Particle Selection and Parameterisation in Virtual Reality

Boaz Keren Gil

Supervised by

Prof. James Gain - Department of Computer Science

Prof. Patrick Marais - Department of Computer Science

Dr. Lucia Marchetti - Department of Astronomy

Prof. Thomas Jarrett - - Department of Astronomy

Department of Computer Science

University of Cape Town

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

Master of Science in Computer Science

Abstract

This dissertation investigates the application of Virtual Reality (VR) technology in enhancing the selection and parameterisation of astronomical data compared to traditional desktop environments. Through the development and evaluation of the Immersive Data Visualisation Interactive Explorer for Particle Rendering (iDaVIE-p), a novel software application designed for both VR and desktop interfaces, this study explores VR's potential to improve accuracy, efficiency, and user experience in scientific research, particularly in astronomy, where datasets are large and complex. The primary focus of this research is to establish whether VR technology surpasses desktop environments in terms of task performance metrics such as accuracy, efficiency, usability, workload, and flow and how parameter tuning influences these metrics.

Our experimental design involves a hybrid of within and between-subject comparison, engaging participants in tasks that require selecting and adjusting parameters of celestial objects represented as particles. Participants utilised the iDaVIE-p software in both VR and desktop, providing feedback through established questionnaires like the System Usability Scale, NASA Task Load Index, and Flow State Scale.

The results indicate that while accuracy remained comparable between VR and desktop interfaces, VR significantly enhanced efficiency, with tasks completed 28% faster on average. Additionally, VR outperformed desktop in usability, workload, and flow metrics, evidencing a more engaging and less taxing experience. Surprisingly, these benefits were realized even among participants with limited VR experience, underscoring VR's intuitive interaction with three-dimensional (3D) environments. However, the study found mixed outcomes regarding parameter tuning, showing minor improvements in accuracy but a similar minor decrease in efficiency, suggesting a possible tradeoff between the two and that further research is needed to optimise parameter adjustments for task performance.

This dissertation underscores VR's transformative potential in scientific research, offering insights into its advantages over traditional desktop interfaces for interacting with complex, 3D datasets. The findings advocate for the broader adoption of VR technology in scientific settings where tasks and datasets are 3D, as was in our case, highlighting its capacity to enhance user satisfaction, efficiency, and engagement in data selection tasks.

Table of Contents

Chapter 1 - Introduction

1.1. Motivation

This dissertation explores the manipulation of astronomical data within the immersive realm of virtual reality (VR). Astronomical datasets cover very large volumes in the universe and contain many different kinds of objects of varying size and complexity, presenting unique challenges and opportunities for researchers and astronomers. Traditionally, these professionals have relied on desktop computers to handle such data, engaging primarily in tasks like selecting and adjusting the properties of various particles (parameterisation). In the context of astronomy, particles can represent celestial entities, most commonly stars or galaxies. Each particle is subject to selection (commonly known as manipulation) and modification. This modification typically focuses on the the particle's parameters such as size, colour, and shape, while other parameters such as the particle's position coordinates, are static. Of course, one can define different position axes for a particle rather than the space coordinates. For example, one can use the source's velocity as the Y-axis instead of the position Y coordinate. However, working with particles in a desktop environment poses certain limitations, primarily due to the lack of depth perception inherent in 2D interfaces like monitors, which are navigated via keyboard and mouse. This can lead to compromises in accuracy, efficiency, and overall user satisfaction when selecting particles.

These drawbacks are the principle motivation for this dissertation, which investigates the potential of VR technology to enhance the process of selecting and modifying particle data, with a focus on accuracy, efficiency, and user experience. VR has undergone significant advancements in recent years, opening up new possibilities not only in entertainment and education but also in scientific research. We examine the application of VR in scientific settings, particularly for interacting with three-dimensional particle datasets typical in astronomy. The culmination of our work is the Immersive Data Visualisation Interactive Explorer for Particle Rendering (iDaVIE-p), a novel software application designed for use in both VR and traditional desktop environments.

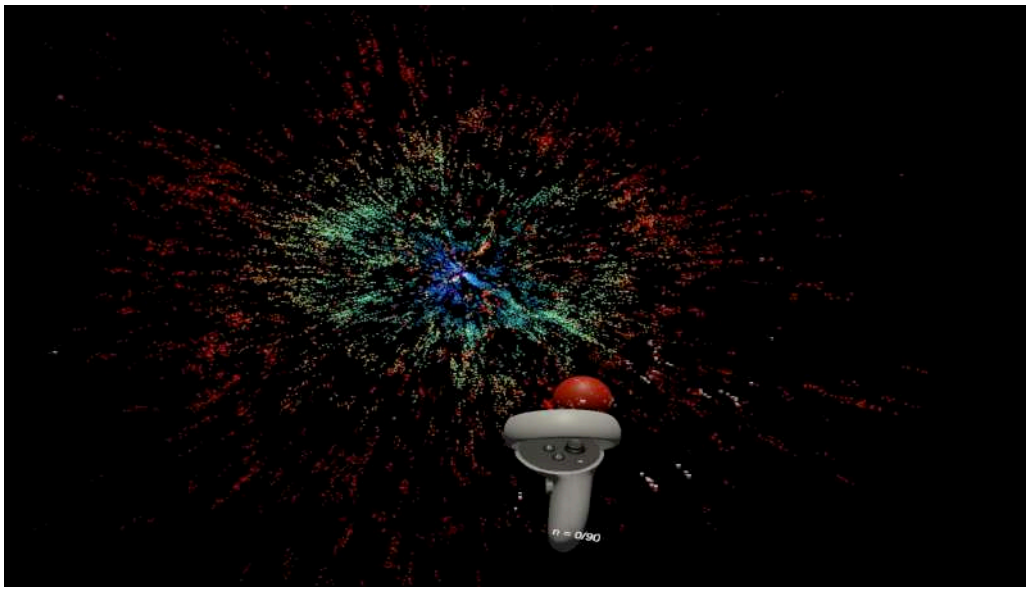


Fig. 1. iDaVIE-p in VR allows depth perception and the selection of particles

This dual-platform approach allows for a comprehensive comparison of user experiences, which is further enriched by analyses based on usability, workload, and flow questionnaires. iDaVIE-p enables users to navigate through datasets, select particles, and adjust their parameters within a VR space; offering a more intuitive understanding of depth and spatial relationships.

Our findings suggest that VR significantly surpasses traditional desktop environments in terms of efficiency and user satisfaction for the selection of astronomical data, while accuracy was not affected. Given the ubiquity of particle data across various scientific disciplines, the potential applications of VR in scientific research are vast and varied, beyond astronomy.

In the following section, we present our research questions and the basic definitions of Big Data, the Visualisation of different types of astronomical datasets and the development history of VR.

1.2. Research Questions

The following research questions have been defined for the scope of this work:

1. Is the selection (manipulation) of particles in Virtual Reality (VR) compared to desktop environments regarding accuracy, efficiency, usability, workload, and flow better?
2. How does tuning the parameters affect the accuracy, efficiency, usability, workload, and flow of selection tasks?

Flow, as described by the Flow State Questionnaire (Jackson & Marsh, 1996), refers to a highly focused and fully immersed state of consciousness where an individual experiences optimal engagement and performance in an activity. Flow characteristics include a strong sense of control, loss of self-consciousness, a distorted sense of time, and immediate feedback, leading to an intrinsically rewarding and enjoyable activity. This state is often achieved during tasks that balance the perceived challenges of the activity and the individual's perceived skills, resulting in heightened productivity and creativity. For a full definition of flow, please see Chapter 5.

