguides are two techniques to project an image to an AR display. The evolution of holographic displays has been primarily focused on the screen presenting 3D depth information allowing for natural gaze focus, preventing the VAC while still being able to view the real world.

Nvidia has presented an approach specifically for VR HMDs boasting a significantly reduced form factor from the lens to the optics with a wearable prototype weighing just 60g and the spatial light modulator (SLM) at 2.5mm thick. SLM is used in many display systems from adaptive optics to holography where spatial light distribution is controlled by applying an external electrical signal to the device, resulting in a change of optical properties to the material. The primary focus of this section is to provide an overview of SLM's potential role in VR displays and should be noted that there are various types of SLMs each with their use case HMD application. Including liquid crystal spatial light modulators (LC-SLM), deformable mirror spatial light modulators, and digital micromirror devices. Existing approaches to display technology for VR require a lens to enlarge an image of a small LCD micro-display. This requires the magnified principle which will result in a distance requirement between the lens and display which is a significant drawback for VR adoption as HMDs end up being bulky and cumbersome to wear [112]. Pancake lenses, however, fold the optical path multiple times to reduce the distance required between the lens and the display, whereas fresnel lenses would be otherwise used, requiring this distance between the display and the lens.

Similar to the retinal laser scanning approach, a projection design method is used. This results in a low FoV for applicable use with VR with similar constraints, with the wearable prototype only allowing for a 22.8 degree diagonal FoV. Previous implementations using a similar approach allowed for 4k SLMs to be used with 16.8° FoV by tiling the SLM [113]. It should be noted however that the implementation of 4K SLMs was completed with a benchtop prototype. The significant leap in visual fidelity in Nvidia's paper [112] was built upon the integration of artificial intelligence to accelerate the computation of the generated holograms. In addition, a pupil-high-order gradient descent algorithm is used to calculate the correct phase of light to present to the user based on their varying pupil size.

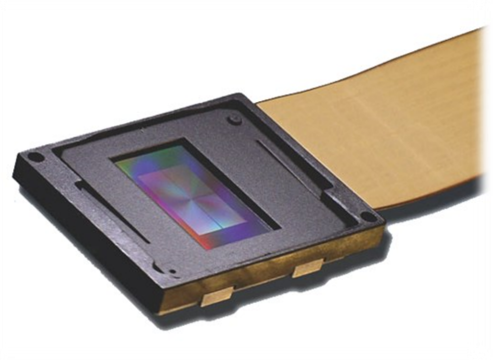


Figure 4.6 *Holographic SLM by Onset* [114] *demonstrating its small form factor*

4.3.5 Light Fields

Various capture technologies have been developed to improve the fidelity of scanning real-world environments evolving from photogrammetry to point cloud LiDAR scanning. These technologies are discussed in detail in Chapter 5. Another promising technique to capture high-detailed real scenes for representation in VR is light fields. Google’s 2018 paper [115], describes how light field rendering specifically can provide an immersive environment with “unsurpassed levels of realism” allowing for stereo parallax, motion parallax, reflections, refractions, and volumetric effects for real-world scenes to be accurately represented. If a light ray data is missing, such as in between cameras, view interpolation techniques are used, such as depth maps to generate geometry or synthesize new images to fill in the gaps.

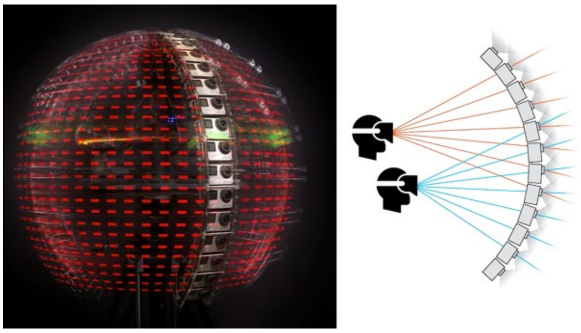


Figure 4.7 *Light Field Rendering graphic from Google* [115]

The technology was initially discussed in 1996 in a paper titled "Light Field Rendering" [116] allowing for arbitrary views of camera angles without depth information to produce detailed perspectives after processing, known traditionally as image-based rendering. Rather than using geometry to render 3D assets into environments, the technique interprets the 2D input images as slides of a 4D function which is the light field, allowing light rays to flow unobstructed in a static 3D scene. This allows for accurate light representation where a vector function is used to describe the collection of the light in space [117].

The plenoptic function describes the intensity (radiance) of all the light rays in a 3D region of space, which may be parameterized by coordinates (x,y,z) and a direction (two angles). This parametrization is the basis of 5D plenoptic modeling or image-based rendering. The 5D model is instead described as light fields as a set of panoramic images captured from different positions in space shown in Figure 4.7 above [118].

However, the limiting factor in this is if the user adjusts their position by rotating their head, they lose the stereoscopic effect. Light field technology has had a series of technical challenges to overcome to produce a suitable render pipeline to acquire, process, and render. Firstly, the parameterization and representation of the light field. Secondly, how we can generate a light field from capture methods (primarily camera). Thirdly the generation of different perspectives to apply the stereoscopic effect desired. Another key problem discussed in the paper [118] is the amount of data required. These issues have strong similarities to rendering in photogrammetry pipelines, such as capturing objects, digital rendering, and then output to a display.

Advancements in rendering pipelines since 1996 have led to proprietary light field displays for both AR and VR [119]. The progression has led to prototype HMDs using integrated light field displays, boasting 240Hz resolution with a continuous depth resolution, and 100° of angular resolution towards infinity [120]. For comparison, the Valve index has a maximum refresh rate of 144Hz in 2023.

Promising solutions that light field display technology can solve include the VAC discussed in Section 4.3.1 [121], and support for a digital prescription solution, removing the requirement for users to append a custom prescription lens into their HMD. In addition, there are displays available on the consumer marketed using light field technology allowing for numerous people to view a volumetric image, showcasing 3D models in a picture frame format all without an HMD being worn [122].

To conclude on light field display use cases, project Flamera is demonstrated to use light field technology to provide passthrough support for a prototype VR HMD, showing another advantage of light field processing and displays [123]. Providing passthrough support is critical for allowing AR/mixed reality experiences using a VR HMD based on camera tracking. The main research issue with mixed reality passthrough is replicating the processed camera frames onto the user's HMD display while not distorting the image or adding significant artifacts to the output. Rendered frames are currently distorted as HMD cameras are not necessarily at the front exactly where the user's eyes are and therefore require reprojection distortion algorithms to blend processed frames.

4.3.6 Highlighting Human Eye Restrictions

The following section aims to outline further considerations for end users using an HMD concerning focal distance and the constraints bound by the human eye, such as the requirement for glasses for some users. More specifically a diopter (D) is a unit of measurement used to express the optical power of a lens or the refractive error of an eye. It quantifies the ability of a lens to converge or diverge light rays. In the context of lenses, it represents the strength of the lens in focusing light. Specifically, one diopter (1 D) represents a lens with a focal length of one meter (approximately 39.37 inches). Unsurprisingly, the HMD design uses **fixed** focal distances for VR headsets in 2023 and is on average 1.3m.

A lens with a higher diopter value has more optical power and can focus light more strongly, while a lens with a lower diopter value has less optical power and focuses light weaker. For eyeglasses, a positive diopter value is used to correct for farsightedness (hyperopia). A positive lens with a certain diopter value helps to converge light rays and bring the focus point onto the retina, allowing people with farsightedness to see nearby objects more clearly. Conversely, a negative diopter value is used to correct for nearsightedness (myopia). A negative lens with a specific diopter value helps to diverge light rays before they reach the eye’s natural focal point, allowing people with myopia to see distant objects more clearly.

As discussed in Section 4.3.1, the VAC is a constraint in the design of fixed focal distant lenses which can be solved with the use of varifocal lenses, however, due to the diopter value of lenses, it is an additional requirement to include users' optical prescription during the manufacture of fixed VR lens displays. It is because of this, that a software prescription would accommodate users' near and far-sightedness. Such solutions are only theoretical or being researched in VR labs currently, but plausible solutions are the use of holographic displays to allow for varifocal lenses with a prescription. Thus, allowing eyes to focus on objects at different depths, while simultaneously supporting the user's prescription lens which would impact their perception of detail. The following sections highlight potential solutions and the cross-over between the latest VR display research and show a path for the numerous technologies that must be combined into a single display lens solution to seamlessly accommodate users' perception when being presented with a 3D object on a display.

4.4 Holistic Interaction Optimisation

With the above discussions on presenting a high-fidelity visual experience through an HMD, further exploration should be made to explore how users physically interact with the virtual world and strive to cater for natural interaction. Although traditional ray-based solutions allow for engagement with 2D interfaces and selection of 3D objects in the environment, modular interfaces discussed in Chapter 2 allow the user to engage with virtual objects similar to that of the real world. Various algorithms and optimisation processes in the VR pipeline that have been discussed in previous sections are integrated to hit metrics that prevent locomotion sickness and to convince the user's visual system that they are present in a seamless digital space.

The outlined section aims to explore solutions found that can be described as accommodating for **physics-based interaction using algorithms**. The discussion should be implemented if developers wish users to directly interact with a physics engine, such as rigid bodies. Traditionally, user interaction with a physics engine can result in unexpected behavior due to the speed at which users move compared to the physics engine's tick rate and floating-point errors. These errors cannot be eliminated due to representing real numbers in a fixed space but only be mitigated. The issues associated with humans interacting with a physics engine are discussed and solutions to how developers can accommodate for these interactions.

4.4.1 Accommodating Rigid-Body Interaction

VR interaction with rigid bodies can result in a significant amount of error caused by floating-point evaluation. In a physics engine environment, it’s common to normalize or filter error values created. The alternative is to avoid using a physics engine for interaction entirely, and instead apply controllable transform updates to objects every frame. For example, in Chapter 1 we use a low pass filter to normalize user input.

Another approach to accommodate input during VR interaction, particularly complementing 1:1 hand tracking in physics-based input, involves allowing developers to predefine hand poses for specific virtual objects. Such a system lets users' hands "snap" to a predefined position or orientation when they come in proximity to certain virtual elements, such as life-sized switches or buttons. This introduces an element of leniency in their interaction, mitigating the need for pinpoint accuracy when engaging with virtual physics objects. This can be likened to a 'magnetic' effect, where the virtual interface aids the user in ensuring the correct interaction posture.

In the physics engine, as the user travels further away from the origin of the world environment, they will run into floating point errors. One simple solution to this is to keep the player's perspective at 0,0,0 and move the world around the player, loading and unloading assets as required. Another solution is to introduce loading screens to move the player back to the origin and load the new environment around them. More commonly for fixed-sized environments, we instead introduce error mitigation techniques as simulated interactions between rigid bodies become more complex.

As discussed in Chapter 2, if developers are going to use the physics engine to power interaction, optimisation needs to be explored to handle unexpected results caused by the core physics engine loop. To counter the accumulation of floating-point errors over time, the physics engine often will employ error correction or error reduction techniques over time to compensate for accumulated errors, preventing artifacts from being shown to the user. Continuous collision detection is a technique used to handle fast-moving rigid bodies that have the potential to clip through other objects in the game. By interpolating between frames, we can resolve collisions more accurately by breaking down the collision slowly. It's for this reason that many physics engines run on a bound framerate, for example, the Unity fixed update method is capped at 50 frames a second. Having uncapped framerate-bound physics engines can lead to artifacts such as time scaling issues.

Interoperability is a challenge facing VR where large-scale multiplayer worlds need to be rendered, although tile mapping scenes together to create an experience can be a solution, the user's vector3 position will have to be clipped to prevent floating point errors if they travel far from the origin. Other areas of the issue include joint stabilization where floating-point errors occur where hinges, springs, or ropes are used as they are rigid-body components interacting with other rigid-body components. Even with error mitigation techniques completed by the core physics engine, unexpected results still occur due to the vast amount of collision types and the processing tick rate variability between interactions.

A spring joint was used for the development of a VR 3D slider found in Figure 3.4, allowing a level of forgiveness for interaction as it responds to the user's feedback in a realistic way to that of a real-world spring. Due to the accumulation of small floating-point errors, stabilization techniques that adjust the force applied to joint or correct objects are applied to prevent excessive vibrations. In the design of the slider system, we have included constraints to lock the y and z axis resulting in the slider only moving between 0 and 1 positions on the x-axis. From this, we can use a rigid body collider attached to the user's virtual hands to manipulate the position. However, there are still unexpected errors such as the spring moving large distances on small movements, these interaction types are discussed and further mitigated through the techniques below.

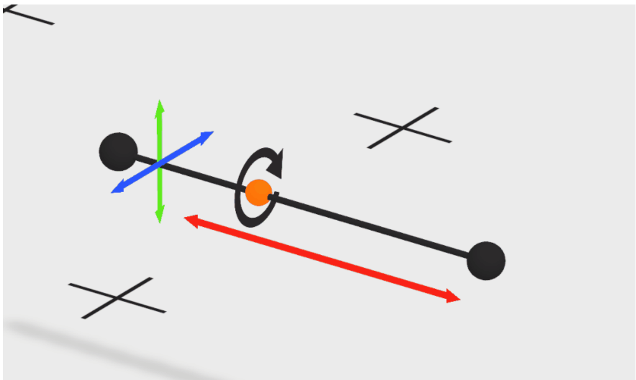


Figure 4.8 *Demonstrating all axes available for a 3D slider on a spring joint*

During user interaction with VR controllers, we may have unfiltered raw input from the user interacting with objects. By applying human movement from the real world as raw input to the physics engine environment, we run into even more physics engine errors. There are mitigation techniques at the low level of physics engines that help alleviate and correct issues that arise. These are traditionally included in all physics engines but highlight problem areas when objects interact in a physics engine.

4.4.1.1 Normalized Impulse Clipping

During collision forces are exerted on each object, usually represented by impulses such as vector3.AddForce(0,1,0) in a time bound function. To prevent unrealistic behavior or instability in the interaction, the magnitude of the impulses across the two objects is clipped or limited, resulting in an impulse within a reasonable range.

4.4.1.2 Normalized Penetration Resolution

If however, objects do penetrate each other there is a resolution technique to separate them again. In high-velocity situations, objects may penetrate each other's collider and result in the physics engine applying forces that are too high. To resolve the intersection or overlap, penetration resolution will detect the penetration depth between the colliders and the direction of the force. By applying corrective forces or displacing the force to the intersecting objects we can separate them. Normalization can again be applied to the value to ensure the forces or corrective displacement value is within reasonable bounds.

4.4.1.3 Normalized Collision Normal Calculation

Following from penetration resolution, during the collision the contact normal which represents the direction of the collision will need to be calculated to apply any form of corrective displacement to the object's positions. Normalization is therefore applied to ensure that the collision normal are unit vectors to maintain object position accuracy.

4.4.2 Proportional Integral Derivative

Controls system theory works with polynomials, zeros, poles, time domain frequency, and state space. To generalize, in control systems theory we take an existing value and aim to apply control values to converge the existing value to the desired value by introducing a control-loop mechanism. Common use cases in game development include character movement speed in 3D point-to-click games, camera position tracking, AI behavior to a node, or gameplay mechanics such as projectile systems. Control systems theory is also used across industries such as industrial control systems that require modulated control. The focus of this section is to introduce control systems theory to mitigate issues caused by human VR interaction with a game physics engine.

In a virtual reality environment, **using control theory throughout interaction systems can support users' interaction with virtual objects' physical properties** such as weight and their rigid-body colliders. With 1:1 body and controller tracking from the real world translating to the virtual physics engine, there are scenarios where a disconnect between users' real-world movement and their virtual counterpart can result in unexpected behavior.

To expand on this, tracking controllers in the real world connect to users' virtual hands which support rigid body collision. The result allows for collision with virtual rigid-body objects. For example, a user moves their virtual hand onto the surface of a virtual table. In the real world, a user's hand can pass through unobstructed space freely, but in virtual space, their virtual hand should stop on the table's surface as it has collided with it.

For an immersive VR experience, we must therefore consider the design implications that enable users' virtual hands to not clip through virtual rigid bodies. Such tracking mechanics work well for climbing scenarios such as ladders where users defy gravity and throw or climb their way through an environment [124].

Many control systems can be used in the above-described scenario. One such is the use of a Proportional-Integral-Derivative controller (PID). In this discussion, we break down a scenario using rigid body controller tracking demonstrating its importance in its integration for physics-bound object interactions [125]. The remainder of this section will break down the PID controller works to support its integration for VR applications.

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}$$

Equation 4.1 *PID Formula*

To highlight, PID control theory is a control strategy that can be used to adjust the position or rotation of a rigid body object moving or rotating to a setpoint. As the rigid body moves closer to the setpoint, it may require adjustment as the setpoint moves location. The advantage of using a PID is there are only three key parameters to adjust over time to get a result, allowing for ease of implementation in code for a variety of game development applications.

$$e(t) = desired\ value - actual\ value$$

Equation 4.2 *Error Value*

In this, $e(t)$ is the error and is the difference between the desired and actual value, where we strive for the error to become 0. Therefore, by applying physics forces such as torque to the rigid body object, we can trend $e(t)$ towards 0.

Applying torque to the body as the control value nears 0 has implications. As it is driven by the physics engine getting exactly to 0 is difficult due to floating point error. Applying torque that is proportional to the error value will result in the object moving toward the desired destination. When the error $e(t)$ is large we apply more torque and where the error is small, less torque. We therefore can turn this into a function.

$$torque(t) = AngularInertia * f(e(t))$$

Equation 4.3 *Angular Inertia*

When the error is getting closer to 0 the corrective torque being applied may be too significant at smaller and smaller distances/angles and overshoot the desired position. Resulting in more torque being applied as the error value has increased again, thus being stuck in a scenario where the object gets close to its destination before swinging out away from the desired point and eventually resulting in an exponentially decaying oscillating effect (sinusoid) at which point the system has stabilized if the value of Kp is not too large. Kp is a function of the error, if it is reduced so the target isn’t moving fast, then the torque is too low to compensate in proportion to the object's speed.