1.3. Scope and Limitations

This thesis focuses solely on the development of iDaVIE-p and is limited in scope to the source code provided by the Inter-University Institute for Data Intensive Astronomy (IDIA) visualisation lab (IVL) at UCT [1]. In order to answer the research questions, the selection functionality must be developed and added to iDaVIE-p. This development stands at the core of this study, and is the prerequisite for it. We focus on selection specifically in this development, although iDaVIE-p can be further developed in many other directions. This selection in VR, as the literature in Chapter 3 shows, can include many types of selection, such as ray casting, selection cone, go-go, etc. We

[1] https://vislab.idia.ac.za/

developed a standard grab selection, often called the simple virtual hand selection. The scope of this work is to compare this selection in VR to the standard (mouse click) selection on a desktop. We did not compare the different selection methods in VR for their efficiency, as it is beyond the scope of this work. In addition, we limited the scope of development of the desktop version of iDaVIE-p, as we use this software only for the evaluation of the data and comparison and not for manipulation of production-ready datasets.

1.4. Overview of the Thesis

The thesis is structured as follows: Chapter 2 provides historical background to big data, astronomy and virtual reality. In addition, this chapter explores the connection between VR and astronomy in recent years, focusing on iDaVIE. In Chapter 3, we discuss the literature and related work done in the VR selection field alongside depth perception. Chapter 4 discusses the design and implementation of iDaVIE-p, with a focus on the design of the selection methods on VR and the comparison between them. Chapter 5 presents our experiment and evaluation strategy, comparing iDaVIE-p selection and parameterisation on desktop and VR. In Chapter 6, we present and discuss the results we received from the experiment and compare each metric (accuracy, efficiency, usability, workload and flow) separately. Lastly, in Chapter 7, we present our conclusions regarding the results and discuss where our developed version of iDaVIE-p can improve.

Chapter 2 - Background

This chapter lays out the background for this thesis. It includes a comparison between the different current solutions for the selection and manipulation of source catalogues, the different formats used to store source catalogues and notable solutions in VR for astronomy and specifically for source catalogue manipulation, selection and parameterisation. Lastly, this chapter discusses the iDaVIE software developed by UCT.

2.1. Big Data and Astronomy

Astronomy, as a field deeply intertwined with the study of vast celestial phenomena, naturally aligns with the concepts of Big Data. Big Data, first defined by John Mashey in 1999 (Mashey, 1999), represents data sets so large and complex that traditional data processing software struggles to manage them efficiently. In astronomy, this manifests in collecting and analysing enormous volumes of data from various celestial sources, posing challenges and opportunities.

Big Data is commonly characterised by four critical attributes (Dhamodharavadhani et al., 2018): volume (how much data is there), variety (how many data types are there), velocity (how real-time the data is), and veracity (how accurate the data is).

Processing Big Data in astronomy typically involves the use of processors or pipelines. These pipelines handle small batches of data, applying various tools and mathematical methods to transform the raw data into a more manageable and interpretable format. For instance, textual datasets containing information about celestial objects might be transformed into graphical representations, enabling astronomers to visualise and analyse the properties of objects in the astronomical catalogue in a more accessible and human-friendly interface.

Understanding the nature of the datasets and their potential visualisation is critical in astronomy. It involves profoundly comprehending the raw data's format and the techniques for converting it into visual or analytical models that can provide meaningful insights into the universe's workings.

In summary, the study of astronomy through the lens of Big Data involves grappling with very large volumes of diverse data, which are rapidly generated and must be processed accurately and reliably. The field's evolution is closely tied to advancements in data processing technologies, which enable astronomers to decode the mysteries of the cosmos more effectively than ever before.

2.2. Visualisation of Astronomical Datasets

Data visualisation aims to enhance and allow further scientific discovery (B.H. McCormick, 1987). This can be achieved by rendering shapes, volumes, particles, and surfaces (Friendly, 1995). Data visualisation specifically handles the manipulation and rendering of volume data sets (such as cubes), allowing complex structures in multiple dimensions to be easily understood by humans.

Visualisation of Volume Datasets

Astronomers use a spectral data cube made out of voxels to visualise volume data. This data structure is a three-dimensional (3D) layer representation of the data in a cube shape. Two axes define the angular sky coordinates, or the spatial dimension, while the third represents the spectral size. The spectral dimension is a real-valued dimension related to the line-of-sight velocity of an object through the Doppler effect. On the scale of the universe this velocity translates to a distance because of the expansion of the universe. Simply put, space coordinates are made from the position and spectral coordinates of each object. It is essential to mention that spectral data cubes are used in other fields of science and not just in astronomy (Kaufman, 1994).

Visualisation of Catalogue (particle) Datasets

Another way to view astronomy data from a macro perspective is by using a galaxy catalogue. In this catalogue, often called the sources catalogue, a source can be a galaxy, nebulae or star. Each source in the dataset table can be likened to a point in space where its features correspond to coordinates, forming a discrete spatial database. Each source has a multitude of dimensions or parameters. For example, the source's distance from the viewer determines the particle's colour, where blue means close by, and red further away - a phenomenon called "redshift" caused by the Doppler effect (de Sitter, 1934). The colour spectrum of blue and red arises from the expansion of the universe, and the wavelength of the light that travels to the observer, where red is a longer wavelength than blue, hence longer distances. The source's size can determine its mass, and the source's brightness can determine its luminosity. Of course, the astronomer can choose freely how each parameter is visualised.

Source catalogues can be created from various observation wavelengths, such as X-ray, visible light, infrared, radio frequencies, etc. Usually, a specific telescope, such as the Chandra X-Ray Observatory (Evans et al., 2010), focuses on one type of wavelength, e.g. X-ray, to produce a source catalogue, as seen in Fig 2.

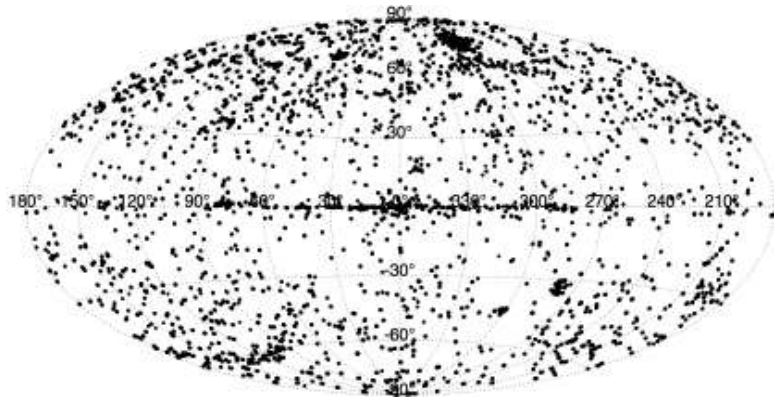


Fig. 2. A source catalogue of the sky from the Chandra X-Ray Observatory (Evans et al., 2010)

As previously mentioned, to create a source catalogue, the raw (big) data can be represented only after it goes through a pipeline that extracts vital features that interest the scientist. This automation is crucial, given the very large quantity of data involved, but manual inspection of the final aggregated product by the human eye is still usually needed. This inspection, typically called manipulation, includes an actual interaction with the data rather than passive viewing.

This dissertation focuses on source catalogues and their respective manipulation methods on desktop and virtual reality devices rather than spectral data cubes. This is because data cube manipulation is already a saturated topic in virtual reality and has some solutions, such as the Immersive Data Visualisation Interactive Explorer for Volume Rendering (iDaVIE-v), which will be presented later in this chapter. Therefore, we chose to focus on source catalogue manipulation in virtual reality. First, we must explore the different formats and datasets used in this work to understand further the required pipelines to transform and visualise the data.

2.3. History and Development of Virtual Reality

The evolution of Virtual Reality (VR) technology is rich in innovation and imagination that spans over a century, tracing its roots back to the 1800s. It began with the artistic endeavour to create immersive environments, such as panoramic paintings designed to engulf the viewer's vision. These early attempts at creating a sense of presence in a separate environment laid the groundwork for what would evolve into modern VR.

In the 19th century, Charles Wheatstone's invention of the stereoscope marked a pivotal moment in the history of VR (Wheatstone, 1843). His research demonstrated that the brain processes two-dimensional images from each eye into a single three-dimensional object, creating a sense of depth and immersion. Later, in the late 1980s, Jaron Lanier, a computer scientist and visual artist, popularised the term "virtual reality" when he started developing VR technology for commercial use (Zheng et al., 1998).

The significant advancement of VR can be primarily attributed to the work of computer scientist Ivan Sutherland. In 1965, he presented the concept of the "Ultimate Display," a vision of a virtual world viewed through a head-mounted display (HMD) that replicated reality so convincingly that the user could not tell it apart from the real world (Sutherland, 1965). Sutherland and his student Bob Sproull created the first VR HMD in 1968, known as "The Sword of Damocles" (Sutherland, 1968). Despite its primitive design and being too heavy for practical use, as seen in Fig. 3, it was a groundbreaking step toward interactive, head-tracked virtual environments.

Fig. 3. Sutherland wearing his HMD “The Sword of Damocles” in 1968 (Sutherland, 1968)

The 1990s saw VR technology gain momentum, with significant developments in the military and entertainment sectors. The military invested in VR for training simulations, most notably in flight simulators (Harris et al., 2023), while the entertainment industry explored VR in gaming, arcades and films. In the early 2000s, VR development hit a few stumbling blocks, mainly due to the technology limitations of the time. However, the introduction of the Oculus Rift in 2010 marked the beginning of a new era in VR. This device, developed by Palmer Luckey and subsequently acquired by Facebook (now Meta), reignited interest in VR with its advanced technology and affordability.

Since then, VR technology has rapidly evolved, with major tech companies like HTC, Sony, and Samsung entering the market. Modern VR headsets offer high-resolution displays, precise head tracking, and immersive audio, providing a highly engaging experience. The development of controllers and motion-tracking technology has further enhanced the interactivity of VR, allowing users to interact with virtual environments naturally.

Today's VR technology is not just confined to gaming and entertainment. It is used in various fields, such as education, healthcare, and real estate. VR offers an immersive learning experience in education, making complex concepts easier to understand (Kavanagh et al., 2017). In healthcare, VR is used for surgical training, patient treatment for phobias, and as a tool for rehabilitation (Pillai & Mathew, 2019).

The development of standalone VR headsets, such as the Oculus Quest, has made VR more accessible to the general public. These headsets do not require a connection to a PC or console, making them more convenient. The VR market is also growing in the enterprise sector, with businesses using VR for training, design, and collaboration.

The VR industry continues to innovate, exploring new ways to enhance the user experience. Advances in eye-tracking technology, haptic feedback, and the exploration of adding real smell

push the boundaries of immersion and realism in virtual reality experiences. The future of VR holds tremendous potential, with ongoing research and development paving the way for more groundbreaking applications. In summary, the history of VR is a journey of technological innovation and creative exploration. From its humble beginnings in panoramic paintings and stereoscopes to the sophisticated HMDs of today, VR has continually pushed the boundaries of how we experience and interact with digital worlds. As technology advances, VR's potential applications are significant, promising to transform various aspects of our lives and work (Bowman, 2007).

2.4. Astronomical Datasets and Formats

This work focuses on two popular dataset formats: NASA Flexible Image Transport System (FITS) and IPAC. We chose these two formats for different reasons; the FITS format is the most popular in astronomy for representing imagery data, and the IPAC format is a very simple tabular format yet robust. By combining both, we can develop and test our software easily using the IPAC format and further develop it to support FITS and, with it, allow a high adaptation rate for our software.

FITS Format

The Flexible Image Transport System (FITS) is an essential standard in astronomy, developed for efficiently storing, sharing, and analysing scientific data such as images, tables, and multi-dimensional arrays (Pence et al., 2010; Wells et al., 1981). Since its creation in the late 1970s, FITS has been celebrated for its adaptability and ability to facilitate data interchange across diverse computing environments, a necessity in the collaborative world of scientific research. Its structure, composed of self-describing 'header-data units' (HDUs), ensures each file carries comprehensive metadata alongside the data itself, allowing for a wide array of information types to be accurately and efficiently stored. This inherent flexibility and future-proofing have ensured FITS remains a vital tool in astronomy and beyond. However, FITS faces challenges, notably its complexity and the steep learning curve for those new to the format, potentially hindering broader interoperability with non-specialised software. Despite these hurdles, FITS's ability to evolve without obsoleting previous data—thanks to its extensible design—preserves its utility and relevance in scientific data management. The format's support for both ASCII and binary table extensions caters to the diverse needs of modern astronomy, making it indispensable for cataloguing celestial objects and their characteristics and ensuring its continued prominence in astronomical research and data analysis.

IPAC Format

The IPAC table format [2], developed by the Infrared Processing and Analysis Center at Caltech, is a simple file format tailored for the astronomical community, particularly beneficial for managing large catalogues and complex datasets in infrared astronomy. Its adoption in major astronomical

[2] https://irsa.ipac.caltech.edu/applications/DDGEN/Doc/ipac_tbl.html

surveys, such as WISE, highlights its significance alongside the more traditional FITS format (Wright, 2010). Characterised by its straightforward, tabular structure, the IPAC format excels in readability and simplicity, featuring space- or tab-separated columns and initial lines dedicated to descriptive metadata. This design facilitates human understanding and debugging and ensures the format is self-describing, making data easily interpretable by humans and software. Its ability to handle diverse data types—from numerical and string values to complex arrays—and to incorporate extensive annotations makes the IPAC format highly flexible and suited for the vast data volumes generated by contemporary astronomical research.

Despite its advantages, the IPAC format's simplicity can be a double-edged sword. While it aids in user accessibility and software manipulation, it may fall short in representing intricate data relationships or extensive metadata, a gap more sophisticated formats like FITS can fill. Additionally, its text-based nature, though beneficial for manual inspection, might lead to less efficient data storage and slower transfers compared to binary formats, posing challenges in the era of big data astronomy. Nevertheless, the IPAC format's practical utility is undeniable, having played a crucial role in the success of significant infrared sky surveys and research projects. Its design priorities ease of use, flexibility, and efficient data management, cementing its place as a valuable asset in the astronomical toolkit, especially as the field generates increasingly large datasets.