Therefore, we need to expand further and introduce proportional feedback. Known as a steady state error which ensures the control output is different from the input, i.e. if the distance is 0 then the error should be 0 but due to the above reasons it may not be.

$$SteadyStateError = [1 + constant * Kp]\ desired$$

Equation 4.4 *Steady state error*

Introducing integral feedback into the formula will consider the time of the object's movement or direction. If the error is constant over time, then the integral will counter the movement if the target suddenly changes its set position over a small distance, *each time step* will adjust the torque to compensate. By increasing Kp the bias is smaller, but oscillation still exists.

We take values over time to adjust $e(t)$ as the rigid body object is moved to the set position. The longer the value integrates and collects, the more torque is applied over time. The integral over the entire timeframe is not calculated due to the large distances that it may gather which isn't relevant in the short term. Instead, calculating the sum of $e(t)$ over a shorter period and multiplying by a "time step" is a first-order approach (such as the Euler Method). PID is of the first order, meaning the local error per step is proportional to the step size squared, and the error at a given time is directly proportional to the step size.

As we still have oscillations as we get closer to $e(t) = 0$, we need to dampen oscillations as we near the set position. By analyzing the derivative of $e(t)$ over time, we can apply a force to drive the value towards not changing *or* increasing from the set position. This is in the form of a counter-torque to the proportional and integral components discussed.

The derivative is largest as the rigid body object swings through the desired set position. Therefore, applying counter torque as we get closer to the set position will counter the oscillation and reduce the derivative as we move further away from the set position where the $e(t) = 0$. The derivative can be calculated the same way we calculate the integral by taking the differences over the last short-term values.

The above PID controller can be converted to pseudocode as follows with a C# implementation of a PID controller found in Appendix C.

```
Control Output = Kp * (error + (1 / Ti) * integral + Td *
derivative)
```

Where Kp is the proportional gain, which determines the strength of the proportional term. Error is the difference between the desired setpoint and the current process variable. Ti is the integral time constant. (1 / Ki) determines the speed of the integral term's response. Integral is the accumulated sum of errors over time. Td is the derivative time constant. (Kd) which determines the speed of the derivative term's response. Derivative is the rate of change of the error over time.

In a given scenario we track users' hands to their virtual counterparts and can prevent a situation where a rigid body hand passes through other rigid body surfaces. To discuss, the user’s real-world hands may have surpassed the virtual table resulting in their hands no longer tracking 1:1 but instead, the virtual hands are now on the table to the intersection point which is the set point as described. This does have the potential to be disorientating, however without physical force feedback to the user to restrict their arm or hand movement, we need to calculate the distance between the user's virtual rigid body hands and physical hands by using the controller transform position. In this scenario, should the user's physically tracked hands pass through the table surface, the difference between the tracked hands position and virtual hands position would require a PID controller to align the virtual hands to the table surface intersection. Allowing for a now-*controlled* interaction with the physics engine. We can then apply physics properties such as surface friction to the virtual surface.

For comparison with a previously discussed interaction algorithm in Chapter 1, the 1 Euro Filter aims to reduce noise and jitter in positional tracking data. Whereas the PID controller mechanism adjusts the output based on the difference between the desired setpoint and the measured variable.

4.4.3 Hooke’s Law Integration

Integration of Hooke’s law in combination with using a set point with PID, we can create a climbing locomotion system where force is calculated based on the distance between the user's virtual rigid body hands and their actual hands. Where the user's virtual hands are on a virtual surface and the user's tracked hands have “gone through the surface”. Hooke’s law is primarily used for spring joints and is therefore applicable for such a locomotion mechanic amongst other forgiving interaction approaches, such as the designed slider in Figure 3.4 and Figure 4.8.

The calculated force – based on the difference in distance - is then directly applied to the player's rigid body, providing spring-like movement. The subtraction in Equation 4.4 shows the force is being directed opposite to the displacement to invert the applied force. In VR locomotion, the user’s controller position is the displacement vector based on the velocity and angle they release/throw/project from.

$$\mathrm{F} = -\mathrm{kx}$$

Equation 4.5 *Hooke’s Law*

- F represents the force applied to the material.
- k is the spring constant or stiffness of the material, which measures how resistant it is to deformation. It is a material-specific constant.
- x denotes the displacement or deformation of the material from its original, equilibrium position.

4.4.4 Alternative Climbing Locomotion

By not including PID we are instead left with a vector position of the player's virtual hand position 1:1 to their tracked hands position. In climbing locomotion scenarios, such as climbing a virtual ladder, we may consider allowing the controllers to disregard or remove rigid body collisions with a surface to prevent unexpected behavior and instead just calculate a velocity to apply to the player's rigid body.

Examples of a user input to apply velocity to a VR player controller – or rig - include the user's arms swinging, where the force being applied to the players VR controller can be calculated using Hooke's law. Elements of constraint must still be considered in such a scenario, where high usability satisfaction for the end user should be considered. Making accurate physics interactions for all objects may lead to frustration as the user desires to move around the virtual space naturally. Considering this, in a ladder-climbing scenario, the user's controllers may pass through the ladder steps as long as their controls are inside the collision mesh area.

In addition, another problem faced by climbing locomotion strategies includes locomotion sickness from unexpected behavior when simply moving off obstacles such as ladders onto a flat surface. Animations can seem jarring if the user is trying to get off a ledge at the top of a ladder, in which an animation is triggered, and they are instantly “carried” or teleported to the ledge. Another example is when users are climbing a wall, and their rigid-body character controller can't get on top of the wall due to colliding with their torso. A solution to this is as the user's hands are on the wall, we cast a ray down from their head. If the ray intersects with the top of the wall's surface, then we forgive the player and move – via a lerp function or similar - the rigid body directly below the head collider resulting in the player's controller on top of the wall.

4.4.5 Drag

To account for hand velocity speed during such a climbing locomotion scenario, a drag can also be applied to character hands. This allows fast movement to be registered while allowing for fine control via slower movements, resulting in a more natural/intuitive locomotion system when introduced. This rewards the user's fast velocity movements, by applying the velocity to the rigid body controller. Such a velocity can be applied using the AddForce() function in Unity. A reoccurring theme in VR interaction design - as discussed in the 1-Euro Filter algorithm – is where users' actions can be adjusted based on velocity movement, supporting user control during fast and slow movement speeds.

We apply more drag when the user's controllers are moving slowly, and less drag when the user's controllers are being moved fast.

To introduce drag into a locomotion system we can use the below pseudocode where a max value of 0.3 is used. In physics simulations, the drag coefficient is typically between 0 and 1, where 0 represents no resistance (no drag) and 1 represents maximum resistance (full drag). In the example, setting the value to 0.3 as a maximum value, we can decide how fast we want the player to move in their presented scenario. In Section 4.4.3, we discuss Hooke's law. This formula is included in the below pseudocode but also does share similarities in isolation. However, by combining drag and Hooke's law, we can expose more parameters to control the user's velocity based on the variety of physics simulation environments possible, such as air drag and water drag.

Sections 4.4.3 to Section 4.4.5 can be converted into the following pseudocode solution to apply force based on hand movement velocity, this value can also be clamped to prevent unexpected behavior with user's rigid body motion.

```
velocityMagnitude = playerRigidbody.GetVelocityMagnitude()
drag = 1 / velocityMagnitude + 0.0
drag = max(drag, 0.3)
force = -playerRigidbodyVelocity * multiplier * drag
playerRigidbody.AddForce(force)
```

4.5 Conclusion

In this Chapter, we have explored the low-level optimisation steps required to process every frame in the VR render pipeline. Due to computationally complex requirements for rendering a high-fidelity experience, we must exploit weaknesses in the human visual system to reduce the amount of computational processing required.

Discussion on the future directions for display and lens technologies are made, including novel insights into frameless eye-tracked foveated rendering (FETFR) that shows promise with new tracking hardware such as eye-tracking being made available in VR HMD's, with a preliminary overview of its implementation using the Vulkan graphics API outlined in Appendix B.

The discussion extends beyond visual stimuli optimisations from Section 4.4. into areas where algorithms can enhance user experience (UX) by reducing artifacts during physical interaction. Notably, while optimisations to prevent motion sickness are well understood related to the visual aspect of VR, there is a significant gap in the literature regarding user motion and interaction optimisation. This chapter addresses this gap by proposing the integration of control systems and filters into the interaction layer of the VR pipeline. With demonstration of how to integrate the PID controller into a physics-based locomotion technique – climbing.

Floating point errors in the physics engine can lead to unexpected behavior in virtual environments during physics-based object interaction, resulting in artifacts such as incorrect item placement or sporadic movement. To mitigate this, reflection on the VR render pipeline is made and how it incorporates LOD algorithms such as variable refresh rates and asynchronous reprojection techniques that trick the human optical system into reducing motion blur. Drawing inspiration from these visual optimisations, we can implement control systems theory into the interaction layer, allowing users to interact with the physics engine predictably and in real-time. By mitigating effects produced due to real world movement interacting with a game engines physics engine, it aims to provide an enhanced immersive experience ensuring consistent and predictable interactions with 3D objects and physics-based locomotion techniques.

By recognising and addressing the need for both visual and interaction optimisations in VR, we can further enhance the fidelity of immersive experiences, bridging the gap between the user's physical interaction and the virtual interaction taking consideration into interface solutions outlined in Chapters 2 and 3.

Chapter 5

3D Data Processing and Visualisation

As photogrammetry, point cloud, and stereoscopic 360-degree cameras become more accessible to the public, a newfound challenge emerges. How to best visualise these formats in the evolving mediums of VR and AR.

The following chapter presents three case studies focusing on reducing barriers to accessing visualisation of generated multidimensional datasets in virtual and augmented space. Considering both the data gathering approaches that can be used with advanced sensor arrays such as LiDAR and approaches visualising the dataset in an accessible medium, mobile phone AR. The case studies aim to provide applications of previous chapters' work through open-source visualisation platforms for use in charity organisations, research, and industry.

Traditionally, visualisation formats vary based on the sensor array being used to capture real-world objects for digitization and viewing – defined as digital twinning. However, in recent years photogrammetry and the use of AI optimisation have allowed users to record a variety of content from vision-exclusive sensors (digital cameras) and process photos or frames into a 3D model or a point cloud with high precision. The following studies aim to exploit and discuss consumer-grade technology that can be used for processing and displaying a variety of data formats in a cost-effective VR and AR solution.

Despite the apparent differences between VR and AR, there is significant overlap in their underlying technologies, content production pipelines and optimisation considerations. Mixed-reality HMD solutions use a VR headset to augment digital objects onto real world space, much like AR augmenting onto reality, again highlighting its similarity. Where photons are recycled in VR and MR HMD's but not AR headsets, as discussed in Section 4.3.3. By examining the presented case studies, we also focus on mobile phone AR, emphasising the shared challenges and development solutions applicable to across VR, MR and AR. Thus, reinforcing the thesis's focus on immersive environments as a whole.

The first case study explores how using a 360-degree camera to record real-world environments can be used for interdisciplinary VR studies by integrating text and audio cues at specific time intervals into a developed open-source solution [4]. This approach enhances immersion using stereoscopic video and provides a more comprehensive understanding of the environment being studied and is available as a compiled solution or codebase to modify freely [126].

The second case study discusses the development and challenges associated with creating an end-to-end solution for scanning and representing 3D scans or digital twins of museum artifacts [3]. This solution, in collaboration with a cultural heritage museum, is designed for representation through a browser or mobile device, utilizing the camera to create an AR experience. The study emphasizes the importance of accessibility and interactivity in presenting 3D models for users, ensuring that everyone with a mobile device can engage with digital replicas of artifacts seamlessly without requiring a VR HMD.

The third study investigates how digital twinning sensors can be adapted for use with drones to scan large areas of land for subsequent analysis. This approach enables the creation of high-fidelity 3D visualisations of extensive landscapes, providing valuable data for various fields such as environmental monitoring, agriculture, and urban planning. Discussion is made on the file formats used and how a point-cloud renderer solution can be made for representation in the Unity game engine.

These three studies highlight development pipelines and bespoke solutions developed and now available as open source for 360-degree stereoscopic video data representation in VR, 3D model representation for browsers and mobile devices, and high-fidelity scanning of large environments using UAS technology. By integrating traditional digital twinning techniques through photogrammetry with modern technological advancements such as AI, we demonstrate a cohesive framework that shows the potential of digital twinning mediums across different academic research and industry use case scenarios. This exploration includes an environment scan with video, an artifact or item scan, and a larger 3D visualisation of areas of land, establishing a comprehensive approach to digital twinning.

5.1 360-Video Toolkit for Interdisciplinary VR Studies

VR is expected to play a significant role in the transformation of education and psychological studies. The possibilities for its application as a visual research method can be enhanced as established frameworks and toolkits are made more available to users, not just developers, advocates, and technical academics, enhancing its controlled study impact. With an accessible first design approach, we can overcome accessibility constraints and tap into new research potential for non-expert users to design and complete immersive studies. The open-sourced toolkit demonstrates how game engine technologies can be utilized to immerse participants in a 360-video environment with curated text displayed at pre-set intervals. Allowing researchers to guide participants through virtual experiences intuitively through a desktop application while the study unfolds in the user's VR headset.

With the adoption of virtual reality becoming increasingly consumer-focused, researchers must decide which VR headset meets their study requirements, such as tethered vs wireless and online vs offline support. Recent meta-reviews accessing more than 53 systematic reviews and meta-analyses support its use in anxiety disorder, pain management, and weight-related disorders showing its effect in clinical psychology [127]. Some headsets require user logins and a requirement for both the participant and researcher to be in the same digital VR space. This proves difficult when documenting results, whilst also guiding the participant through the experience. With these considerations in mind of the researcher, how will the developed study be conducted at locations where headsets require an internet connection? What applications can be used?

AltspaceVR, Horizon Worlds, and VRChat are third-party software programs that have been previously used to design VR studies [128]. These are social applications that can also be used to help researchers plan their studies utilizing environment design tools. However, the capability of these applications is limited. This affects research design, and participant satisfaction and results in unforeseen technical issues.

The developed toolkit [129] could be a potential solution to this issue by providing an open source, intuitive, and accessible interface to carry out visual studies. With a study in progress using its features for memory recall [130]. It is built using the Unity game engine and provides basic immersive interaction with 6-DoF controllers and Visualisation of video and text. Technical knowledge of the game engine is therefore not required to use the compiled tool. Functionality can be addressed and added through the Unity Engine if desired, but for researchers who want to place a participant in an immersive environment with timed text and audio cues, the designed tool provides them with this functionality.

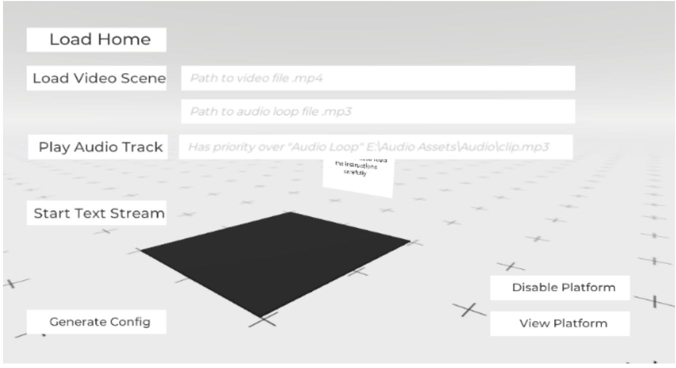


Figure 5.1 *Main home interface displayed on the desktop to guide the session.*

5.1.1 Accessibility

Commonly, crucial evaluations of interaction tactics are made at the discretion of the developer or researcher during the design of the environment, rather than through thorough end-user testing. The virtual reality (VR) sector still lacks a well-defined set of accessible design concepts to adhere to. A more efficient interaction system can be supported by a collaborative reduction of design options. Due to a reduction of features, first-time VR participants can have a reduced adjustment period without the pre-requisite of understanding controller features, or how to interact with specific components.

When using VR to conduct an immersive study, it is important to think about user accessibility. Headsets are traditionally heavy and wired, require physical movement to work with, and are difficult for new users to operate. During a 360-degree experience, user head movement may be required while reading text. Focusing user attention in VR is problematic due to the 360-degree play space, thus a solution is to use graphical cues such as arrows to direct the user toward the area of interest. This form of spatial focus is known as reference frames. Similarly, without additional instruction, locating the source of in-world audio can be difficult. Spatial sound will provide an omnidirectional auditory experience that will acoustically focus the user's attention in the desired direction.

However, persons hard of hearing may not be able to benefit from directional audio must also be considered. As a result, W3 [131] recommends that users have access to a mono audio sound option. Another method for directing user attention to the appropriate area is to utilize spatial arrows that point the user to where relevant information may be shown. Using the haptic rumbling in the motion controller to steer the participant to an area of interest is an alternate technique to draw attention to specific in-world areas of interest. These approaches have been considered in this solution.

5.1.2 Visual Elicitation

Having a controlled reproducible environment is advantageous for psychological studies. VR visual-elicitation studies could help improve our understanding of environments including users' emotional responses to a presented situation. Visual elicitation is the method of conducting an interview with participants responding to visuals exhibited, which is traditionally done using a 2D image or video. They would describe the image's societal, personal, and value implications. Meanings conveyed by the participant from the images supplement the verbal discussion in such a way that without images different emotions may be described [32].

5.1.3 Technical Discussion

Used in this toolkit, the Unity XR plugin used to interface with the OpenXR framework is now becoming a widely adopted standard to support a one-build solution for compatibility with VR and AR headsets.