2.5. 2MASS Redshift Survey (2MRS) Dataset and the 2MRS Group Catalogue

As part of our work, we had to choose a dataset to test and develop on. We chose to use the 2MASS Redshift Survey (2MRS) dataset as this dataset is complex enough to be considered “Big Data” and interesting enough to be used in real-life applications. This extensive astronomical survey combines data from the Two Micron All-Sky Survey (2MASS) with redshift data from various sources (Huchra et al., 2011; Skrutskie et al., 2007). This survey has played a critical role in mapping the local universe (Jarrett, 2004). The primary goal of 2MRS is to create a comprehensive and detailed three-dimensional map of galaxies within 300 million light-years of the Earth. It is one of the most complete surveys, covering 91% of the sky and roughly 45,000 galaxies, as seen in Fig 4.

One of the significant achievements of the 2MRS has been the detailed mapping of the cosmic web in our local universe. This includes clusters, superclusters, and voids of galaxies, providing valuable insights into the universe's large-scale structure. The survey has also helped identify previously unknown structures and has been pivotal in dark matter research and the overall mass distribution of the nearby universe. The data from 2MRS continues to be a valuable resource for astronomers and astrophysicists, contributing to ongoing research and discoveries in the field.

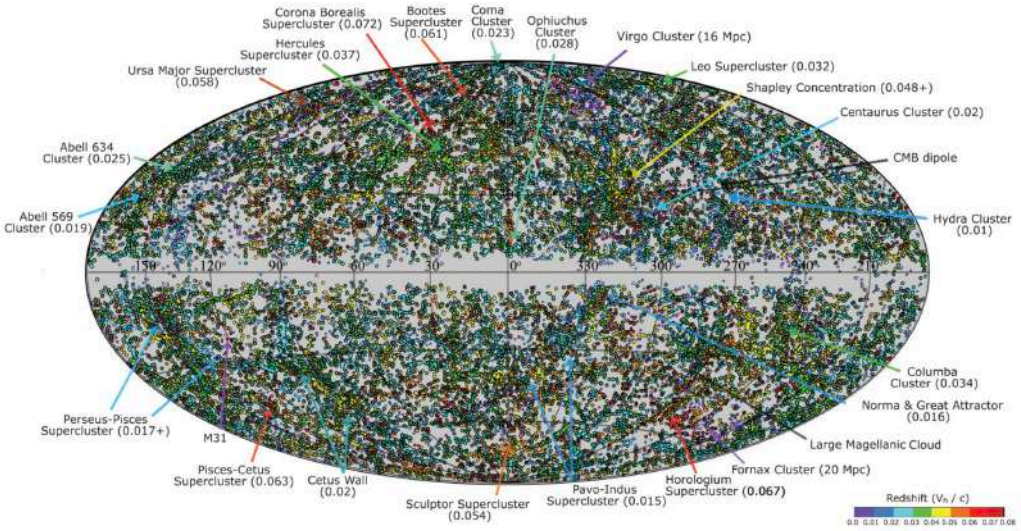


Fig. 4. The full 2MRS Dataset colour represents the redshift (Huchra et al., 2011)

An additional aggregated version of the 2MRS dataset, the 2MASS Redshift Survey (2MRS) Galaxy Group Catalogue (Lambert et al., 2020), represents a significant advancement in studying the universe's large-scale structure, leveraging a novel approach based on graph theory and a modified version of the Friends-of-Friends (FoF) algorithm (Feigelson, 2012). This catalogue, derived from the comprehensive data of the 2MASS Redshift Survey, includes 44,572 redshifts and introduces 1,041 new measurements for galaxies, with a significant focus on those within the Zone of Avoidance (ZoA). The ZoA is a region in space obscured by the dust and stars of the Milky Way, making observations challenging (Herschel, 1863). Despite this, a truly "whole-sky" redshift survey has been a long-standing goal in astronomy to fully understand phenomena such as the Cosmic Microwave Background (CMB) (Sunyaev, 1974).

2.6. Modern Visualisation Tools

Most researchers are interested in manipulating specific particles or a group of particles to understand the relevant information better. This manipulation, often called interaction, allows the researcher to select and rearrange the particles depending on specific properties. For example, a researcher might be interested in identifying all the particles with a specific redshift or mass, isolating (selecting them) from the full sample and seeing how their other properties relate to one another.

Current solutions allow particle data interaction and visualisation with a two-dimensional (2D) desktop application. One of the popular applications among researchers for such visualisation is CARTA (Wang K.-S, 2020), which plots particles on top of a 2D images. For interacting and rendering particles in 3D space, researchers use, e.g., Partiview (Levy, 2003) and TOPCAT (Taylor, 2017). Each of these programs has different advantages, disadvantages, and specific tasks at which they excel. Still, they all share the same layout: a desktop application running on a 2D device, such

as a monitor, including a toolbar for the various tools and a data viewer. The manipulation of data with these programs is achieved using the traditional methods of interaction: a keyboard and mouse. Selecting particles in a 3D space (such as the universe) on a 2D desktop application can be, in the best case, cumbersome and, in the worst case, inaccurate due to the lack of depth perception.

CARTA

CARTA (Cube Analysis and Rendering Tool for Astronomy) in its remote (client/server) version is a powerful visualisation tool specifically designed for large astronomical datasets, often reaching gigabytes to terabytes in size. It utilises a client-server architecture, where the server often is a powerful computer located over a local or a cloud network, making it suitable for handling large file sizes typically obtained from advanced observatories like ALMA, VLA, or SKA pathfinders (Wang K.-S, 2020). CARTA's architecture allows for efficient processing and visualisation, as remote servers handle heavy computations and data storage while clients receive processed products for visualisation. This architecture has a notable drawback: latency. If the server is connected to the internet and experiencing a low data transmission rate, there can be latency between the user's action and the actual execution by the server. This latency can cause frustration and poor user experience. CARTA can use the same computer as a client and as a server; while removing the latency drawback, this setup lacks the clear advantages mentioned in heavy processing.

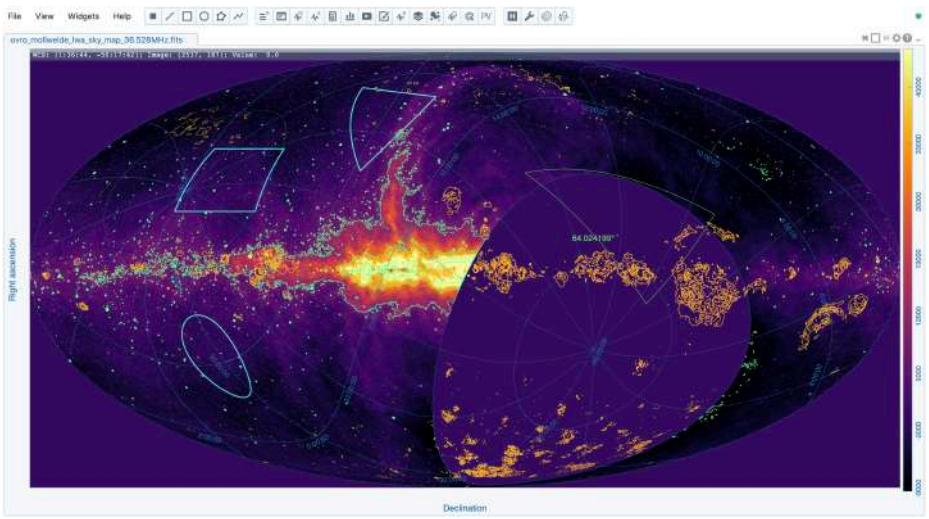


Fig. 5. A screenshot from the CARTA application [3]

CARTA supports various data formats, including FITS. The server version of CARTA is compatible with Linux, enabling its deployment across multiple research environments.

While primarily a data viewer, CARTA's memory-efficient design allows the rapid loading of large image cubes, making it highly efficient for visualising vast datasets. The tool leverages parallelisation and GPU-accelerated rendering for efficient data processing and display. CARTA

[3] https://cartavis.org/

includes spectral line cube analysis tools and supports efficient catalogue processing and rendering, handling catalogues with millions of sources. The limitations in CARTA's user interface, particularly in selection and direct data manipulation, primarily manifest as constraints in precise data selection, restricted abilities for in-depth, real-time data editing, and limited interactivity for complex data exploration. CARTA does not allow a view of the particles in 3D space but plots them on top of an image. These limitations can hinder users' capacity to perform advanced or nuanced analysis tasks directly within CARTA, potentially necessitating supplementary tools for data preparation or modification. Although selection and direct data manipulation are limited, CARTA's viewing capabilities are robust, making it a vital tool for researchers dealing with large-scale astronomical data. The latest release of CARTA includes features like interactive preview, enhanced image fitting capabilities, and support for angular distance measurement, among others. Future releases plan to introduce features like full workspace support, RGB image blender, and a Python scripting interface, indicating ongoing development and enhancement of the tool's capabilities.

Partiview

Partiview is an interactive graphical viewer and editor designed specifically for the visualisation and editing of particle data (Abbott et al., 2004; Levy, 2003). Its core functionality revolves around providing a 3D visualisation environment that allows users to not only observe but also manipulate particle datasets in three dimensions. This feature is instrumental in comprehending the spatial relationships and dynamics within large sets of particle data, offering a more intuitive understanding of the data's structure and behaviour.

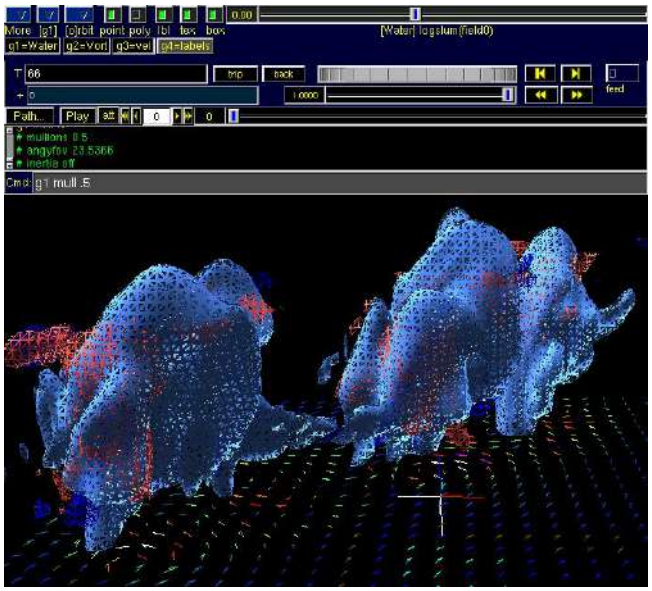


Fig. 6. A screenshot from Partiview showing a 3D rendering of particle data [4]

[4] https://virdir.ncsa.illinois.edu/partiview/

One of the significant advantages of using Partiview is its comprehensive suite of visualisation tools, which include options for rendering particles in various styles, such as points, spheres, or particle density, to best suit the data's nature and the analysis requirements. Users can dynamically adjust the visualisation parameters, such as colour coding, transparency, and lighting effects, to uncover hidden patterns or highlight specific aspects of the data. Furthermore, Partiview supports real-time interaction with the dataset, enabling users to rotate, zoom, and pan through the data, as well as to select and isolate subsets of particles for detailed examination. Partiview's ability to integrate with simulation software and data analysis tools further enhances its utility, allowing for seamless workflows from data generation through to analysis and interpretation. This integration facilitates the direct visualisation of simulation results, making it easier for researchers to validate models and hypotheses against observed data. Partiview stands out from all other 3D rendering tools, due to its unique 2D cursor that functions within the 3D environment, making it powerful for handling real datasets.

However, there are some disadvantages to consider. The complexity of 3D visualisation and the high level of detail in Partiview can demand significant computational resources, especially when dealing with extremely large datasets. This requirement may limit its accessibility or performance on lower-end hardware. Additionally, Partiview's support is discontinued by its authors, which can affect this tool’s adoption rate in the near future.

Despite these limitations, Partiview's advanced 3D visualisation capabilities make it an essential tool for anyone working with particle data. Its ability to provide deep insights into the structure and dynamics of particle systems far outweighs the challenges associated with its use, making it a valuable asset in the toolkit of researchers and professionals in relevant fields.

TOPCAT

TOPCAT, short for Tool for OPerations on Catalogues And Tables, is a highly regarded interactive graphical viewer and editor for tabular data (Taylor, 2017). It is used in the astronomical community for the examination and manipulation of data tables, including catalogues of astronomical objects. One of its key strengths lies in its powerful capabilities for 2D visualisation of particle data, making it an indispensable tool for astronomers and physicists who work with large datasets. The advantages of using TOPCAT for 2D visualisation are numerous. Firstly, it offers a wide range of plotting options, including scatter plots, histograms, density plots, and contour plots, which are crucial for the detailed analysis of large and complex datasets. Its simple interface allows for easy manipulation of these plots, including zooming, panning, and selection of subsets of data for closer inspection. Moreover, TOPCAT integrates seamlessly with various astronomical databases and services, enabling users to retrieve and visualize data from multiple sources quickly. Its flexibility in handling large datasets, combined with its robust data manipulation and filtering capabilities, makes it an excellent choice for researchers dealing with vast amounts of tabular data repressing particles.

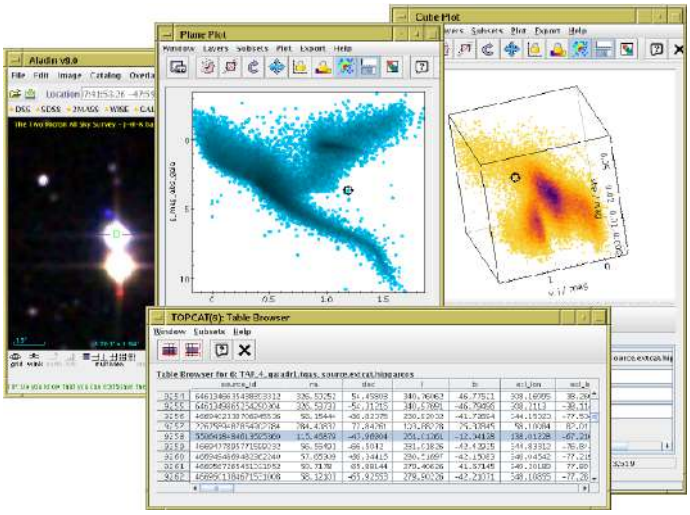


Fig. 7. A screenshot from TOPCAT showing a source catalogue with particle representation in 2D (Taylor, 2017)

However, TOPCAT's primary focus on 2D visualisation is also a limitation, particularly when it comes to the exploration of data that inherently exists in three dimensions. While it does provide some support for 3D plots, these capabilities are not as developed or as intuitive as its 2D visualisation tools. This limitation can be a significant drawback for research that requires the detailed spatial analysis of objects in three dimensions, such as in certain fields of astrophysics and cosmology where the 3D distribution and movement of objects are of interest. Users needing advanced 3D visualisation may need to supplement TOPCAT with other software specifically designed for 3D data analysis, which can introduce additional complexity into their workflow.