The architected solution supports streaming multiple audio files simultaneously, displaying text at timed intervals, and allowing for 360 video playback. Due to the streaming mechanism using web requests, it is also possible to stream media to the environment with an internet connection. This option has been made available as 360-video files can have large file sizes. Building on this foundation of web requests, the application could be further enhanced by supporting more media types such as 3D model streaming.

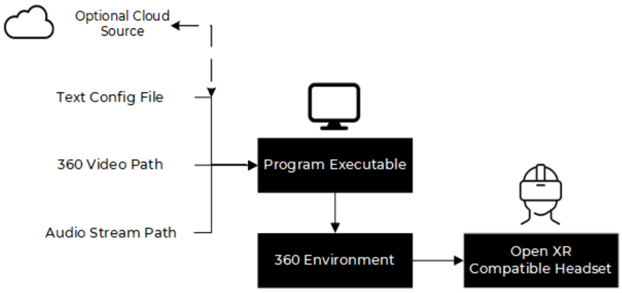


Figure 5.2 *Data flow diagram from input files to VR headset*

By allowing the researcher to share the participant's perspective on a monitor, accurate guidance can be provided to the participant whilst also being able to update scenarios as desired. A text output is displayed front and center to the play space's default position. A string and specified time interval can be sequentially read and displayed in VR from a configuration file. Another option is to play back words from a large body of text or book. It has been discussed if cognitive processes related to reading traditionally are complemented and reinforced by the tactile stimuli of the experience vs their digital counterpart [132]. This digital approach allows for audio, visual, and haptic feedback if desired to complement reading.

5.1.4 Analytics

Metrics are not exclusively recorded by the researcher. Qualitative analysis of user movements can be recorded using the HMD orientation in relation to the text being displayed. This approach can also facilitate identifying areas of interest without the use of an eye tracker, but rather by approximating the user's gaze using the area they're facing in the 360 video. There is an opportunity to evaluate more than just the user's gaze on areas of the virtual environment. By fusing recorded user motion data with their biometric data such as the user's pulse, we can evaluate the user's emotional response to a presented scenario with the capability to adjust the experience at runtime. This idea is not integrated into the deployed platform currently but is being considered. User body tracking provides another big data avenue to understand how users move in an immersive world, allowing for the recording of user motion and playback for further analysis.

5.1.5 Conclusion

VR can support guided experiences for studies in healthcare to psychology with perception-based tasks. We searched for current design solutions that enable researchers to quickly develop text-based immersive studies and found limitations in guiding and directing a designed experience.

With increased research into accessible tooling, we can alleviate physical and cognitive accessibility issues for both researcher and participants, allowing for accurate, fresh research possibilities across scientific fields. Traditional photo-elicitation and walk-through interview procedures have failed to yield rich replies in previous photo-elicitation VR studies, however, with further experiments and advances in accessible design, this may no longer be the case [133]. By supporting best practices out of the box and focusing on accessible first-design approaches, researchers from many fields can design and run a VR study effectively.

5.2 Augmenting Heritage: An AR Solution for Artifact Preservation on the Web

AI NeRF [134] and Gaussian splatting [135] algorithms, capable of cloud processing, have significantly reduced hardware requirements and processing efficiency in photogrammetry pipelines. This accessibility has unlocked the potential for museums, charities, and cultural heritage sites worldwide to leverage mobile devices for artifact scanning and processing. However, the adoption of augmented reality platforms often necessitates the installation of proprietary applications on users' mobile devices, which adds complexity to development and limits global availability. This case study demonstrates a cost-effective pipeline for visualizing scanned museum artifacts using mobile augmented reality, leveraging an open-source embedded solution on a website.

The loss of the additional dimension when visualizing 3D scanned objects on a 2D screen can lead to a reduction of user insight and engagement when compared to physically interacting or viewing museum artifacts. Many museums hold collections that are not available for interaction due to being behind a display case or aren't available due to a lack of exhibition space.

This section presents a cost-effective pipeline for adopting 3D photogrammetry scans into a browser-based solution allowing for display on both mobile and desktop devices using Google's 3D Model viewer framework supporting Augmented Reality (AR) [136]. Due to the low-cost options available of high-fidelity scanning technology today, there has been mass adoption of digital twinning of cultural heritage sites and museum artifacts for preservation [137], allowing future generations to experience historically rich sites that may have since been damaged or lost.

By 3D scanning artifacts using photogrammetry or LiDAR-based approaches, we can accurately reproduce a digital twin for an artifact or historical site for use in visualisation and interactive applications. Although scanning technologies have improved, providing an engaging and accessible presentation environment for scanned models has been lagging.

Mobile augmented reality allows users with smartphone devices to augment information in their real-world environment. In addition to overlaying information such as text, we can present access to the third dimension for artifacts that are traditionally behind glass protection in museum exhibits allowing visitors to explore all its perspectives.

Without an accessible and always available database of models, scanned objects may not be fully exploited by members of the public and research community. In addition, due to the closure of museums around the world during global pandemics, artifacts have been preserved without physical access. A generous interface would support representing the richness of cultural heritage collections, allowing for visualisation for visitors around the world at any time to explore and enrich interpretation between associated collections [138].

In many highland museums in Scotland - such as the Highland Museum of Childhood [139] and Grantown Museum [140] - their rich costume collections for visitor try-one experiences have been severely restricted due to the pandemic due to social distancing and hygiene concerns. It’s therefore been identified that an interactive digital pipeline needs to be established to support museums and charities with a cost-effective medium to present their artifacts and clothing with a reduced barrier to access digitally. This will support users in developing actionable insights with the scanned artifacts provide another dimension for the spontaneity of ideas and draw connections between heterogeneous datasets.

5.2.1 Review of Frameworks

Creating 3D visualisations traditionally is completed with proprietary software such as the use of CAD products like Autodesk Maya and Blender, an open-source alternative. These 3D model viewers are designed for experts and are not encouraged for use with the general population to view 3D models, however, they are widely used for video and cinematic representation. With the adoption of game engine technologies, scanned artifacts using photogrammetry techniques can be placed into a real-time environment allowing users to explore scanned objects in their own time from any device, regardless of the location of the original scan. With the use of a game engine, exploration of a 3D object no longer requires entirely passive viewing. Interactive approaches can be used to engage with the scanned artifact in new ways such as the manipulation of an object in real-world space using AR and virtual reality environments (VR). However, using game engines and application programming interface (API) frameworks requires extensive background knowledge and an interdisciplinary skillset involving 3D scanning, 3D model development, programming, interface, and interaction design. Due to this, many artifacts are left in storage waiting to be presented in a digital medium unless large funding pools are made available. Although platforms do exist for a fee to host 3D scanned content, such as Sketch fab [141] some scanned models may be of too high a fidelity to use in a real-time system due to large texture and polygon count. It is therefore an essential part of the process to optimise scanned artifacts. Pre-optimised assets for use in real-time environments are a growing market and attract considerable interest due to their interdisciplinary use amongst creatives such as video games, visual effects in movies, and advertising. Fab [142] is a platform announced in 2023 that aims to combine existing 3D model asset collections into a single platform allowing for materials, models, and other assets used in real-time engines' direct access and integration into their 3D pipeline of choice. 3D development may be completed in applications such as the Unity game engine, Unreal game engine, Cinema 4D, and Autodesk Maya to name a few. These pre-optimised scans support various levels of detail of real-world objects, allowing the opportunity to view using real-time platforms whilst still supporting denser resolution representation when using a baked pre-rendering solution - such as the visual effects industry for movies and animation. The scanned photogrammetry assets found in the Fab library are high fidelity with multiple optimised meshes of single scans allowing for immediate use in real-time engines depending on their requirements.

Further API frameworks for 3D model viewers include three.js which is a JavaScript-based Visualisation library supporting Visualisation and real-time lighting reproduction into a browser. Due to the library running the client side on the user’s browser, there isn’t a large requirement for any server-side graphical processing capability. If the users’ mobile device meets the necessary specifications, all the processing required to render the 3D model will be performed directly on the device.

The above frameworks and APIs are options available to present scanned objects in a collection allowing for presentation in a multidimensional format using browser or augmented reality mobile environments. However, they are not necessarily interoperable, there is a portability constraint when employing augmented reality if frameworks are not designed for multiplatform Visualisation and are exclusive to desktop environments. 3D Model representation for multiplatform environments have constraints such as operating system – Android and Apple – or browser support on the desktop all require different development pipelines.

5.2.2 Review of Outlined Framework

The case study project aims to support museums' digital transformation by exploring the scanning of museum artifacts and garments from the Museum of Childhood and Grantown Museum. Once scanned, an exploration into a Visualisation medium that allows users to interact and visualise the digitally twinned artifacts in augmented reality will be presented. It's hoped that this will support subjective benefits to users who use the developed product by enhancing understanding, facilitating new insights, or promoting creative thinking in their area of expertise.

The described augmented reality solution involved three key requirements. Must support iOS and Android devices due to museums and visitors using both devices. There must be a Visualisation of sorts to convey a digital twin of an asset. The application must be AR in nature, allowing the visitors to interact and save their experience with photos or screenshots of the augmented environment.

The project's short 3-month deadline from concept to deployment was an essential consideration during requirements gathering. Feasibility to support time for exploration into novelty, whilst still providing a stable solution for digital interaction and supporting the museum's future projects by creating optimised scanned 3D models for future use. The following three points were also considered before development.

- 3D Costume digitization
- Explore a mixture of virtual and physical experiences on location and online, sharing the stories of costumes to engage visitors and wider audiences to aid the resilience of rural museums in post-COVID recovery.
- Develop experiences using augmented reality (AR) applications to build knowledge on costume collections and engage with wider audiences.

Further discussions were held with the museum staff to understand their ideas for the project during the analysis stage of the development lifecycle. Comments from museum partners during this stage were as follows and an essential part of requirements gathering.

"We would like a magic mirror to augment clothing onto visitors."

"What is displayed on the tablet device should be mirrored onto the TV."

"We want garments scanned and also toys from the Museum of Childhood."

"Allow visitors to take pictures of family members with the augmented reality garment or item."

5.2.3 Pose Estimation

Initial considerations into the development pipeline focused on a Unity application that supports body tracking for a “virtual try-on experience”, allowing for a TV or projection to overlay 3D scanned garments on top of the user in real-time. During the feasibility study, one of the drawbacks of mobile pose estimation solutions was discovered. The Android SDK AR Core does not support full body pose estimation and only simultaneous localization and mapping (SLAM). Due to this constraint, two separate development pipelines to implement a body tracking solution would be required, as Apple’s ARKit does support 3D pose estimation. Therefore, an alternative pose estimation algorithm for Android body tracking would need to be explored and developed. It was found that Google’s TensorFlow lite libraries do support pose estimation on Android but would result in two separate applications being developed as both Android and iOS will require separate software development kits (SDK) to be developed preventing a single build process using the Unity engine. This is something of a disadvantage due to the unknown risks of developing two separate applications using low-level libraries for different platforms. A custom application layer would also be required to make tensor flow lite compatible with Unity thereby separating the libraries further. In addition, application libraries when out of support require an uplift that can take as long as the initial development as certain mobile operating systems rely on libraries which may not be in sync between iOS and Android.

5.2.4 Scanning

Exploration into off-the-shelf scanning products found scanning, then processing artifacts for cleanup to be a time-consuming exercise. Exploration into a pipeline using a DSLR camera followed by importing the photos taken into photogrammetric processing software such as Agisoft Metashape or Reality Capture. Metashape processing gave concerning results regarding camera pose estimation failing to converge during testing. The Metashape application generated 16 virtual cameras compared to the 33 unique photos passed for processing. Given the platform is highly technical and adjustment of parameters can take a lot of time, in addition to 3 hours of processing, it was deemed too time-consuming to progress further as a total of 10 artifact scans were to be processed, with each artifact different in shape and size to the other. The red dress below shows an example of the scanned result using a traditional photogrammetry pipeline with Metashape [143]. It demonstrates a problem during the photogrammetry pipeline, where topology could not be generated from photos as the lighting or points to converge were not located in the software resulting in an incomplete mesh. Processing into a mesh required a desktop machine and 2-5 hours of processing to see a result.

Figure 5.3 *Dress failing to converge resulting in a low-quality, incomplete mesh*

The photogrammetry process entails using virtual cameras to align photos in 3D space followed by generating a point cloud by identifying the convergence points between the aligned photos. The point cloud can then be processed into a mesh and exported into a 3D model format such as .fbx or .obj. Scans were not all completed in the same location, the differences between the scanning environments between the two museums resulted in the garments and artifacts requiring large adjustments to the photogrammetry software's parameters for each model.

Further exploration into scanning solutions is considered using Neural Radiance Fields (NeRF) which is a recent technique for 3D capture using deep learning to infer a 3D environment [134]. It exploits deep learning to infer a 3D representation of an environment from a collection of images resulting in highly detailed 3D models like traditional photogrammetry pipelines. Traditional photogrammetry pipelines rely on explicit geometric representations such as point cloud scans as previously discussed, resulting in intensive computing processing requirements. NeRF, however, represents the scene as a continuous volumetric function known as a radiance field which describes how light interacts with the scene [134]. A developed solution using NeRF is available from the iOS app store created by Luma AI [144]. This approach uses an iPhone camera to scan an object from numerous perspectives following a 360-degree motion. Once completed the images are processed in the cloud, and the resulting 3D model is available for download after processing is completed in .obj format. The processed mesh can be viewed on the mobile device or a desktop browser, allowing for analysis of problem areas in the mesh, for example where photos were not taken resulting in dead space, which can then be rectified by appending images to the problem areas rather than scanning the entire mesh.

The result of our experiments showed scanned models producing significantly fewer artifacts than the photogrammetry DSLR solution after processing. However, the main advantage of using Luma AI was the cloud processing capability, with no parameters requiring adjustment for each model. This approach facilitated a seamless scanning process, where local computing resources could be utilized for the refinement of 3D topology for previously processed scans, while cloud-based processing continued for the subsequent model in the pipeline.

5.2.5 Final Pipeline

By manually removing noisy mesh data generated by the resulting Luma AI model, Google model viewer API was chosen to present the 3D scanned artifacts in the browser supporting mobile AR. Depending on the size of the model and the surrounding noise mesh generated from the background environment where the scan was taken, it can take 30 minutes to an hour to remove the background topology. AR visualisation was further supported by developing a progressive web application that allowed models to be cached for offline use. The website was hosted for free on a GitHub repository using GitHub pages. This allows for an open-source approach to storing the 3D models which are scanned and allows the museums access to the models should they wish to hire subcontractors in the future to modify and reanimate the models. In addition, the visualisation of the assets via a webpage can support the in-house development of artifact tracking and Visualisation of garments when in storage. The process aimed to have the following.

1. GitHub Pages - Cost-Efficient hosting platform
2. Luma AI - Fast processing of high-fidelity scans
3. Autodesk Maya – Cleanup of mesh topology if required.
4. Model Viewer - Fast deployment of scans to the AR platform

5.2.6 Scanning

The scanning procedure was carried out resulting in an average of 30 photographs per artifact or garment, all photos taken from different perspectives ready to be loaded into Luma AI for processing using the NeRF approach.

Figure 5.4 *Photo of the scanning environment at the Museum of Childhood Dingwall*

By using Luma AI, scans can be processed in the cloud and exported to the obj format. Scanning using an iPhone also allowed for an initial preview of the scan to be viewed only minutes after scanning which helped identify troublesome areas quickly should a rescan be required. Some issues include: scanning under the arm for dolls and garments, garments where the material touched the floor, garments with high specular material, toys that were small and required to be hung such as puppets, and dresses with laced backs which had depth and high detail between the laces.

5.2.7 Dataset Cleanup

The file formats used by Luma AI convert models to obj format which is a well-accepted industry standard model container file format. It is a binary format which allows for conversion to other formats and is not proprietary to any platform / vendor and materials can be stored in a separate file. From here the model was loaded into Autodesk Maya where the mesh and textures are visible together. Mesh retopology focused on reducing polygon count and texture resolution, allowing for the model to contain no background noise that was generated from the initial scan – such as chairs or other materials in the vicinity of the artifact. Firstly, the background noise was removed, then mesh retopology by reducing the vertex count until the mesh didn't look distorted but was smaller in file size. An example of this can be found in Figure 5.5. Once the model was reduced and optimised, the format required for the platform of choice – Google 3D Model Viewer – required .glb file format. An advantage of this format is that it can be modified again after conversion and allows for textures to be packaged into a binary file over a reference to textures.

The use of FBX2glTF [145] was used to convert the cleaned 3D model into a processed .glb model. This is an essential part of the process as after cleaning the model, file sizes reached upwards of 80 MB per scan which has numerous drawbacks. One such drawback is large load times for initial internet download specifically in this use case where it would be used at rural museums, and the other requires larger device memory to display the augmented reality model in the physical play space which can reduce the effectiveness of the AR projection. As the website may not have an internet connection on tablet devices at the museums themselves, the models are cached and are available after initial download.

Figure 5.5 Unprocessed 3D scan of a dress to the left with processed to right

5.2.8 Evaluation

The presented case study demonstrates a low-cost multiplatform AR application supporting the presentation of a variety of museum artifact types with global access via a website. The scanned garments and toys from the Museum of Childhood and Grantown are optimised for internet representation with high fidelity after a series of post-processing stages are completed discussed in Section 5.2.7. As the 3D models are now available and processed, the scans can be used in a variety of applications such as advertising, and digital cataloging for tracking the garment's current location but also provide an opportunity for further immersive interaction engagement.

5.2.9 Further Research

It is envisioned that this application could be the start of a digital tour where users from around the world can view museum artifacts without having to travel directly to the museum. The solution as described above provides online visitors with a 3D digital library of artifacts allowing for engagement from the public on a global scale, without the requirement of physical presence. With the use of associated metadata from scanned garments and artifacts such as their current longitude and latitude position, further Visualisation approaches can be leveraged to view garments and artifacts from a topographic map. In Figure 5.6 we can conceptualize this in a VR environment. Previous research into this area has shown positive results with engaging with GIS datasets and their respective scanned environment. The figure showcases the locations of electric vehicle chargers around the country with a yellow electric bolt pinpointing the location of the queried scan.