The differences between the different features of all the software mentioned in this chapter are shown in Tbl. 1.

Feature	CARTA	Partiview	TOPCAT
Primary Focus	Visualisation and analysis of large images	3D visualization and editing of particle data	2D visualisation of tabular data
Data Handling	Optimized for large-scale astronomical data	Capable of managing large datasets with efficiency	Excellent for large datasets, integrates with astronomical databases
Visualisation	Advanced, with robust viewing capabilities	Advanced 3D visualization tools, including various rendering styles	Advanced 2D plots, limited 3D visualisation capabilities
Customisation	Highly customizable UI	High degree of customisation available for visualising particle data	High degree of plot customization, including markers, axes, and scaling
User Interface	Modern, intuitive interface	Old GUI, not as intuitive for novice users	Intuitive GUI for easy navigation and manipulation of data
Data Manipulation	Limited direct manipulation	Supports interactive manipulation and selection of the dataset using 3D cursor	Comprehensive tools for data filtering, selection, and operation
Interactivity	High, especially in data viewing	Real-time interaction capabilities allow users to rotate, zoom, and pan through datasets	Interactive plots with zooming and panning, clickable points for data interrogation
Community	Growing, with active development. Widely used in astronomy	Specialised, focusing on fields like physics, offering targeted support and resources, but may not be as large or broad-based as communities for more general-purpose tool	Strong community support, widely used in astronomy and astrophysics and other fields with particle data
Ease of Use	Simple for its target audience	Complexity of features may present learning-curve, affecting ease of use for beginners	Generally simple, though may have a learning curve for advanced features
Input Data Format	FITS, CASA, HDF5	TIPSY Binary, ASCII, HDF5	IPAC, ASCII, CSV, HDF5, FITS, SQL, CDF

Tbl. 1. A feature comparison table between CARTA, Partiview and TOPCAT

2.7. Virtual Reality and Astronomy

Integrating VR in astronomy has revolutionised how researchers visualise and interact with astronomical data. VR technology, including HMDs and spherical dome displays, have opened up new space science and exploration dimensions (Djorgovski, 2013; Ferrand, 2016; Fluke, 2018; Marchetti, 2018).

One of the pioneering applications of VR in astronomy is the detailed visualisation of complex astronomical phenomena. For example, scientists at Harvard University utilised VR to create a 3D visualisation of a supernova remnant, revealing its spherical and disk-like components in detail (Arcand et al., 2020). This visualisation was previously unachievable with traditional methods, showcasing VR's potential to enhance our understanding of celestial events 5.

NASA has also been at the forefront of employing VR for scientific discovery. By animating the speed and direction of millions of stars in the Milky Way, researchers could classify star groupings in a way that was impossible with conventional six-dimensional paper graphs. This approach provided new insights into the stars' motions and origins, demonstrating VR's ability to offer novel perspectives on astronomical data (NASA, 2020). NASA even published all the VR code as open-source via GitHub 6. Furthermore, VR and Augmented Reality (AR) are being explored for their applications in mission planning, operations, and data visualisation of high-dimensional abstract data spaces. These technologies provide a natural, intuitive, and highly effective means for collaborative data visualisation and exploration, significantly benefiting the fields of astrophysics and space sciences. It is even being used on the International Space Station 7.

NASA's Glenn Research Center employs VR headsets 8 with hand controllers that allow users to view 3D models and feel completely immersed in virtual environments. This technology is used in multiple applications, including manipulating objects onboard the International Space Station and educational outreach events, further illustrating VR's versatility in space exploration and public engagement (Delgado & Noyes, 2017). VR in astronomy is not just about visualising existing data; it also opens the door to new ways of interacting with and interrogating data. For instance, combining multi-wavelength images and volumetric cubes in VR environments allows astronomers to explore the past and present growth of cosmic systems more intuitively and interactively (Jarrett et al., 2021; Marchetti, 2020). In summary, VR technology in astronomy advances research possibilities and transforms how astronomical data is visualised and understood.

[5] https://chandra.cfa.harvard.edu/vr/

[6] https://github.com/nasa/PointCloudsVR

[7] https://www.nasa.gov/missions/station/nine-ways-we-use-ar-and-vr-on-the-international-space-station/

[8] https://www1.grc.nasa.gov/facilities/gvis/devices/

2.8. Immersive Data Visualisation Interactive Explorer (iDaVIE)

The Immersive Data Visualisation Interactive Explorer (iDaVIE), created by the IDIA Visualisation Laboratory at the University of Cape Town, marks another development in astronomy data visualisation. Designed to handle Big Data adeptly, iDaVIE's initial design utilised the Unreal Engine, a choice that underscored its focus on robust data handling and visual fidelity. However, the team later transitioned to Unity for enhanced flexibility, especially in creating custom shaders using High-Level Shader Language (HLSL). This transition highlights iDaVIE's commitment to maintaining adaptability in the ever-evolving field of scientific data visualisation (T.H. Jarrett, 2021). iDaVIE distinguishes itself with two primary modes: iDaVIE-p and iDaVIE-v. The former, iDaVIE-p, is dedicated to rendering particles, facilitating the visualisation of discrete data points in a spatial context. iDaVIE-v, on the other hand, caters for volumetric rendering, dealing with cube-shaped data sets. Before this dissertation, iDaVIE-p was not as developed as iDaVIE-v and offered only visualisation without selection and manipulation of the data. As will be discussed in detail in the next chapters, we show how we designed, developed and evaluated iDaVIE-p to allow the selection of particles and the manipulation of the data, often called parametrisation. These improvements are important for both the sake of this research and the general functionality of iDaVIE-p as an end-user product. While available only on Windows via SteamVR, plans to extend support for iDaVIE to Linux reflect an acknowledgement of the varied operating systems prevalent in the astronomical community. However, the lack of SteamVR macOS support may present a barrier for researchers accustomed to Apple's ecosystem. Beyond its rendering capabilities, iDaVIE stands out for its ability to process a range of data formats. This includes text-based tables in IPAC-table format, binary FITS tables and XML files. Users can customise their visualisation experience through a JSON file, which allows for mapping spatial coordinates and fine-tuning rendering parameters such as colour map, opacity, particle size, and shape. This level of customisation ensures that iDaVIE can be effectively tailored to meet specific research needs and preferences.

The interactive aspect of iDaVIE is another significant feature. It offers an immersive experience, where users can navigate the virtual environment by physically moving around, through intuitive navigation gestures using controllers. In iDaVIE-v, users can also select and interact with the volumetric data. This interaction style not only enhances user engagement but also allows prolonged and productive visualisation sessions, thereby augmenting the overall utility of the tool. As previously mentioned, iDaVIE-p did not allow interaction and selection before this dissertation.

In summary, iDaVIE represents a pioneering innovation in the visualisation of astronomical data. It leverages the power of virtual reality technology to provide an immersive and interactive platform for the complex exploration of big data sets. The development and capabilities of iDaVIE illustrate a forward-thinking approach to data analysis and presentation in astronomy, signalling a new era in how researchers interact with and interpret vast amounts of astronomical data.