Figure 5.6 *Future discussions led to GIS digital tours of cultural heritage sites*

For further development, in Figure 5.7 we can visualise how the already scanned and processed garments can be overlayed onto users' avatars for use in metaverse applications such as VRChat [146]. As the framework for rigging a humanoid character is well-researched and understood, the time constraints on integrating the garments onto a humanoid avatar are the main drawback when wanting to create a real-time animation with live input from users. With the aid of a camera or depth sensor such as the Azure Kinect, we can track users' real-world movements and project them onto a digital avatar wearing scanned garments. Where artifacts are not wearable, they can be rigged and animated to support engagement with the public, this pipeline is similar to video game asset creation for characters and props.

Augmented Fashion Library
Tap the arrows to explore the digital collection

Cashmere Kilt, 1988

This Kilt was developed by Paul Rutherglen located in Orkney and took inspiration from natural world byt integrating wool colour.

Figure 5.7 *Demonstration how a scanned kilt can be animated onto a humanoid rig*

5.3 Point Cloud Processing for UAS Drone Technologies

In Section 5.2 discussions are made for a standalone, low-cost, end-to-end solution for visualizing artifacts scanned using an AI supported photogrammetry pipeline for low cost viewing online. For larger datasets such as bare earth scans, landscapes, and buildings the use of airborne LiDAR drones can be used to scan environments. Although not a cost-effective solution in comparison to using mobile photogrammetry solutions, the technology potential for digital twinning large objects and areas allows researchers access to new datasets to analyze.

With the advancement of lighter airborne scanning technologies, more data is becoming available for scientists and analysts to optimise workflows and assist in the alleviation of issues that may not have been previously conceptualized. The scanned environment can be processed remotely to extrapolate data from areas of interest by digitally replicating the physical world using unmanned aircraft systems (UAS) equipped with LiDAR sensors and high-resolution cameras. Utilizing these scans in simulated digital environments, we can simulate erosion, flooding, and other natural weathering forces that may occur on the landscape. In such scenarios, digital twinning provides a technological advantage in preserving real-world environments that can be reviewed and made accessible for analysis remotely at a later date.

5.3.1 Data Collection

Retro-fitted unnamed aircraft systems using LiDAR, high-resolution cameras, and FLIR sensors are now able to scan and reconstruct physical landscapes into digital formats. Remote sensing (RS) platforms used by UAS can be configured to produce a point cloud data set of an environment and the resulting digital twin can then be processed and further analyzed. Capture approaches utilizing different sensors vary based on expense and scanning requirements. Each approach has its advantages, and with processing algorithms improving, cheaper solutions to capturing in-depth information are becoming increasingly available.

Both video scanning followed by frame segmentation and flash LiDAR variants will use synchronized time-of-flight cameras to detect depth. Pulse-based approaches will use scanner-less technologies and instead, these high-powered optical pulses are emitted toward objects with nanosecond return delays that once processed capture depth information.

Single Photon LiDAR (SPL) provides an efficient approach to rapid, high-resolution 3D mapping. SPL requires only one detected photon per ranging measurement, as opposed to hundreds or thousands of detected photons per ranging measurement for conventional or Geiger Mode LiDAR.

Current airborne scanning trends use a combination of sensors; LiDAR, FLIR for nighttime scanning, and high-resolution cameras for daytime capture. An example of using other sensor units is the use of a global navigation satellite system (GNSS) to record land prone to flooding and compare scans over time to this geographic location [147]. Another such example is digital twin clones of land mass which can allow for simulations of natural fire hazards. High-definition resolution cameras are also applied with an inertial measurement unit (IMU) which is used to get the pitch and attitude of drones while scanning the environment. This allows for algorithms to consider the difference in UAS height when scanning to have a consistent 3D spatial scan. Frameworks have been established to understand trends, overlaps, and gaps between datasets [148].

Due to scanning and moving at speed in the air, it is frequent that pulses from the single photon laser record the terrain before the data is reflected to the sensor. This alignment problem in high-fidelity scans can now be mitigated with the advancement of real-time 3D graphical engines and their processing pipelines to render datasets. Issues with algorithmically aligning datasets are related to the noise produced from scans typically prevents point cloud systems from matching accurately.

Photogrammetry approaches to point cloud generation have outperformed LiDAR for point cloud densities. At the cost of longer processing times, passive image-based depth approaches used in photogrammetry do not work well in darker or shadow-heavy environments. It is for this reason that height or distance estimation will use LiDAR systems. An example of the use of LiDAR-based sensors for distance estimation can be found on the Dragon Crew capsule by SpaceX [149]. By integrating computer vision with LiDAR, the accuracy of the Dragon capsule's position and velocity can be more precisely verified. This is achieved by comparing the LiDAR image to the thermal image depth map and then fusing the depth maps to generate a precise distance representation. Unsupervised learning techniques have been studied to improve the estimation accuracy of depth data using a single thermal image to tackle the low illumination issue [150]. Issues involved include data generation which is restrictive due to the illumination conditions at the time of data capture. It is therefore still preferred to use a combination of LiDAR and thermal images to get accurate depth estimation in low-light conditions. Synchronization between the LiDAR scanner and camera sensors, however, proves problematic during alignments of the different datasets. Without multispectral LiDAR sensors which are novel, accurate colour representation requires a camera to be used simultaneously or run into alignment issues if recorded at different times. A LiDAR scanner operating on a single wavelength will not be able to scan surface colour.

5.3.2 Manual Processing Stages

After point cloud data has been collected, visualisation, segmentation, classification, filtering, transformations, gridding, and further regression activities can be completed [151]. Segmentation, classification, filtering, and data transformations are used to support understanding LiDAR point cloud scans. Segmentation allows for areas of a scan to be separated based on photon intensity value, highlighting dense areas of foliage or highly reflective material alongside less dense and reflective materials. This can be completed through the use of AI or a processed library of point intensity to reference. When used with classifiers, areas of scans can be further grouped into classes such as buildings, signs, and vegetation. Some classifiers are based on AI solutions and others use a processed library of values such as point intensity to estimate what the object is that the point originated from [152]. Filtering this data further includes removing unwanted noise from scans or areas that are not relevant to the subject of the scan. A use case for this is to generate a bare earth scan and remove foliage, and vegetation that may be occluding the earth's surface during satellite or drone scanning.

5.3.3 AI-Assisted Classification

Using machine learning to enhance understanding and interrogation of recorded data can support identifying and classifying relevant areas in shorter periods, reducing issues caused by human perception and cognition. A use case for this is to scan and then classify objects in the dataset to support understanding what is included in the digital scan. When compared to manual techniques, AI-assisted tree segmentation can achieve up to 98% accuracy, resulting in highly accurate bare earth scans.

Hough space from LiDAR point clouds has been used with a convolutional neural network (CNN) to classify 3D objects in point cloud space generated by a LiDAR [153]. The Hough space is rasterized into uniform grids where processing is then completed in a convolutional neural network. Additionally, the study used a semi-automatic 3D labeling tool which resulted in a library capable of identifying four objects: wall, bush, pedestrian, and tree. This technique of labeling will analyze and extract shape attributes to classify by training a model [154]. The CNN would then use this library to train the neural network through large amounts of offline datasets resulting in object classification with an accuracy of up to 93.3%. For neural nets to be trained using supervised learning, large quantities of dense point cloud datasets are required with high-fidelity scans for clear object recognition, without noise or obfuscation. The labeling of these clouds needs to be accurate. Due to the unstructured distribution, the nature of point cloud datasets makes it difficult to use techniques such as deep learning for classification, segmentation, and object feature detection [155].

5.3.4 Unsupervised Classification

Cluster analysis is used to categorize areas of a point cloud or other dimensional dataset to assist in analyzing objects and their context within a scanned environment. Clustering algorithms are often used during the exploratory stage of data analysis and focus on analyzing points with their neighboring points. Two points close together within a specified value are likely to be clustered together and points far apart are not. Therefore, cluster shape is an important consideration when working with noisy datasets as inaccuracies can occur. By using Euclidean distance, the center point of a cluster can be calculated, and the homogeneity can be compared against another cluster's center point. By averaging the distance between points in each cluster we can then correctly group points to the correct cluster. It is critical in unsupervised classification that separation and homogeneity are viewed as requirements for an accurate clustering system. To validate clusters generated, measuring the silhouette coefficient for given cluster points can support validating points within clusters [156].

5.3.5 Depth Estimation

A benefit of LiDAR over a vision-based alternative is the conditions it can operate in without the need for further sensors to supplement the dataset retrieved. Passive images from RGB sensors such as high-resolution cameras are not suitable for dark environments but sensors such as FLIR are suitable. When the stereo pair are used together depth estimation algorithms can synthetically generate thermal images from their RGB counterpart [150], allowing for a significant increase in depth data available for FLIR sensors. Depth estimation is an important component of generating 3D environments from scans and is directly related to the field of computer vision. Many UAS and autonomous vehicles require a form of depth estimation to vectorize the world the sensor is in to make sense of it and provide the necessary steps to interact with the environment such as SLAM.

5.3.6 Data Analysis

A clear challenge during the analysis and processing of point cloud datasets has been to streamline the ingestion process of data where multiple data types are being used. Previous data fusion techniques of multi-modal datasets have issues such as image registration and angular distance automation. Proprietary software packages are limited in functionality and are often expensive to procure.

Referred to as data holidays, some return pulses from LiDAR sensors have interference preventing full data collection from the returned point. Overlapping the data points by repeating scans can help preserve the data but at the cost of increased and potentially redundant data points if the data is retrieved successfully. Dropouts of scanning can occur with specular materials such as water ripples intersecting at an angle can cause data dropouts. Similarly, if the material is not specular and absorbs most light it is difficult to get a return photon to the LiDAR sensor causing dropouts of data.

Reflected objects such as birds can interfere with high scanning points such as tree canopies. The elevation data will be offset by the noise added by birds during a scan from a height. Low points can also have issues related to reflective materials in the surrounding environment. Similar issues occur during GPS tracking in urban environments where user tracking can become obfuscated due to tall buildings.

Due to light absorption and sub-surface scattering, subsurface LiDAR scanning has been notoriously difficult without post-processing. Used by ROV pilots, the additional 3D LiDAR representation of the surrounding environment is used for navigation, and alignment during surveying and construction of sub-surface pipework [157].

Issues related to scaling multi-dimensional systems and image registration include how the points are interpreted and work with other multi-modal data sets. Coordinate systems for various sensors are different so there is a requirement to transform dataset formats from WGS84 to UTM for example, where it is likely easier to convert any coordinate system to latitude and longitude so it is compatible with EXIF data which may be used from photogrammetry assets.

By taking into consideration the above issues, the processes available for visualizing high-density point clouds may be best suited for real-time alignment and then playback by using game engine technologies. A clear problem for depth-related data is losing data during data format conversion during loading to proprietary software products or even ASCII where crucial header information is lost. It is therefore important to merge datasets into a single file format that includes header information from each frame. A potential solution to this is by merging into the .mkv file format as this container file format can allow for numerous file types to be embedded and processed in numerous applications. This is the file format used for real-time recording and playback with the Azure Kinect DK LiDAR sensor [158] and further discussion can be found in Section 5.3.8.

5.3.7 Data Visualisation

Most 3D points recorded by LiDAR are stored in the form of a vector coordinate system and saved to a file type that is suited for the processing application that is used. The x, y, z points are referenced collectively after scanning, to produce a point cloud space. ASCII formats end the file extension with *.txt, *.xyz or similar, and traditionally store the vector coordinates in 3 columns, representing a single point per line. Reading a single point would take one line read which will add additional computation time when reading the points out to a buffer or rendering system and take up system memory.

Additional file formats are now more accepted due to the increased meta information that can be gathered about points. A smaller file size is also available due to the binary format type which in turn increases read-time efficiency. Unlike ASCII, the LAS format has a header that contains more metadata about the scan such as the number of total points, data extents (how much continuous area of storage is required for the file in memory), flight date, flight time, number of returned points, offset values, and scale values. Each LiDAR pulse contained in the binary body includes the x, y, and z location of the point, GPS time, return number, intensity value, point classification value, scan angle, RGB values, scan direction, edge of flight line, user data point ID and waveform information.

The LAZ format is the same as LAS but uses optimised compression and is also open source. Additionally, the zLAS format also compresses the binary file. However, due to its closed access by ESRI government and private sector agencies are hesitant to use it.

Tiles are used in the LAS data format to improve loading times of desired areas of point cloud data [159]. It works similarly to a voxel chunk engine and tiles of points are loaded based on the LAS file header attributes, allowing for correct identification of scanned land areas and therefore more appropriate loading of land mass. By combining with multi-threading applications, tiles can be asynchronously loaded alongside other tiles allowing for volumes of data to be loaded concurrently.

Proprietary processing software such as Autodesk ReCap converts raw scan data to scan files (RCS files), and project files (RCP files) that reference multiple RCS files. Both these formats can be attached to an AutoCAD drawing. This approach is not open and requires software licensing. Given that this area is still disputable based on the use case requirements for the scans, a discussion below will outline the core concepts behind storing and retrieving point cloud datasets in a file.

5.3.8 Point Cloud Storage

3D LiDAR recording for playback has the advantage of anonymity during data processing tasks due to the potentially unstructured distributed points, which only show the full scanned object once processed. Applications for such a feature can include site security and counting people entering a venue. Playback of the captured recordings can be completed using proprietary point cloud graphics engines, however, game engines such as Unity and Unreal are becoming more favorable for real-time graphics processing in multimodal data representation. A potential file format for serializing and deserializing content is a bitmap image format. Although unconventional to the file types described above, discussion on how to serialize and deserialize the format is valuable for understanding how to represent points in 3D space in a game engine. The EXR format is described as a lossless high dynamic range (HDR) file stored in binary format.

Optimisation techniques involved in this include pooling the points for the images to prevent memory spikes and then passing the vector 3 coordinates to the points for representation. Point cloud datasets may not align due to sensors being in different locations during recording; By manually assigning datasets for each frame, it may be tedious at first, however with technologies such as virtual reality and point classification, this process is becoming ever more streamlined. During the playback of LiDAR images, interpolation can be used to support the masking of gaps between scan rotations.

An alternative format for storing LiDAR scans for playback and further analysis involves converting LiDAR scanned values to 16-bit/32-bit colour EXR images and playing the x, y, and z positions stored in the RGB values back over time.

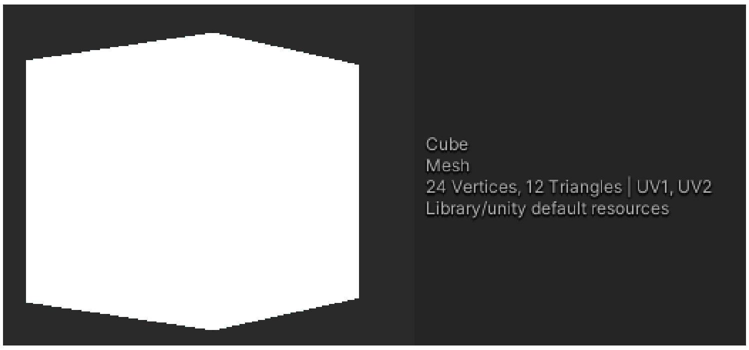
Microsoft's Azure Kinect supports scanning using numerous sensors simultaneously and is recorded in a .mkv binary format. Each recorded data type can then be accessed by deserializing the .mkv file type to play back desired data over time as it is not a static scan but a scan over time format. Due to hardware still being in its infancy in the research and development market, this solution allows for access to large *multi-fusion datasets* in one recording where audio, infrared, point cloud and video are stored in one file.

Further optimisation steps can be completed during playback of the recorded data such as 2x2 binning where diagonal pixels are merged into one, resulting in the resolution being quartered. A side effect of this, however, is doubling the signal-to-noise ratio on the output. Other optimisations that are used for traditional video and image compression can be used if the point cloud is unstructured. Depth delay during playback of a recording is another option. By delaying the depth for high-density clouds, we can assign vector3 positions of the depth values before the colour values have been read. This approach allows for pooling point positions before colour textures have been assigned.

If points are not linear after scanning due to the unstructured distribution of points during the scan. By using an image storage approach, cropping the image used to store the points would reduce the image size buffer for reading and the total amount of points being rendered on screen for further optimisation. This is found to be a novelty in using a bit map-based approach.

If the cloud data is unstructured in point distribution and stored in a bitmap, during deserialization we can crop the bitmap image to reduce points rendered while still keeping the point cloud "model" structure.

How do we store point cloud datasets in an image? A rendering system has been developed during the experimentation stage of this study that allows for point cloud data to be stored into an EXR image (RGBA values per pixel) and rendered on screen using particle systems with a custom shader. Using the Microsoft Kinect sensor [158], we can record areas of 3D space in real-time and then play back a point cloud recording using the Matroska (.mkv) container format. This approach allows for each frame of video footage to be used with additional metadata such as accelerometer data and gyroscope time stamps. Another advantage of a bitmap graphic for storing 3D data is the potential to exploit computer vision algorithms for the segmentation and classification of pixels which are actually representing a 3D dataset. A step-by-step process of an alternative renderer that has been developed can be described below but for playback in the Unity game engine. Additionally, serializing 3D models was also developed which will take a 3D model and parse its vertices into 3D points, which are then saved to an EXR format. A cube with 24 vertices is serialized into the following EXR image for demonstration. Note, the vertex count would be expected to be 8 vertices, but the default cube mesh in unity was used which has 24. As there are 8 vertices on a cube, you can't assign multiple normals to a single vertex, therefore 6 faces of the cube would require 4 vertices each.



Cube
Mesh
24 Vertices, 12 Triangles | UV1, UV2
Library/unity default resources

Figure 5.8 *Cube to be serialized into EXR image using vector 3 coordinates in RGB*

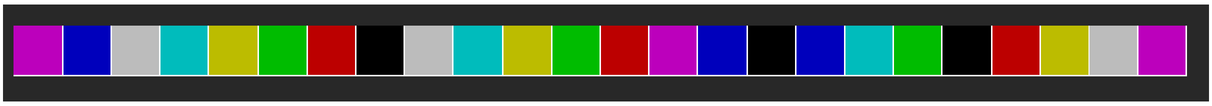


Figure 5.9 *24 Colours are required to represent each vector 3 position*

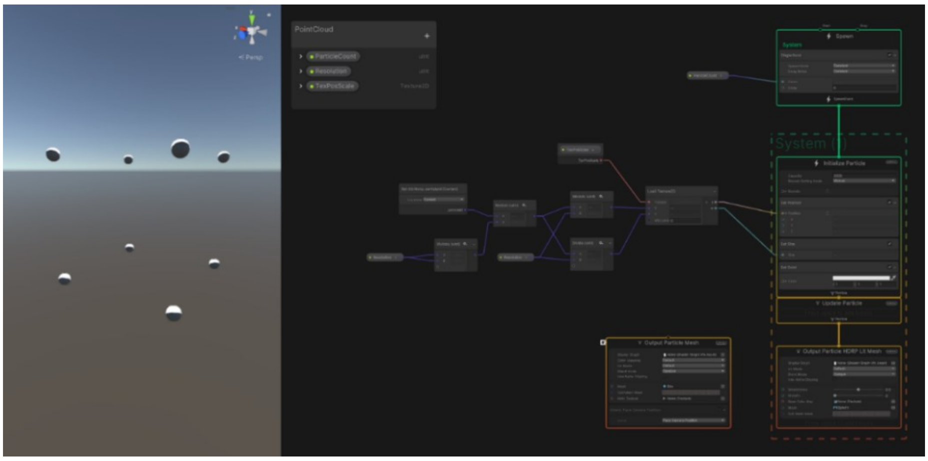


Figure 5.10 *EXR image to point cloud renderer using custom particle system approach*

An alternative format described in “A New Method of storage and Visualisation for Massive Point Cloud Datasets” [160] allows for a level of detail (LOD) system to be embedded into a multi-band image using the GeoTiff file format. The diagram below describes how this system works by storing information for each LOD system in its own colour band. It is possible to store a GeoTiff file format into a container file format such as .mkv, allowing for further meta data.

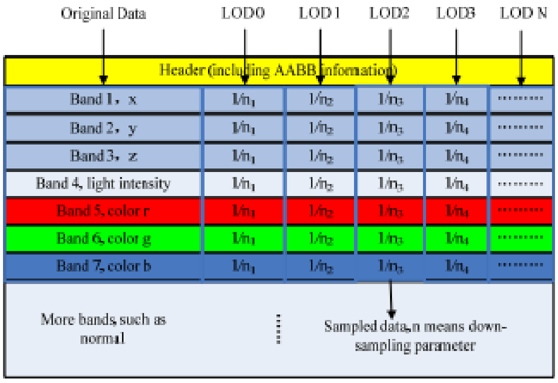


Figure 5.11 *EXR GeoTiff LOD System for storing point cloud data* [160]

Additionally, point cloud model images can also be sequenced together to create a real-time playback system. This would further enhance experiences that are happening live, for example a public event or performance. The ability to stream the point cloud data in a format that can be lossless is important to prevent data drops which could lead to areas of the 3D model representation that are missing. The TCP protocol can help alleviate this over a network when used with the .mkv format. The .mkv video format allows for the file to still be read with incomplete line endings and is why it’s used predominantly as an initial lossless video recording format to prevent video footage from being corrupt should the recorder fail.

For example, a 2000x2000 pixel 16bit RGBA EXR will store 2000 x y z positions on each row of pixels. Using a playback standard of NSTC framerate at 29.97fps this means that this one image can provide 83.3 seconds of playback.

Taking this approach further, we can take frames of points in a cloud and play them back with different positions using image formats described. Unstructured distribution of points can also support an efficient, low performance mechanism for a LOD system by cropping the image to any resolution desired.

5.3.9 Remote Sensing of Forestry, Glacier, and Ocean

Climate smart forestry preservation techniques have been used to focus on more sustainable solutions to forestry management. An example is using an AI platform to identify log density accurately allowing for increasingly accurate logging stockpile predictions which can further support in understanding the environmental impact such as soil densities, erosion used by trees in the area if they are ready to be logged. By creating a digital twin of a forest, segmentation algorithms can be used to extract individual trees and branches with 95-98% accuracy [161]. Furthermore, once the environment has been scanned and the trees segmented, the bare earth scan can be isolated to support in simulation of erosion, flooding, and other natural weathering forces.

This industry taking advantage of digital twinning focuses on sustainable forestry development in response to climate change and reducing greenhouse emissions while additionally capturing carbon from the atmosphere. With conifers reaching 30m in height, it has been technically difficult to scan between the different layers of forestry with spotty shrub layers and dense mosses and lichen layered ground. Accurately recording the variations in scanned forestry has been difficult for this reason, with a variety of species living in the different forestry layers.

For forest inventory management, terrestrial LiDAR scanners (TLS) allow for mass digitization of large areas quickly, with post processing, further classification and segmentation completed at a later stage. Manual measurements are not required when using this approach, therefore data collection speed is increased and has the opportunity for using AI methods to increase accuracy of the data. Classical methods for calculating tree area involve collecting diameter at breast height (DBH) which is affected by an error rate of around 5.6%. There has also been bias to the DBH collection approach of up to 26%, this is another clear indicator that traditional approaches are not suitable for accurate information retrieval or estimation. During digital twinning of forestry using the VirtSilv platform [162], a novel software that takes into the exact shape and measurements from trees not just a more general sphere shape in which a DBH formula can be applied, it uses the unique shape of each tree for its calculations.

5.3.10 Conclusion

Areas that have a demand for interactive visualisations of point cloud datasets vary from static scanning to create digital twins of historical sites, to real time visualisation of forestry for remote sensing. Industry applications have used LiDAR UAV for applications related to forestry, habitat monitoring to stockpile recording and digital twinning for preservation. By combining 3D datasets of bare-earth scans or raw earth scans we can see how point cloud scanning technologies can be used to reconstruct a digital twin of the environment to support simulation and area understanding based of delta change between datasets. In addition, search and rescue applications to 3D map the environment with object recognition based of computer vision is also being explored in real-time engines. Environmental monitoring [163] such as glaciers is another situation where multiple drones can be used [164]. One with FLIR and the other with LiDAR scanning, once overlaid, demonstrate how the glacier surface is changing and apply more current thermal imagery to gain a greater understanding about the debris cover and melt rate.

The use of drone scanning technology has allowed for dense data capture of historical sites which can be used to explain and highlight areas of historical interest while also uncovering unknown stories to tell. This section has discussed the technology behind creating dense point cloud scans and an interesting approach to data storage through bitmap graphics.

By digitally replicating the physical world, more data becomes available for scientists and analysts to optimise workflows and support in alleviating issues that may not have been previously conceptualized.

This can include simulating a virtual entity such as a business, allowing for predictive analysis on changes to operations simulated to support in alleviating issues should the changes move to the real business model. Additionally, 3D models are also forms of digital twins and allow for areas as large as cities to be scanned and digitally constructed for further analysis. These high-resolution datasets can then be used to simulate scenarios such as flooding, analyze areas of land mass that otherwise would require onsite presence or an entire city providing data about infrastructure, businesses, and movement of people.

LiDAR sensors' advantages over traditional video surveillance systems include increased privacy when tracking users in a specific field of view. Issues using video surveillance systems can arise during tracking where persons are occluded, failing to accurately track personnel count and privacy storage when tracking with faces visible. With the addition of recording in colourless 3D, LiDAR tracking systems for monitoring personnel count at venues such as concerts, conferences and areas with large gatherings of people can be private. Identifiable features on a person such as their face will not be of sufficient fidelity using such a system at range. This provides an element of privacy built from the data collection stage of the process. By having anonymous data from the start in real time processing, there would be no further anonymization processing required.

Chapter 6

Conclusion

Improving access to immersive digital experiences has implications across industry and research sectors, from quality of life for individuals to understanding heterogeneous datasets more intimately to support the evolving world as society tackles big problems from neuron and cellular visualisation to the migration towards a more sustainable future. By focusing on enhancing accessibility and interaction within VR environments, we can enable users to leverage multidimensional datasets in ways they otherwise may not have been able to conceptualize. The initial research question addressed in this thesis was "**how can we improve insight and engagement in a virtual environment?"**. The presented research has led to advancements in interface design and optimisation for standalone with consideration for the future of edge computing, with a strong emphasis on accessibility and user-centric design principles.

A cornerstone of the work has been the development and integration of a low-pass filter targeting ray-based interface input, specifically designed to normalize benign essential tremors discussed in Chapter 2. The integration of the 1-Euro low pass filter [68] into the interaction layer of VR applications targets ray-based interface input, striving to normalize benign essential tremors which are common amongst the population with a frequency between 9-12Hz.

This innovation is pivotal in making *existing* VR applications more accessible to a broader range of users, including those with movement disorders. This accessible-first design philosophy was then applied in the creation of a modular 3D GUI discussed in Chapter 3 with metrics from previous studies to also support developers in creating these immersive experiences such as Figure 2.6. This interface is a fully working system and required VR interface design from the ground up, demonstrating a significant leap forward in VR design as we consider the future advancements of interaction modalities such as hand tracking, voice, eye tracking and BCI. It addresses the need for multimodal input mechanisms and paves the way for more intuitive and inclusive VR and AR experiences. The modular 3D GUI allows for the integration for future input modalities as they become released in commercial VR and AR HMD’s, allowing for a familiar interface design paradigm to be used, so that users can intuitively adopt a new input medium without having to relearn a new interface experience.

Exploration into VR render pipeline optimisation discussed in Chapter 4 highlights the necessity of leveraging the limitations of the human visual systems in respect to HMD display development, we must advance HMD visual systems resolution whilst reducing latency without overburdening computational resources. The concept of 'frameless eye-tracked foveated rendering' emerges as a promising solution for reducing processing requirements in both standalone and desktop VR setups by exploiting eye tracking. This approach not only enhances the user's experience by providing high-fidelity environments but also marks a significant step in sustainable computing practices within VR technology.

A key achievement in the research is the development of world ready solutions from filter reduction to developed Visualisation platforms that enable the analysis and collaboration on heterogeneous datasets in VR and AR environments. This capability allows for cross-national collaboration spaces, improving insight and engagement with multidimensional datasets without the necessity of on-site presence. Such advancements are crucial in fields ranging from environmental monitoring to urban planning and historical preservation with discussions on advanced capturing technologies that can be used for digital twinning discussed in Chapter 5.

The work underscores the importance of considering human motion and interaction patterns in VR interface design. By focusing on reducing barriers to access, we have created more engaging and less frustrating interaction experience for novice and expert users alike. This human-centric approach is vital as interface input approaches evolve within spatial environments, and it becomes increasingly important to re-evaluate the effectiveness of new tracking abilities made available for VR. We can employ user evaluation studies capturing an additional dimension of data in a 6-DoF environment, capturing users tracking over a journey in a VR experience. This allows for replay and analysis of user actions during an interaction, which we can then aspire to improve through iterative user testing once problem areas are highlighted.

The significance of VR in enhancing accessibility and mental well-being, particularly for users with mobility challenges and communication, cannot be overstated. The research contributes to the advancement of multimodal spatial interfaces, highlighting key constraints and opportunities in areas that have been previously underexplored. This thesis not only addresses the technological aspects of VR but also its potential in improving communication, comfort, and inclusivity for a diverse range of users.

By presenting a comprehensive body of work that lays forwards the frameworks for developers and fully integrated solutions allowing for a strong foundation for future innovations in VR technology. By adopting an accessible-first design approach and focusing on human-centric interaction, we have opened new cross disciplinary avenues for research and development. The contributions aim to inspire further exploration and development in VR and AR, ensuring that these digital spaces are improving in fidelity, inclusive, immersive, and accessible to all.

6.1 Limitations

Despite significant metrics gathered from participants during a key user interaction study in Chapter 2 using the 1-Euro filter, insights into the relationship between 2D UI button and slider interactions have highlighted a potential area for further study with multimodal input. For example, the potential to integrate the filter into Chapter 3's study using the modular 3D UI. The main study in Chapter 2 focuses on ray-based interaction, which is not necessarily required for the 3D GUI developed in Chapter 3, and it is unknown if it would have the same effect on task completion time but an hypothesis is that it would decrease task completion time for 3D graphical elements with the filter added to the users virtual hand tracking during button or slider selection.

Chapter 3 presents a novel modular 3D user interface empowering users to place UI elements where they desire. Limitations in this study were predominantly related to the user focus group count, as it was a preliminary study. Although a key output of the study was a framework that can be used for future research focus groups in VR. It is the novelty of the focus group format and data processing of study results which made it challenging to run additional focus groups in a given time period.

In addition, the interface is designed to work with current and future input modalities such as ray, eye tracking, voice, and BCI. As the Quest Pro was not released at the time of the study, the eye-tracking input was not included in the focus groups. This study could be further enhanced with the addition of another group focusing on eye-tracking inputs with ray for comparison. Another limitation is with BCI as a new technology, without integration for VR currently available without experimental hardware, it demonstrated that there is potential for future work here when the technology becomes more available.

Chapter 4 explores optimisation using control systems theory to mitigate oscillating movement produced through real-world user action in the virtual environment. This results in floating point errors and potential interaction artifacts, which is an area for further study. However, Pixel Arcade [124] – a VR platformer - exploits this technique, allowing players to use a "swinging" locomotion mechanism to progress through the level, validating its implementation. What was not validated, however, was the frameless eye-tracked foveated rendering discussed. This is only in pseudocode, where one half of its implementation, the eye tracking, and the other half, frameless rendering, needs to be integrated into one coherent render pipeline solution to run experiments to validate the hypothesis.

These limitations highlight the need for further research and development to enhance the robustness and applicability of the presented solutions, ensuring they can be effectively integrated and utilized across various VR contexts and input modalities.

6.2 Future Work

6.2.1 Collaborative Spatial Video Player

The most accessible visualisation platform to encourage migration into VR, may just be video playback. This can be traditional 2D or 360 video. It's a familiar medium for users who have never interacted with a real time 3D environment before, such as games, specifically the ageing population. However, it is a common design space where users exploit their proprioception and visual spatial awareness in reality. It is theorized that a simple 2D video player, exploiting spatial interface design elements using tremor reduction and a multimodal modular 3D GUI will help reduce the barrier to access to experiencing VR (as discussed in Chapter 2 and 3). With similar reasoning to the 3D map being presented in Chapter 3, as the map being a familiar visualisation medium with minimal controls, being zoom and pan. For video playback controls being selection, play and pause.

The ability to put on a HMD and immediately be placed into a shared environment with video playback is a familiar metaphor for users who have TVs or monitors. Bypassing other applications complex UI and allowing users to only see interface elements related to the spatial video, preventing overwhelming users with a plethora of VR options they may have not been exposed to before. Once the play space settings have been configured, and anchor placement of fixed spatial elements in real world space, users don't need to use joysticks for locomotion and can instead rely on natural, high-fidelity movement. This approach allows for a one-time set up with only essential spatial UI elements being displayed.

As many traditional 2D users are not convinced by the impact of VR, the majority of the population may require slow migration into the VR platform. A collaborative video play space supporting mixed reality is theorized to allow families and friends to communicate in a present digital environment, with an enhanced 2D video and cinema experience. This can enhance video digestion allowing for effects such as reactive lighting and room anchor points to anchor virtual screens into real world space. Thus, removing the need for a TV or monitor as HMD formfactor reduces in size whilst enhancing video over traditional mediums as discussed in Chapter 3. The passthrough capabilities in newly released headsets can allow for this collaborative environment if users don't want to be fully immersed but presented an augmented environment in their familiar living room or office. Once users become more exposed through mixed reality solutions, they may migrate comfortably to their own virtual environment. An experimental user study of novice to intermediate users could be completed again with a comparison to traditional mediums to understand the bare minimum requirements for interaction in a digital space.

6.2.2 VR Analytics for Enhancing UX Design

Preliminary exploration has been completed into developing an analytics platform during the development of applications used for VR studies throughout this thesis, which allows developers to replay and highlight users' exact motions during interaction. In Chapter 2 we tracked users task completion time by hand using an external monitor, its possible this study could instead be crowd sourced online and metrics automatically populated into a database for further analysis. By recording user interaction and then replaying for analysis at a later date, we can use techniques such as journey mapping to evaluate the problem areas users encounter from time spent on task completion. Reanimating user action remotely after recording, allows for capturing of user datasets in large quantities and even remotely through the use of platforms such as VR Chat where problem areas can be visualised through the use of heat infographics and analysis on more intricate user motion. Such a solution would enhance the efficacy of the data gathered from the studies conducted, such as those in Chapters 2 and 3.

Evaluation of users in a variety of 6-DoF environments is advantageous also for view in VR or a 3D application on a traditional monitor. The extra dimension made available can enhance datasets which are typically 2D for getting a better overview of user activity during a task. Similar to keystroke-level modeling (KLM) [165] and GOM model [166] based approaches, this area also looks at the third dimension, including the ability to see what position the user is in such as if they are seated or standing using world space. By selecting a suitable demographic of VR users to evaluate a VR interface, whether advanced interaction concepts such as BCI and 3D interaction or a more simple design taken from our understanding of 2D interaction, we can process large amounts of data with the goal of getting further insight into metrics that are not explicitly provided, such as user's height, reach, tremor rate, speed of movement during interaction, headset adjustment periods, controller usage time and play time.