As this thesis tackles the selection of catalogue data, we focus solely on iDaVIE-p, specifically catalogue rendering, selection, interaction and parameterisation of particles, and set aside the volumetric mode, iDaVIE-v, as it is irrelevant to our context.

iDaVIE-p (Catalogue Rendering)

iDaVIE-p, a mode within iDaVIE, is tailored explicitly for particle rendering in the context of astronomical data visualisation (see Fig. 8). This mode allows users to interact with and navigate through large catalogues of astronomical data in a virtual reality (VR) environment. Users can zoom in on specific areas, measure distances between particles, and change their viewpoints to explore different aspects of the data. Such capabilities are vital in astronomy, where understanding the spatial distribution and relationships between celestial objects is crucial (T.H. Jarrett, 2021).

However, iDaVIE-p faces performance and rendering challenges when dealing with extremely large datasets or when the particles in the dataset are densely packed. For instance, visualisation issues arise when more than a certain number of particles are visible simultaneously. This limitation poses a significant gap in the tool's implementation, as astronomical datasets often contain a vast number of particles, and an insufficient particle count can lead to a loss of detail and accuracy in the representation of raw data. In addition, iDaVIE-p completely lacks interaction methods functionality. The user cannot select particles and manipulate them. Without this capability, iDaVIE-p cannot be considered production-ready, as it serves only as a viewer of data and thus does not utilise the full potential of VR interaction.

Furthermore, while iDaVIE-p allows tuning parameters in the JSON file, this must be done on the desktop. This means that adjustments to parameters, such as the size of the particles, require users to temporarily exit the immersive environment by removing the HMD, change settings, and then re-enter, which can disrupt the immersive experience and the flow of analysis. Despite these limitations, iDaVIE-p proves effective when the data is pre-processed or clustered into groups. For example, when working with data from the 2MRS Group Catalogue, iDaVIE-p can provide meaningful and insightful visualisation. While iDaVIE-p offers a unique and immersive way to explore astronomical data, its limitations in handling extremely large or dense datasets and the lack of on-the-fly parameter adjustment highlights areas for potential improvement, which is the goal of this thesis.

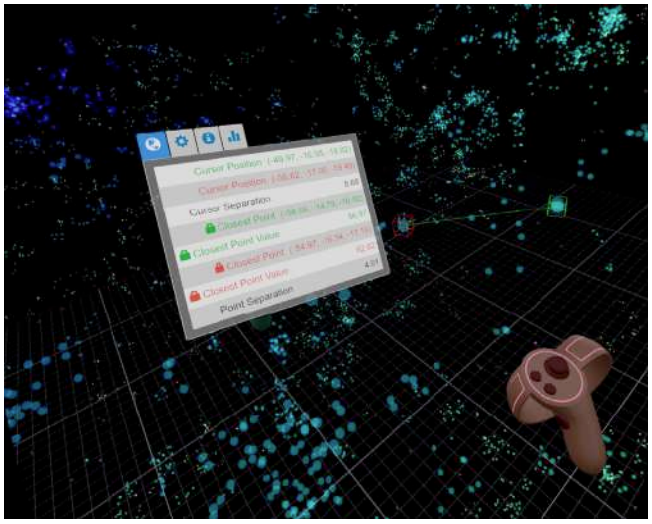


Fig. 8. A screenshot from the first version of iDaVIE-p, before the development introduced in this thesis

Chapter 3 - Related Work

This chapter examines literature and previous work related to the core features of selection and parameterisation. It is divided into four sections: selection and interaction, depth cues, selection in VR and parameterisation in VR. The selection and interaction section examines previous work about selection, which is an essential part of Human-Computer Interaction or HCI. The depth perception and cues section dives deeper into techniques that allow us to improve selection using 3D depth, specifically depth cues that can benefit particle clouds. We compare the depth cues and highlight cues that might be relevant for us in the development process. Then, the selection in VR chapter takes focuses on selection methods and techniques in VR, such as ray casting, go-go selection and more. Lastly, we examine the existing literature about parameterisation, specifically in VR, and the different ways of typing a keyboard on VR, from ray casting to eye-gaze solutions.

3.1. Selection and Interaction

Selection in computer science stems directly from interaction with the computer. Human-computer interaction, or in short, HCI, is the research field in design and engineering that focuses on the interaction between the human (user) and the computer. Selection in HCI is a fundamental aspect crucial in the interaction between users and computer systems. This process involves the user choosing from multiple options a computer system presents, from selecting a piece of text, an icon, a menu item, or any other element within a graphical user interface (GUI). The efficiency, accuracy, and user satisfaction of the selection process are critical for the usability of computer systems, making it a primary focus of HCI research. Selection, often referred to as manipulation in literature (Sellen et al., 1992), will be used throughout this thesis as a synonym to manipulation.

According to research in HCI (Sellen et al., 1992), the design of selection mechanisms should consider several key factors, including a users' physical capabilities, the context of use, and the technological constraints and affordances. For instance, the size and spacing of targets can improve the speed and accuracy of selection tasks, based on Fitts' Law, which models the time required to rapidly move to a target area as a function of the distance to and size of the target.

Moreover, research has been conducted on text entry systems, an essential subset of selection tasks, showing how different input methods can significantly affect text selection and entry speed and error rates (MacKenzie et al., 1991). This research underscores the necessity of designing input systems that align with human motor capabilities and limitations. In recent developments, HCI researchers have explored advanced selection techniques that leverage gestural interfaces, eye tracking, and brain-computer interfaces to create more natural, efficient, and accessible ways for users to interact with technology (Mridha et al., 2021; Poole & Ball, 2006). For example, eye gaze as a pointer for selection tasks, was found to make selection fast and intuitive with appropriate design adjustments, opening new avenues for interaction beyond traditional pointing devices (Zhai et al., 1999).

Collectively, these studies emphasise the importance of understanding human capabilities and behaviour in designing selection mechanisms that are both effective and satisfying for users. In this thesis, we focus on traditional, yet popular, 2D inputs such as mouse and keyboard for selection as part of the desktop version of iDaVIE-p. For the VR version, we explore later at this chapter the possible selection methods.

3.2. Depth Perception and Cues

Depth perception is a crucial aspect of human vision, enabling individuals to perceive the world in three dimensions and understand the relative position and size of objects in their environment. It arises from a complex interplay of various sensory cues, which the brain integrates to form a coherent three-dimensional understanding from two-dimensional retinal images. These cues are categorised into two main types: binocular cues, which require input from both eyes and monocular cues, which can be perceived with just one eye.

Binocular cues include stereopsis, convergence, and accommodation. Stereopsis, or retinal disparity, refers to the slight difference in the images projected on each eye due to their horizontal separation. The brain uses these disparities to gauge the distance of objects, with closer objects having a larger disparity. Convergence involves the inward movement of the eyeballs as they focus on nearer objects, providing the brain with distance information based on the degree of convergence. Evidence suggests the brain is not good at incorporating this information unless it is within arm's length (Viguier et al., 2001), but it is very quick at recalibrating when presented with other spatial information (Fisher, 1980). Accommodation is changing the lens shape inside the eye to focus on objects at different distances at a rate of 200 milliseconds (Ware, 2013). According to some theories, the choice of what to focus on is heuristic (Wolfe, 1996).