6.2.3 Web RTC

Optimisation will become more prevalent with new approaches aiming to exploit AI for edge computing solutions. Edge computing is an emerging paradigm striving to use cloud computing to bring solutions to users that otherwise wouldn't be feasible by using a distributed architecture for processing, to then rendering or displaying the result to the end user.

With Web Real-Time Communication (RTC) providing real time streaming from 3D scene environment to a video feed, allowing users to manipulate camera and item movement as the 3D scene is processed in a cloud environment and the video stream being represented back to the user's client. This approach only requires the user to have input capabilities which will then be processed by the server hosting the 3D environment. Platforms such as Xbox and PlayStation support streaming of content from the cloud onto their client console, allowing for content that isn't supported by that platform to be played. As we progress to an era of faster and faster bandwidth with Wi-Fi 6E, we can see how streaming cloud rendered 3D environments with the latency being as small as < 0.2ms is feasible for a VR solution. For VR specifically, a scenario using existing Steam link or Quest link allows content to be wirelessly streamed to the headset. This removes cables and frees the user to walk around the real world play space. Although processing is currently completed in house through the user's desktop computer and then streamed over their internal LAN, it shows how the progression to exclusive cloud processing is a potential if latency can stay within the tolerance of <7ms to 20ms as discussed in Section 4.2.1. Scenarios where adoption of such a pipeline would be limited in areas of low internet latency such as the Museum of Childhood case study discussed in Section 5.2.

It is theorized, that rendering low fidelity assets in a 3D scene on the user's client, and then relying on cloud Web RTC processing of complex geometry and then streamed back to the user's client will allow users to be immersed in a high fidelity environment without the requirement for high end processing hardware on site, reducing the barrier to entry for high graphical experiences requiring a larger up-front financial investment. By masking objects of high fidelity, we can then overly the Web RTC feed processed in the cloud onto the user's experience. This results in the user having low latency experience, and for dense objects to be rendered and interacted with only when they are visible. Where we can only render essential interactable components client side (such as the GUI elements discussed in Chapter 3). Considerations into using a segmented/masked streaming approach for static objects that cannot be interacted with would benefit from further study allowing for latency to be compared with a full Web RTC experience to a masked experience, showing distribution of processing in a 3D scene vs the traditional one system rendering approach. To conclude, a scenario is provided.

A 3D scene environment with dense photogrammetry scans is presented to the user in a VR headset. The environment consists of trees and sticks. The player can interact with the sticks on the floor of the 3D environment and rendered on the user's client system for displaying onto the VR HMD. The trees in the distance rely on a LOD system that is rendered in the cloud. For far away trees – LOD 1 - that are billboard, PNG graphics can be processed for rendering on the user's client. Demonstrating the minimum amount of local processing required to display an item, which is an important point in the proposed designed architecture. As the user moves closer to the high geometry tree, the client can render the LOD 2 tree asset and trigger the WebRTC server to take the players perspective through rotation and position data. It is at this point that the Web RTC server is ready to overlay a masked outline of the high-fidelity 3D tree to be streamed back to the user's view in the HMD. As the user moves closer to the object and detail can be made out – theory of 20m based on depth perception being high as discussed in Chapter 2 – the user is then presented with LOD 3 system which is the full detailed model of the tree asset. This being rendered in the cloud and streamed back to the user's client HMD. It is in this process that the user's client is now just rendering locally LOD 1 and potentially LOD 2 tree assets and user interactable objects such as the sticks, with the LOD 3 assets being processed and overlayed on to the user's perspective, allowing for a richer fidelity without impacting local processing locally, and thereby reducing latency.

This pipeline would align well with the Modular 3D GUI designed in Chapter 3, where the user interface is close to the user when required and rendering simple primitive shapes to reduce processing requirements.

Bibliography

7.1 References

[1] C. Green, Y. Jiang, and J. Isaacs, “Modular 3D Interface Design for Accessible VR Applications,” pp. 15–32, 2023, doi: 10.1007/978-3-031-35634-6_2/TABLES/2.

[2] C. Green, D. Y. Jiang, D. J. Isaacs, and D. M. Heron, “Tremor Reduction for Accessible Ray Based Interaction in VR Applications,” May 2024, Accessed: May 28, 2024. [Online]. Available: https://arxiv.org/abs/2405.07335v1

[3] C. Green, “Augmenting Heritage: An Open-Source Multiplatform AR Application,” Sep. 2023, doi: 10.3390/APP12199859.

[4] C. Green, C. Farr, Y. Jiang, J. Isaacs, and R. Gordon, “An Accessible Toolkit for 360 VR Studies,” Apr. 2023, doi: 10.1016/B978-0-12-818697-8.00006-6.

[5] W. Barfield and D. Zeltzer, “Presence and Performance Within Virtual Environments,” *Virtual Environments and Advanced Interface Design*, Jul. 1995, doi: 10.1093/OSO/9780195075557.003.0023.

[6] A. Steed, T. M. Takala, D. Archer, W. Lages, and R. W. Lindeman, “Directions for 3D User Interface Research from Consumer VR Games,” *IEEE Trans Vis Comput Graph*, vol. 27, no. 11, pp. 4171–4182, Jun. 2021, doi: 10.48550/arxiv.2106.12633.

[7] M. P. Seevinck, “Holism, Physical Theories and Quantum Mechanics,” *Studies in History and Philosophy of Science Part B - Studies in History and Philosophy of Modern Physics*, vol. 35, no. 4, pp. 693–712, Feb. 2004, doi: 10.1016/j.shpsb.2004.08.001.

[8] J. Fromm, J. Radianti, C. Wehking, S. Stieglitz, T. A. Majchrzak, and J. vom Brocke, “More than experience? - On the unique opportunities of virtual reality to afford a holistic experiential learning cycle,” *Internet High Educ*, vol. 50, p. 100804, Jun. 2021, doi: 10.1016/J.IHEDUC.2021.100804.

[9] R. Abbasi-Asl, M. Keshavarzi, and D. Y. Chan, “Brain-Computer Interface in Virtual Reality,” *International IEEE/EMBS Conference on Neural Engineering, NER*, vol. 2019-March, pp. 1220–1224, May 2019, doi: 10.1109/NER.2019.8717158.

[10] P. Millais, S. Jones, and R. Kelly, “Exploring Data in Virtual Reality: Comparisons with 2D Data Visualizations,” *Conference on Human Factors in Computing Systems*. pp. 1–6, 2018. doi: 10.1145/3170427.3188537.

[11] P. K. Pisharady and M. Saerbeck, “Recent methods and databases in vision-based hand gesture recognition: A review,” *Computer Vision and Image Understanding*, vol. 141, pp. 152–165, 2015, doi: 10.1016/j.cviu.2015.08.004.

[12] E. Olshannikova, A. Ometov, Y. Koucheryavy, and T. Olsson, “Visualizing Big Data with augmented and virtual reality: challenges and research agenda,” *J Big Data*, vol. 2, no. 1, pp. 1–27, Dec. 2015, doi: 10.1186/S40537-015-0031-2/FIGURES/14.

[13] J. M. Loomis, R. L. Klatzky, and N. A. Giudice, “Representing 3D Space in Working Memory: Spatial Images from Vision, Hearing, Touch, and Language.” pp. 131–155, 2012. doi: 10.1007/978-1-4614-5879-1_8.

[14] A. H. Robinson, *The Nature of Maps*, 1st ed. Chicago: University of Chicago Press, 1976.

[15] Google, “Google Maps Augmented Reality & Virtual Reality,” vol. 2021, no. 03/05/2021. Jan. 2021.

[16] F. Windhager *et al.*, “Visualization of Cultural Heritage Collection Data: State of the Art and Future Challenges,” *IEEE Transactions on Visualization and Computer Graphics; IEEE Trans Vis Comput Graph*, vol. 25, no. 6, pp. 2311–2330, 2019, doi: 10.1109/TVCG.2018.2830759.

[17] E. Segel and J. Heer, “Narrative Visualization: Telling Stories with Data,” *IEEE Transactions on Visualization and Computer Graphics; IEEE Trans Vis Comput Graph*, vol. 16, no. 6, pp. 1139–1148, 2010, doi: 10.1109/TVCG.2010.179.

[18] Tableau, “Tableau,” vol. 2021, no. 03/05/2021. 2021.

[19] G. Schall, S. Zollmann, G. Reitmayr, “Smart Vidente: advances in mobile augmented reality for interactive visualization of underground infrastructure,” *Pers Ubiquitous Comput*, vol. 17, no. 7, pp. 1533–1549, 2013, doi: 10.1007/s00779-012-0599-x.

[20] M. N. Varga, R. Merrison-Hort, P. Watson, R. Borisyuk, and D. Livingstone, “Tadpole VR: virtual reality visualization of a simulated tadpole spinal cord,” *Virtual reality: the journal of the Virtual Reality Society*, vol. 25, no. 1, pp. 1–17, 2021, doi: 10.1007/s10055-020-00431-z.

[21] C.-Y. Lin, K.-L. Tsai, S.-C. Wang, C.-H. Hsieh, H.-M. Chang, and A.-S. Chiang, “The Neuron Navigator: Exploring the information pathway through the neural maze,” *PACIFICVIS*. pp. 35–42, 2011. doi: 10.1109/PACIFICVIS.2011.5742370.

[22] W. Usher *et al.*, “A Virtual Reality Visualization Tool for Neuron Tracing,” *IEEE Transactions on Visualization and Computer Graphics; IEEE Trans Vis Comput Graph*, vol. 24, no. 1, pp. 994–1003, 2018, doi: 10.1109/TVCG.2017.2744079.

[23] F. Bonato, A. Bubka, S. Palmisano, D. Phillip, and G. Moreno, “Vection Change Exacerbates Simulator Sickness in Virtual Environments,” *Presence (Camb)*, vol. 17, no. 3, pp. 283–292, 2008, doi: 10.1162/pres.17.3.283.

[24] C. Donalek *et al.*, “Immersive and Collaborative Data Visualization Using Virtual Reality Platforms.” 2014. doi: 10.1109/BigData.2014.7004282.

[25] H. Kharoub, M. Lataifeh, and N. Ahmed, “3D User Interface Design and Usability for Immersive VR,” *Applied sciences*, vol. 9, no. 22, p. 4861, 2019, doi: 10.3390/app9224861.

[26] J. Cha, J. Kim, and S. Kim, “Hands-Free User Interface for AR/VR Devices Exploiting Wearer’s Facial Gestures Using Unsupervised Deep Learning,” *Sensors (Basel, Switzerland); Sensors (Basel)*, vol. 19, no. 20, p. 4441, 2019, doi: 10.3390/s19204441.

[27] Z. Li, M. Kiiveri, J. Rantala, and R. Raisamo, “Evaluation of haptic virtual reality user interfaces for medical marking on 3D models,” *Int J Hum Comput Stud*, vol. 147, 2021, doi: 10.1016/j.ijhcs.2020.102561.

[28] R. Wedoff, L. Ball, A. Wang, Y. X. Khoo, L. Lieberman, and K. Rector, “Virtual Showdown: An Accessible Virtual Reality Game with Scaffolds for Youth with Visual Impairments,” *Conference on Human Factors in Computing Systems*. pp. 1–15, 2019. doi: 10.1145/3290605.3300371.

[29] A. Chu, “VR Design: Transitioning from a 2D to a 3D Design Paradigm,” vol. 2021, no. 05/05/2021. Jan. 2014.

[30] C. McKenzie, “Designing Screen Interfaces for VR,” vol. 2021, no. 05/05/2021. Jan. 2017.

[31] D. R. Olsen, “Evaluating user interface systems research,” *Symposium on User Interface Software and Technology*. pp. 251–258, 2007. doi: 10.1145/1294211.1294256.

[32] D. Harper, “Talking about pictures: A case for photo elicitation,” *https://doi.org/10.1080/14725860220137345*, vol. 17, no. 1, pp. 13–26, Jan. 2010, doi: 10.1080/14725860220137345.

[33] B. S, “Approaches to the successful design and implementation of VR applications.,” 1st ed., R. A. Earnshaw, Ed., in Virtual Reality Applications. , London: Academic Press Ltd., 1995, pp. 3–15.

[34] B. R. Blais, “Visual ergonomics of the office workplace,” *Chem Health Saf*, vol. 6, no. 4, pp. 31–38, Jul. 1999, doi: https://doi.org/10.1016/S1074-9098(99)00013-1.

[35] A. Sutcliffe, *Multimedia and Virtual Reality: Designing Multisensory User Interfaces*, 1st ed. Psychology Press, 2002.

[36] W. H. Organization, “World Report on Disability,” vol. 2021, no. 01/07. 2011.

[37] H. J. Chae, J. Hwang, and J. Seo, “Wall-based Space Manipulation Technique for Efficient Placement of Distant Objects in Augmented Reality,” *Symposium on User Interface Software and Technology*. pp. 45–52, 2018. doi: 10.1145/3242587.3242631.

[38] H. Ro, J.-H. Byun, and Y. J. Park, “AR Pointer: Advanced Ray-Casting Interface Using Laser Pointer Metaphor for Object Manipulation in 3D Augmented Reality Environment.” In Applied Sciences 2019.

[39] M. Mott, “Accessible by Design: An Opportunity for Virtual Reality.” In 2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct.

[40] Oculus, “Introducing the Accessibility VRCs,” vol. 2021, no. 02/07/2021. 2020.

[41] J. Cornacchioli *et al*, “Virtual Reality tremor reduction in Parkinson’s disease.” 2020.

[42] R. Balakrishnan, “‘Beating’ Fitts’ law: virtual enhancements for pointing facilitation,” *Int J Hum Comput Stud*, vol. 61, no. 6, pp. 857–874, 2004, doi: 10.1016/j.ijhcs.2004.09.002.

[43] J. Rantala, J. Kangas, O. Koskinen, T. Nukarinen, and R. Raisamo, “Comparison of Controller-Based Locomotion Techniques for Visual Observation in Virtual Reality,” *Multimodal Technologies and Interaction 2021, Vol. 5, Page 31*, vol. 5, no. 7, p. 31, Jun. 2021, doi: 10.3390/MTI5070031.

[44] M. Martez, J. Tang, and S. Kane, “I just went into it assuming that I wouldn’t be able to have the full experience”: Understanding the Accessibility of Virtual Reality for People with Limited Mobility.” 2020.

[45] K. Gerling and K. Spiel, “A critical examination of virtual reality technology in the context of the minority body,” *Conference on Human Factors in Computing Systems - Proceedings*, May 2021, doi: 10.1145/3411764.3445196.

[46] L. Evans, “Barriers to VR use in HE,” pp. 3–13, 2017, doi: 10.1255/vrar2018.ch2.

[47] J. Wentzel, S. Junuzovic, J. Devine, J. Porter, and M. Mott, “Understanding How People with Limited Mobility Use Multi-Modal Input,” pp. 1–17, Apr. 2022, doi: 10.1145/3491102.3517458.

[48] B. Smith *et al.*, “Constraining dense hand surface tracking with elasticity,” *ACM Trans Graph*, vol. 39, no. 6, pp. 1–14, 2020, doi: 10.1145/3414685.3417768.

[49] B. L. de Freitas *et al.*, “Analysis of Different Device Interactions in a Virtual Reality Task in Individuals With Duchenne Muscular Dystrophy-A Randomized Controlled Trial,” *Frontiers in neurology; Front Neurol*, vol. 10, p. 24, 2019, doi: 10.3389/fneur.2019.00024.

[50] “Half-Life: Alyx on Steam.” Accessed: May 27, 2022. [Online]. Available: https://store.steampowered.com/app/546560/HalfLife_Alyx/

[51] “VRChat.” Accessed: Apr. 29, 2024. [Online]. Available: https://hello.vrchat.com/

[52] C. Joshue, H. Sajka, and *et al*, “XR Accessibility User Requirements,” vol. 2021, no. July 07. 2020 https://www.w3.org/TR/xaur/.

[53] A. Siu, M. Sinclair, R. Kovacs, E. Ofek, C. Holz, and E. Cutrell, “Virtual Reality Without Vision: A Haptic and Auditory White Cane to Navigate Complex Virtual Worlds,” *Conference on Human Factors in Computing Systems*. pp. 1–13, 2020. doi: 10.1145/3313831.3376353.

[54] “BDA Dyslexia Style Guide,” 2014.

[55] T. Grossman, D. Wigdor, and R. Balakrishnan, “Exploring and reducing the effects of orientation on text readability in volumetric displays,” *Conference on Human Factors in Computing Systems - Proceedings*, pp. 483–492, 2007, doi: 10.1145/1240624.1240702.

[56] K. Larson, M. van Dantzich, M. Czerwinski, and G. Robertson, “Text in 3D: Some legibility results,” *Conference on Human Factors in Computing Systems - Proceedings*, pp. 145–146, 2000, doi: 10.1145/633292.633374.

[57] H. Cai and P. A. Green, “Legibility index for examining common viewing situations: A new definition using solid angle,” *LEUKOS - Journal of Illuminating Engineering Society of North America*, vol. 5, no. 4, pp. 279–295, Apr. 2009, doi: 10.1582/LEUKOS.2008.05.04.002.

[58] “File:Estimating angular size with hand.gif - Wikipedia.” Accessed: Mar. 22, 2022. [Online]. Available: https://en.wikipedia.org/wiki/File:Estimating_angular_size_with_hand.gif

[59] M. Alger, “VR Interface Design Pre-Visualisation Methods,” vol. 2021, no. 27/07/2021. 2015.

[60] W. B. Thompson, V. Dilda, and S. H. Creem-Regehr, “Absolute distance perception to locations off the ground plane,” *Perception*, vol. 36, no. 11, pp. 1559–1571, 2007, doi: 10.1068/P5667.

[61] M. B. Şenol, “Anthropometric evaluation of cockpit designs,” *Int J Occup Saf Ergon*, vol. 22, no. 2, pp. 246–256, Apr. 2016, doi: 10.1080/10803548.2015.1126456.

[62] H. S. Singer, J. W. Mink, D. L. Gilbert, and J. Jankovic, “Tremor,” *Movement Disorders in Childhood*, pp. 129–138, Jan. 2010, doi: 10.1016/B978-0-7506-9852-8.00012-6.

[63] J. Blackburn and M. Parnes, “Tics, tremors and other movement disorders in childhood,” *Curr Probl Pediatr Adolesc Health Care*, vol. 51, no. 3, p. 100983, Mar. 2021, doi: 10.1016/J.CPPEDS.2021.100983.

[64] J. S. Lora-Millan, G. Delgado-Oleas, J. Benito-León, and E. Rocon, “A Review on Wearable Technologies for Tremor Suppression,” *Front Neurol*, vol. 12, p. 1253, Aug. 2021, doi: 10.3389/FNEUR.2021.700600/BIBTEX.

[65] J. K. Nelson, “Mitigating the effects of intersymbol interference: algorithms and analysis,” 2005.

[66] R. J. Elble, “Characteristics of physiologic tremor in young and elderly adults,” *Clinical Neurophysiology*, vol. 114, no. 4, pp. 624–635, Apr. 2003, doi: 10.1016/S1388-2457(03)00006-3.

[67] A. Lenka and J. Jankovic, “Tremor Syndromes: An Updated Review,” *Front Neurol*, vol. 12, p. 1280, Jul. 2021, doi: 10.3389/FNEUR.2021.684835/BIBTEX.

[68] G. Casiez, N. Roussel, and D. Vogel, “1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems,” *Conference on Human Factors in Computing Systems*. pp. 2527–2530, 2012. doi: 10.1145/2207676.2208639.

[69] “corriedotdev/OneEuroFilterUnity-VR: One Euro Filter Implementation for VR with Accessibility guidelines.” Accessed: Aug. 03, 2022. [Online]. Available: https://github.com/corriedotdev/OneEuroFilterUnity-VR

[70] “DarioMazzanti/OneEuroFilterUnity: OneEuroFilter utility for Unity.” Accessed: Aug. 03, 2022. [Online]. Available: https://github.com/DarioMazzanti/OneEuroFilterUnity

[71] A. V. Zakharov *et al.*, “Proprioception in Immersive Virtual Reality,” *Proprioception*, Mar. 2021, doi: 10.5772/INTECHOPEN.96316.

[72] W. Stuerzlinger and C. A. Wingrave, “The value of constraints for 3d user interfaces,” *Virtual Realities: Dagstuhl Seminar 2008*, pp. 203–223, 2011, doi: 10.1007/978-3-211-99178-7_11/COVER.

[73] S. Noël, S. Dumoulin, T. Whalen, M. Ward, J. A. Stewart, and E. Lee, “A breeze enhances presence in a virtual environment,” *Proceedings - 3rd IEEE International Workshop on Haptic, Audio and Visual Environments and their Applications - HAVE 2004*, pp. 63–68, 2004, doi: 10.1109/HAVE.2004.1391883.

[74] I. E. Sutherland, “The Ultimate Display”.

[75] M. T. Atta, A. Romli, and M. A. Majid, “The Impact of AR/VR on Spatial Memory Performance of Learners: A review,” *Proceedings - 2021 International Conference on Software Engineering and Computer Systems and 4th International Conference on Computational Science and Information Management, ICSECS-ICOCSIM 2021*, pp. 75–79, Aug. 2021, doi: 10.1109/ICSECS52883.2021.00021.

[76] D. A. Bowman, R. P. McMahan, and E. D. Ragan, “Questioning naturalism in 3D user interfaces,” *Commun ACM*, vol. 55, no. 9, pp. 78–88, Sep. 2012, doi: 10.1145/2330667.2330687.

[77] “(PDF) The Efficiency of Various Multimodal Input Interfaces Evaluated in Two Empirical Studies.” Accessed: Feb. 10, 2023. [Online]. Available: https://www.researchgate.net/publication/49392564_The_Efficiency_of_Various_Multimodal_Input_Interfaces_Evaluated_in_Two_Empirical_Studies

[78] “Understanding Guideline 2.5: Input Modalities | WAI | W3C.” Accessed: Jan. 06, 2023. [Online]. Available: https://www.w3.org/WAI/WCAG21/Understanding/input-modalities

[79] R. J. Teather, A. Pavlovych, W. Stuerzlinger, and I. S. MacKenzie, “Effects of tracking technology, latency, and spatial jitter on object movement,” *3DUI - IEEE Symposium on 3D User Interfaces 2009 - Proceedings*, pp. 43–50, 2009, doi: 10.1109/3DUI.2009.4811204.

[80] R. J. Teather and W. Stuerzlinger, “Assessing the effects of orientation and device on (constrained) 3D movement techniques,” *3DUI - IEEE Symposium on 3D User Interfaces 2008*, pp. 43–50, 2008, doi: 10.1109/3DUI.2008.4476590.

[81] H. Ro, J.-H. Byun, and Y. J. Park, “AR Pointer: Avanced Ray-Casting Interface Using Laser Pointer Metaphor for Object Manipulation in 3D Augmented Reality Environment.” 2019.

[82] “Inside Facebook Reality Labs: Wrist-based interaction for the next computing platform.” Accessed: Sep. 06, 2022. [Online]. Available: https://tech.fb.com/ar-vr/2021/03/inside-facebook-reality-labs-wrist-based-interaction-for-the-next-computing-platform/

[83] P. Piotrowski and A. Nowosielski, “Gaze-Based Interaction for VR Environments,” *Advances in Intelligent Systems and Computing*, vol. 1062, pp. 41–48, 2020, doi: 10.1007/978-3-030-31254-1_6/FIGURES/5.

[84] C. Summers, V. Lympouridis, and C. Erkut, “Sonic interaction design for virtual and augmented reality environments,” in *2015 IEEE 2nd VR Workshop on Sonic Interactions for Virtual Environments (SIVE)*, IEEE, Mar. 2015, pp. 1–6. doi: 10.1109/SIVE.2015.7361290.

[85] J. A. Brown, “An Exploration of Virtual Reality Use and Application Among Older Adult Populations,” *Gerontol Geriatr Med*, vol. 5, p. 233372141988528, Jan. 2019, doi: 10.1177/2333721419885287.

[86] M. Recenti *et al.*, “Toward Predicting Motion Sickness Using Virtual Reality and a Moving Platform Assessing Brain, Muscles, and Heart Signals,” *Front Bioeng Biotechnol*, vol. 9, p. 132, Apr. 2021, doi: 10.3389/FBIOE.2021.635661/BIBTEX.

[87] K. A. Mills, L. Scholes, and A. Brown, “Virtual Reality and Embodiment in Multimodal Meaning Making,” *Writ Commun*, vol. 39, no. 3, pp. 335–369, 2022, doi: 10.1177/07410883221083517.

[88] J. Wang, R. Shi, W. Zheng, W. Xie, D. Kao, and H. N. Liang, “Effect of Frame Rate on User Experience, Performance, and Simulator Sickness in Virtual Reality,” *IEEE Trans Vis Comput Graph*, vol. 29, no. 5, pp. 2478–2488, May 2023, doi: 10.1109/TVCG.2023.3247057.

[89] “Testing and Performance Analysis | Oculus Developers.” Accessed: Nov. 26, 2023. [Online]. Available: https://developer.oculus.com/documentation/unity/unity-perf/

[90] A. Dixit and S. R. Sarangi, “Minimizing the Motion-to-Photon-delay (MPD) in Virtual Reality Systems” [Online]. Available at Arxiv: https://arxiv.org/abs/2301.10408.

[91] “(PDF) Manual Pursuit Camera and a Method of Verifying Pursuit Accuracy - slides.” Accessed: Aug. 07, 2023. [Online]. Available: https://www.researchgate.net/publication/281117238_Manual_Pursuit_Camera_and_a_Method_of_Verifying_Pursuit_Accuracy_-_slides

[92] C. W. Tyler, "Analysis of visual modulation sensitivity. III. Meridional variations in peripheral flicker sensitivity," *J Opt Soc Am A*, vol. 4, no. 8, p. 1612, Aug. 1987, doi: 10.1364/JOSAA.4.001612.

[93] F. Peng, H. Chen, and S.-T. Wu, "Can LCDs Outperform OLED Displays in Motion Picture Response Time?," 2017, doi: 10.1002/sdtp.11657.

[94] A. Dayal, C. Woolley, B. Watson, and D. Luebke, "Adaptive Frameless Rendering," *Eurographics Symposium on Rendering*, 2005.

[95] A. Dayal, S. Gupta, S. Ponnada, and D. J. Hemanth, "Coherence and Adaptivity in Frameless Rendering-A Practical and Information Theoretic Analysis," *IEEE Access*, vol. 9, pp. 67752–67760, 2021, doi: 10.1109/ACCESS.2021.3076898.

[96] S. Friston, A. Steed, S. Tilbury, and G. Gaydadjiev, "Construction and Evaluation of an Ultra Low Latency Frameless Renderer for VR," *IEEE Trans Vis Comput Graph*, vol. 22, no. 4, pp. 1377–1386, Apr. 2016, doi: 10.1109/TVCG.2016.2518079.

[97] R. Albert, A. Patney, D. Luebke, J. Kim, "25 Latency Requirements for Foveated Rendering in Virtual Reality," *ACM Trans. Appl. Percept*, vol. 14, no. 25, 2017, doi: 10.1145/3127589.

[98] M. R. Ibbotson and S. L. Cloherty, "Visual Perception: Saccadic Omission — Suppression or Temporal Masking?," *Current Biology*, vol. 19, no. 12, pp. R493–R496, Jun. 2009, doi: 10.1016/J.CUB.2009.05.010.

[99] L. Dong, W. Lin, C. Zhu, and H. S. Seah, "Selective rendering with graphical saliency model," *2011 IEEE 10th IVMSP Workshop: Perception and Visual Signal Analysis, IVMSP 2011 - Proceedings*, pp. 159–164, 2011, doi: 10.1109/IVMSPW.2011.5970372.

[100] J. Iskander, M. Hossny, and S. Nahavandi, "Using biomechanics to investigate the effect of VR on eye vergence system," *Appl Ergon*, vol. 81, p. 102883, Nov. 2019, doi: 10.1016/J.APERGO.2019.102883.

[101] Y. Su, J. Lin, S. Chen, S.-T. Wu, Z. He, and Y. Li, "Planar Alvarez tunable lens based on polymetric liquid crystal Pancharatnam-Berry optical elements," *Optics Express, Vol. 30, Issue 19, pp. 34655-34664*, vol. 30, no. 19, pp. 34655–34664, Sep. 2022, doi: 10.1364/OE.468647.

[102] "About importing and simulating a phase profile of a DOE lens | Zemax Community." Accessed: Jul. 31, 2023. [Online]. Available: https://community.zemax.com/people-pointers-9/about-importing-and-simulating-a-phase-profile-of-a-doe-lens-1186

[103] A. Jamali, D. Bryant, A. K. Bhowmick, and P. J. Bos, "Large area liquid crystal lenses for correction of presbyopia," *Opt Express*, vol. 28, no. 23, p. 33982, Nov. 2020, doi: 10.1364/OE.408770.

[104] J. F. Algorri, N. Bennis, V. Urruchi, P. Morawiak, J. M. Sánchez-Pena, and L. R. Jaroszewicz, "Tunable liquid crystal multifocal microlens array," *Scientific Reports 2017 7:1*, vol. 7, no. 1, pp. 1–6, Dec. 2017, doi: 10.1038/s41598-017-17688-1.

[105] Y. Zhao *et al.*, "Retinal-Resolution Varifocal VR," *ACM SIGGRAPH 2023 Emerging Technologies*, pp. 1–3, Jul. 2023, doi: 10.1145/3588037.3595389.

[106] Z. Xia, Y. Zhang, F. Ma, C. Cheng, and F. Hu, "Effect of spatial distortions in head-mounted displays on visually induced motion sickness," *Opt Express*, vol. 31, no. 2, p. 1737, 2023, doi: 10.1364/OE.478455.

[107] "Perceptual Requirements for Eye-Tracked Distortion Correction in VR - Meta Research." Accessed: Jul. 22, 2023. [Online]. Available: https://research.facebook.com/publications/perceptual-requirements-for-eye-tracked-distortion-correction-in-vr/

[108] "Considering the effects of display persistence on eye movements and readability in virtual reality - Meta Research." Accessed: Aug. 02, 2023. [Online]. Available: https://research.facebook.com/publications/considering-the-effects-of-display-persistence-on-eye-movements-and-readability-in-virtual-reality/

[109] D. Scott Dewald, A. T. Evans, N. Welch, A. Gross, and G. Hill, "The Avegant Glyph: Optical Design Considerations and Approach to Near-eye Display," *Dig Tech Pap*, vol. 47, no. 1, pp. 69–71, 2016, doi: 10.1002/SDTP.10609.

[110] "The LaSAR Alliance - Laser Scanning for Augmented Reality." Accessed: Aug. 02, 2023. [Online]. Available: https://lasaralliance.org/

[111] Y. Zhang and F. Fang, "Development of planar diffractive waveguides in optical see-through head-mounted displays," *Precis Eng*, vol. 60, pp. 482–496, Nov. 2019, doi: 10.1016/J.PRECISIONENG.2019.09.009.

[112] "Holographic Glasses for Virtual Reality | Research." Accessed: Aug. 05, 2023. [Online]. Available: https://research.nvidia.com/publication/2022-08_holographic-glasses-virtual-reality

[113] H. Gao *et al.*, "Holographic three-dimensional virtual reality and augmented reality display based on 4K-spatial light modulators," *Applied Sciences (Switzerland)*, vol. 9, no. 6, 2019, doi: 10.3390/APP9061182.

[114] "PLUTO-2 PHASE ONLY SPATIAL LIGHT MODULATOR (REFLECTIVE) | ONSET." Accessed: Sep. 22, 2023. [Online]. Available: https://www.onset-eo.com/product/pluto-2-phase-only-spatial/

[115] R. S. Overbeck, D. Erickson, D. Evangelakos, and P. Debevec, "The making of welcome to light fields VR," pp. 1–2, Aug. 2018, doi: 10.1145/3214745.3214811.

[116] M. Levo and P. Hanrahan, "Light Field Rendering," 1996, Accessed: Jul. 22, 2023. [Online]. Available: http://www-graphics.stanford.edu

[117] A. Gershun, "The Light Field," *Journal of Mathematics and Physics*, vol. 18, no. 1–4, pp. 51–151, Apr. 1939, doi: 10.1002/SAPM193918151.

[118] P. A. Kara and A. Simon, "The Good News, the Bad News, and the Ugly Truth: A Review on the 3D Interaction of Light Field Displays," *Multimodal Technologies and Interaction 2023, Vol. 7, Page 45*, vol. 7, no. 5, p. 45, Apr. 2023, doi: 10.3390/MTI7050045.

[119] D. Lanman and D. Luebke, "Near-Eye Light Field Displays", Accessed: Oct. 05, 2023. [Online]. Available: http://g.co/projectglass

[120] "Virtual Reality in focus I CREAL light-field display." Accessed: Jul. 22, 2023. [Online]. Available: https://creal.com/vr/

[121] Y. Zhou, J. Zhang, and F. Fang, "Vergence-accommodation conflict in optical see-through display: review and prospect," *Results in Optics*, vol. 5, p. 100160, Dec. 2021, doi: 10.1016/J.RIO.2021.100160.

[122] "Immersive 3D for the masses. Holograms and Conversational AI Characters." Accessed: Oct. 05, 2023. [Online]. Available: https://lookingglassfactory.com/

[123] G. Kuo, E. Penner, S. Moczydlowski, A. Ching, D. Lanman, and N. Matsuda, "Perspective-Correct VR Passthrough Without Reprojection," *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*, pp. 1–9, Jul. 2023, doi: 10.1145/3588432.3591534.

[124] "Pixel Arcade Legacy on Steam." Accessed: Oct. 06, 2023. [Online]. Available: https://store.steampowered.com/app/618570/Pixel_Arcade_Legacy/

[125] "PID Control of Physics Bodies - Math and Physics - Tutorials - GameDev.net." Accessed: Jul. 31, 2023. [Online]. Available: https://www.gamedev.net/tutorials/programming/math-and-physics/pid-control-of-physics-bodies-r3885/

[126] "GitHub - corriedotdev/vr-360-player." Accessed: Nov. 01, 2023. [Online]. Available: https://github.com/corriedotdev/vr-360-player

[127] G. Riva, "Virtual Reality in Clinical Psychology," *Comprehensive Clinical Psychology*, vol. 10, p. 91, Jan. 2022, doi: 10.1016/B978-0-12-818697-8.00006-6.

[128] D. Saffo, C. Yildirim, S. Di Bartolomeo, and C. Dunne, "Crowdsourcing virtual reality experiments using VRChat," *Conference on Human Factors in Computing Systems - Proceedings*, Apr. 2020, doi: 10.1145/3334480.3382829.

[129] "corriedotdev/3D-Museum-Library." Accessed: May 23, 2024. [Online]. Available: https://github.com/corriedotdev/3D-Museum-Library

[130] A. Shamei and B. Gick, "The effect of virtual reality environments on auditory memory," *J Acoust Soc Am*, vol. 148, no. 4_Supplement, pp. 2498–2498, Oct. 2020, doi: 10.1121/1.5146928.

[131] "XR Accessibility User Requirements." Accessed: Aug. 23, 2023. [Online]. Available: https://www.w3.org/TR/xaur/

[132] P. Delgado, C. Vargas, R. Ackerman, and L. Salmerón, "Don't throw away your printed books: A meta-analysis on the effects of reading media on reading comprehension," *Educ Res Rev*, vol. 25, pp. 23–38, Nov. 2018, doi: 10.1016/J.EDUREV.2018.09.003.

[133] R. McLaughlan, "Virtual reality as a research method: is this the future of photo-elicitation?," *Vis Stud*, vol. 34, no. 3, pp. 252–265, Jul. 2019, doi: 10.1080/1472586X.2019.1680315.

[134] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12346 LNCS, pp. 405–421, Mar. 2020, doi: 10.1007/978-3-030-58452-8_24.

[135] "Paper page - 3D Gaussian Splatting for Real-Time Radiance Field Rendering." Accessed: Sep. 19, 2023. [Online]. Available: https://huggingface.co/papers/2308.04079

[136] "<model-viewer>." Accessed: Jun. 06, 2023. [Online]. Available: https://modelviewer.dev/

[137] R. G. Boboc, E. Băutu, F. Gîrbacia, N. Popovici, and D. M. Popovici, “Augmented Reality in Cultural Heritage: An Overview of the Last Decade of Applications,” *Applied Sciences 2022, Vol. 12, Page 9859*, vol. 12, no. 19, p. 9859, Sep. 2022, doi: 10.3390/APP12199859.

[138] M. Whitelaw and M. Whitelaw, “Generous Interfaces for Digital Cultural Collections,” *Digital Humanities Quarterly*, vol. 009, no. 1, May 2015.

[139] “Highland Museum of Childhood .” Accessed: Jun. 06, 2023. [Online]. Available: https://highlandmuseumofchildhood.org.uk/visit-us/

[140] “Grantown Museum - Home.” Accessed: Jun. 06, 2023. [Online]. Available: https://www.grantownmuseum.co.uk/

[141] “Sketchfab - The best 3D viewer on the web.” Accessed: Jun. 06, 2023. [Online]. Available: https://sketchfab.com/

[142] “Fab | A Digital Asset Marketplace for Content Creators.” Accessed: Jun. 06, 2023. [Online]. Available: https://www.fab.com/

[143] “Agisoft Metashape: Agisoft Metashape.” Accessed: Jun. 19, 2023. [Online]. Available: https://www.agisoft.com/

[144] “Luma AI.” Accessed: Jun. 06, 2023. [Online]. Available: https://lumalabs.ai/

[145] “GitHub - facebookincubator/FBX2glTF: A command-line tool for the conversion of 3D model assets on the FBX file format to the glTF file format.” Accessed: Jun. 06, 2023. [Online]. Available: https://github.com/facebookincubator/FBX2glTF

[146] “VRChat.” Accessed: Jun. 19, 2023. [Online]. Available: https://hello.vrchat.com/

[147] M. v Eeckhaut *et al.*, “Use of LIDAR-derived images for mapping old landslides under forest,” *Earth Surface Processes and Landforms; Earth Surf.Process.Landforms*, vol. 32, no. 5, pp. 754–769, 2007, doi: 10.1002/esp.1417.

[148] Q. Feng, J. Liu, and J. Gong, “Urban Flood Mapping Based on Unmanned Aerial Vehicle Remote Sensing and Random Forest Classifier—A Case of Yuyao, China,” *Water (Basel)*, vol. 7, no. 12, pp. 1437–1455, 2015, doi: 10.3390/w7041437.

[149] S. Nasa, “Spacex CRS-6 Mission,” 2015.

[150] Y. Lu and G. Lu, “An Alternative of LiDAR in Nighttime: Unsupervised Depth Estimation Based on Single Thermal Image,” IEEE, 2021, pp. 3832–3842. doi: 10.1109/WACV48630.2021.00388.

[151] E. al Fernandez-Diaz Juan & Singhanic, “An overview of lidar point cloud processing software.” 2008.

[152] L. Liu, S. Lim, X. Shen, and M. Yebra, “A hybrid method for segmenting individual trees from airborne lidar data,” *Comput Electron Agric*, vol. 163, Aug. 2019, doi: 10.1016/J.COMPAG.2019.104871.

[153] W. Song, L. Zhang, Y. Tian, S. Fong, J. Liu, and A. Gozho, “CNN-based 3D object classification using Hough space of LiDAR point clouds,” *Human-centric Computing and Information Sciences*, vol. 10, no. 1, pp. 1–14, Dec. 2020, doi: 10.1186/S13673-020-00228-8/TABLES/2.

[154] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, “3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence; IEEE Trans Pattern Anal Mach Intell*, vol. 36, no. 11, pp. 2270–2287, 2014, doi: 10.1109/TPAMI.2014.2316828.

[155] S. A. Bello, S. Yu, and C. Wang, “Review: deep learning on 3D point clouds,” 2020.

[156] A. Dudek, “Silhouette index as clustering evaluation tool,” *Studies in Classification, Data Analysis, and Knowledge Organization*, pp. 19–33, 2020, doi: 10.1007/978-3-030-52348-0_2/COVER.

[157] K. J. O. Evans P.M. and M. L. Bjarnar, “New technology for subsea laser imaging and ranging system for ispection and mapping,” *Sensor Review*, vol. 18, no. 2, pp. 97–107, 1998.

[158] “Use Azure Kinect Sensor SDK to record file format | Microsoft Docs.” Accessed: Mar. 05, 2022. [Online]. Available: https://docs.microsoft.com/en-us/azure/kinect-dk/record-file-format

[159] USNA, “LAS Point Cloud Format.” 2021.

[160] Z. Du and Q. Li, “A NEW METHOD OF STORAGE AND VISUALIZATION FOR MASSIVE POINT CLOUD DATASET”.

[161] M. D. Niță, “Testing Forestry Digital Twinning Workflow Based on Mobile LiDAR Scanner and AI Platform,” *Forests*, vol. 12, no. 11, p. 1576, 2021.

[162] “(2) (PDF) VirtSilv A.I. 3D platform for sustainable forest management.” Accessed: May 23, 2024. [Online]. Available: https://www.researchgate.net/publication/357621898_VirtSilv_AI_3D_platform_for_sustainable_forest_management

[163] S. G. Kaufmann Viktor Kellerer-Pirklbauer Andreas, “Conventional and UAV-Based Aerial Surveys for Long-Term Monitoring (1954–2020) of a Highly Active Rock Glacier in Austria,” *Frontiers in Remote Sensing*, vol. 2, no. https://www.frontiersin.org/article/10.3389/frsen.2021.732744, 2021.

[164] R. R. Bisset, A. Dehecq, D. N. Goldberg, M. Huss, R. G. Bingham, and N. Gourmelen, “Reversed Surface-Mass-Balance Gradients on Himalayan Debris-Covered Glaciers Inferred from Remote Sensing,” *Remote Sensing 2020, Vol. 12, Page 1563*, vol. 12, no. 10, p. 1563, May 2020, doi: 10.3390/RS12101563.

[165] S. K. Card, T. P. Moran, and A. Newell, “The Keystroke-Level Model for User Performance Time with Interactive Systems,” *Commun ACM*, vol. 23, no. 7, pp. 396–410, Jul. 1980, doi: 10.1145/358886.358895.

[166] B. E. John and D. E. Kieras, “Using GOMS for user interface design and evaluation,” *ACM Transactions on Computer-Human Interaction*, vol. 3, no. 4, pp. 287–319, Dec. 1996, doi: 10.1145/235833.236050.

7.2 Appendix

Appendix A

```
public class OneEuroFilter {

    private float minCutoff;
    private float beta;
    private float dCutoff;

    private float xPrev;
    private float dxPrev;
    private float lastTime;

    public OneEuroFilter(float minCutoff = 1.0f, float beta = 0.0f,
float dCutoff = 1.0f) {
        this.minCutoff = minCutoff;
        this.beta = beta;
        this.dCutoff = dCutoff;
        this.xPrev = 0.0f;
        this.dxPrev = 0.0f;
        this.lastTime = -1.0f;
    }

    private float Alpha(float cutoff) {
        float te = Time.deltaTime;
        float tau = 1.0f / (2.0f * Mathf.PI * cutoff);
        return 1.0f / (1.0f + tau / te);
    }

    public float Filter(float x) {
        if (lastTime == -1.0f) {
            lastTime = Time.time;
            xPrev = x;
            return x;
        }

        float deltaTime = Time.time - lastTime;
        lastTime = Time.time;

        float dx = (x - xPrev) / deltaTime;
        float edx = Alpha(dCutoff) * (dx + (1.0f - Alpha(dCutoff)) *
dxPrev);
        float cutoff = minCutoff + beta * Mathf.Abs(edx);

        xPrev = x;
        dxPrev = edx;

        return Alpha(cutoff) * (x + (1.0f - Alpha(cutoff)) * xPrev);
    }
}
```

Appendix B

Vulkan based pseudocode for Frameless Eye-tracked foveated rendering

```
1. Initialize Vulkan:
      - Create Vulkan instance
      - Initialize GPU
      - Create a logical device to render to (VR Display dimensions
      etc.)
      - Setup a mechanism to send pixels / micro-frame directly to the
      display as they're rendered

2. Setup Eye Tracking:
      - Initialize eye-tracking hardware
      - Start eye-tracking data stream

3. Micro Frame Rendering V-SyncLoop:
      WHILE running DO
      - Poll eye-tracking data to get current gaze position, foveal
      center
      - Determine foveal and peripheral regions based on gaze position
      and determine the micro-frame to render
      - Start a new micro-frame for the selected foveal region:
      - Record commands into a Vulkan command buffer for the foveal
      region:
      - Set viewport and scissor for the foveal region with highest
      resolution
      - Start to render the foveal region with objects from the
      frustum
      - Output pixel when ready
            - Potential Area for Injection of distortion and other VR
            effects on a pixel   or micro-frame - Distortion
            correction for the foveal region pixels
      - Prediction Adjustment
      - As corrected pixels for the foveal region are ready, send them
      directly to the display
- Handle Vulkan synchronization as needed for semaphores, fences at
the end of the micro-frame render and output micro-frame to the
display

4. Cleanup V-Sync Cycle:
- Destroy Vulkan resources
- Cleanup eye-tracking resources
- Low persistence and potential injection of other VR distortion
requirements
- Return to step 2
```

Appendix C

```
Public class PIDController {

      public float Kp = 0f;
      public float Ki = 1f;
      public float Kd = 0f;

      private float previousError = 0f;
      private float integral = 0f;

      public PIDController(float Kp, float Ki, float Kd) {
            this.Kp = Kp;
             this.Ki = Ki;
             this.Kd = Kd;
      }

      public float Update(float setpoint, float actualValue, float
      deltaTime) {
             float error = setpoint - actualValue;

             integral += error * deltaTime;
             float derivative = (error - previousError) / deltaTime;

             float output = Kp * error + Ki * integral + Kd *
      derivative;

             previousError = error;

             return output;
      }
}
```

Appendix D

Graphic demonstrating the architecture for asynchronous loading of game scene worlds in VR using a memory addressable, minimizing or removing loading screens using a 3D portal interfaced between scenes

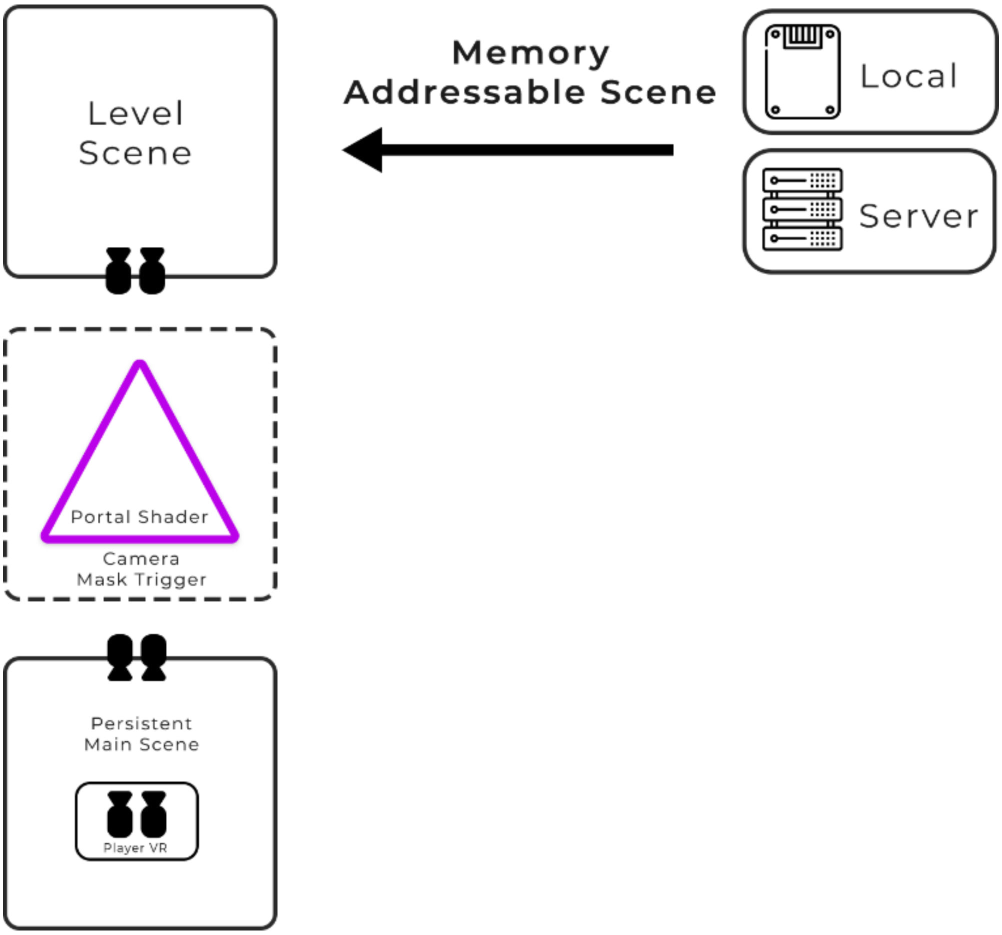


Figure 7.1 *Architecture diagram of asynchronous loading*

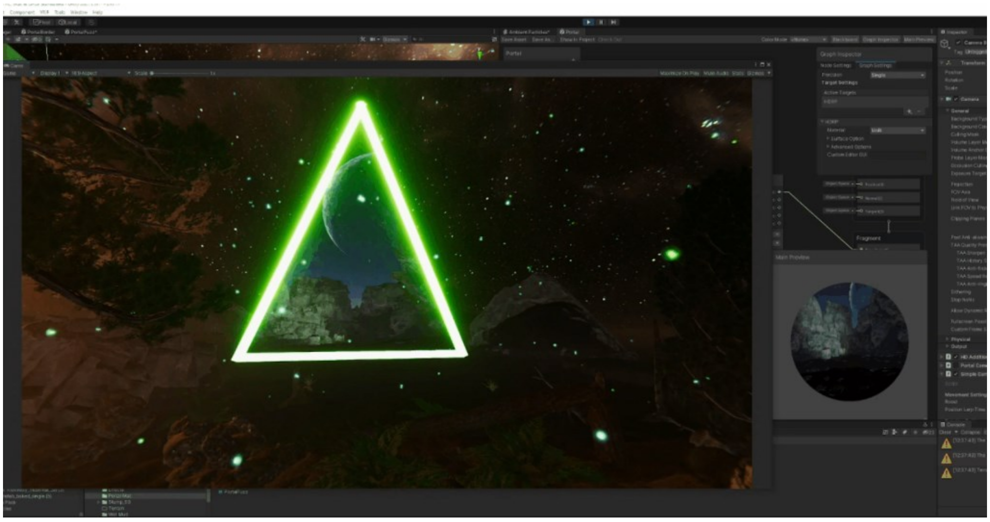


Figure 7.2 *Implementation of Figure 7.1 in Unity using a Stencil buffer portal*

Appendix E

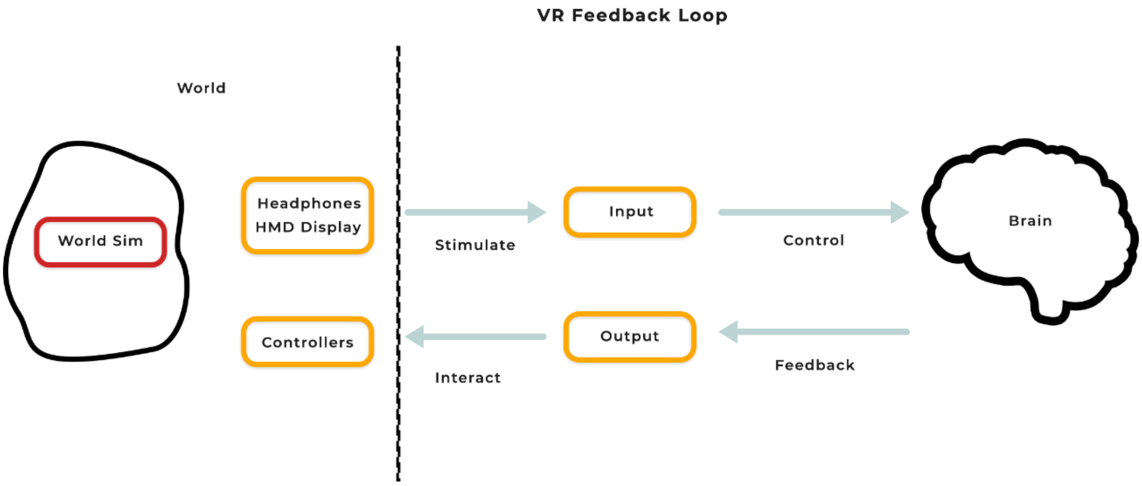


Figure 7.3 *Brain feedback loop flow for use in VR interaction specifically*