Monocular cues, on the other hand, encompass a range of indicators that can be observed with one eye and include motion parallax, texture gradient, interposition, relative size, relative height, linear perspective, and light and shadow (Ware, 2013). Motion parallax occurs as one moves through the environment; closer objects move faster across the field of view than distant ones. Texture gradients reveal depth by the way textures become denser with distance. Interposition, or overlap, suggests that an object covering part of another is closer. Relative size indicates that if two objects are known to be similar, the smaller one is further away. Relative height in the visual field suggests objects lower in the field of view are closer than those higher up because it is assumed that most objects ar positioned on the ground. Linear perspective refers to the way parallel lines appear to converge with distance. Finally, light and shadow can provide cues about objects' three-dimensional shape and distance through the patterns of light and shading. Monocular cues have an important subset of cues - perspective cues. Perspective cues are important in visual perception, particularly in conveying depth and spatial relationships within a two-dimensional representation of a three-dimensional scene, such as in virtual environments. These cues rely on the geometric principles of perspective to suggest depth. The most common forms include linear perspective, where parallel lines appear to converge in the distance at a vanishing point, and atmospheric perspective, which uses changes in

colour and clarity to indicate distance, with distant objects appearing bluer and less distinct than those closer to the viewer. This is often also called proximity luminance contrast covariance (Dosher et al., 1986). Another aspect, the size perspective, suggests that objects of the same known size will appear smaller as they are further away, reinforcing the perception of depth through relative size. These cues are fundamental in computer graphics, enabling designers to create the illusion of three-dimensionality on a flat surface. By manipulating perspective cues, designers can guide the viewer's eye through a scene, enhancing realism and providing a sense of immersion, even without binocular vision.

Combining depth cues together requires attention to their effect on the task. Research suggests that depth cues are not weighted the same for different tasks (Ware, 2013). For example, shadows can benefit one task but cause confusion and distortion in another. Some depth cues also depend on each other, causing a dependence chain, as shown in Fig. 9.

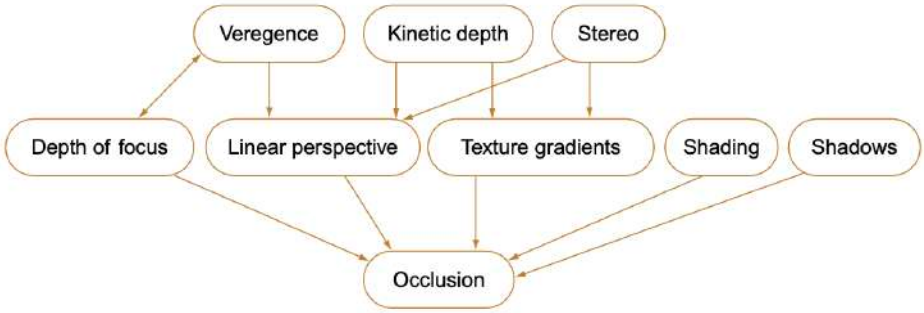


Fig. 9. A dependency graph for depth cues. Arrows indicate how depth cues depend on each other (Ware, 2013)

In computer software, particularly in graphics and virtual reality (VR) applications, replicating the depth cues found in natural vision could prove essential essential for creating immersive and realistic three-dimensional environments. These software solutions employ various techniques to simulate binocular and monocular depth cues, enhancing the user's perception of depth and spatial relationships within a virtual scene. To mimic the binocular cues of stereopsis, convergence, and accommodation, VR systems and 3D graphics software often use stereoscopic rendering. This involves generating two slightly different images for each eye, simulating the retinal disparity found in natural vision. Specialised hardware, such as VR headsets, can present these images in a way that the brain interprets as three-dimensional space. Some advanced systems also track the user's eye movements and adjust the images accordingly to simulate convergence and accommodation, further enhancing the sense of depth. Computer graphics algorithms also incorporate monocular cues to suggest depth. Techniques such as texture mapping and gradient shading replicate the texture gradient and light and shadow cues, while algorithms for rendering perspectives and occlusion mimic linear perspective and interposition (Ware, 2013). Motion parallax is achieved in interactive environments by altering the scene's perspective in response to the user's movement, making closer objects appear to move faster relative to the background; this is often called a kinetic cue.

Additionally, the relative size and height of objects are carefully designed in virtual environments to align with expectations based on natural vision, aiding in the perception of depth even on a flat screen (Ware, 2013).

However, depth cues for point clouds or particles work differently. Most perspective cues do not help with perceiving depth in a 3D point cloud. Particles, being small and discrete points, have no perspective information that can be projected using depth cues. In this case, stereoscopic depth and motion cues become more important (Ware, 2013). To visualise the point cloud and reveal its boundaries, one can enhance the perception by using shadow cues on the whole point cloud, as shown in previous research (see Fig. 10) (Li, 1997).

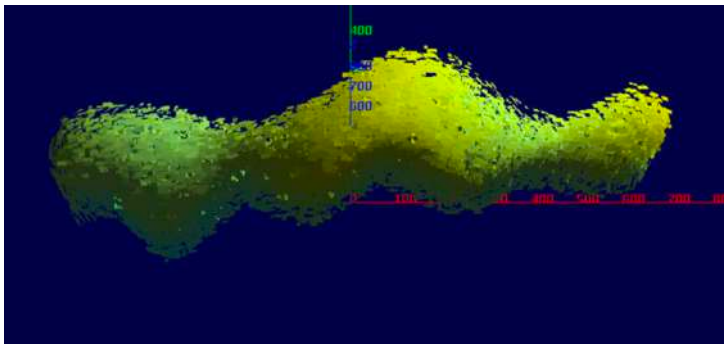


Fig. 10. Shading the point cloud gives it boundaries and depth (Li, 1997)

As per research, artificial spatial cues have also been effective (Gibson, 1986). These depth cues add artificial objects to the scene, such as lines or a ground plane, to allow the user to perceive the depth of the individual objects, as shown in Fig. 11.

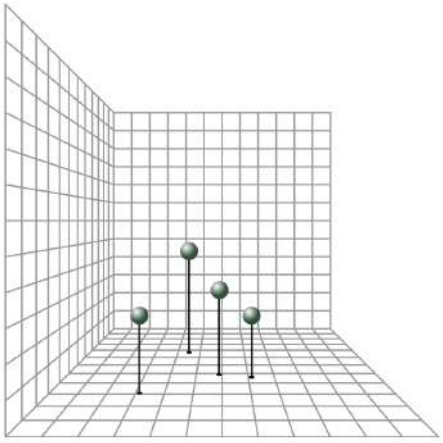


Fig. 11. Lines to the ground can add depth to particles (Ware, 2013)

We compare the different depth cues in the following table (Tbl. 2). Notice that not all depth cues are suitable for the software we developed, as our data is particle data. The depth cues that are relevant and that will be explored in designing and developing iDaVIE-p are highlighted.

Depth Cue	Type	Description
Binocular Disparity	Binocular	The difference in the images received by each eye due to their horizontal separation, allowing for depth perception through stereopsis.
Convergence	Binocular	The inward turning of the eyes when focusing on a close object, providing a cue to depth based on the degree of convergence.
Motion Parallax	Monocular	Objects closer to the observer move faster across the field of vision than objects further away when the observer moves, providing a cue to relative distance.
Linear Perspective	Monocular	Parallel lines appear to converge in the distance at a vanishing point, providing a cue to depth.
Texture Gradient	Monocular	The density of texture increases with distance, providing a cue for depth as closer objects have more detailed texture.
Interposition	Monocular	When one object overlaps another, the overlapping object is perceived as closer.
Relative Size	Monocular	If two objects are known to be of similar size, the one that appears smaller is perceived to be further away.
Height in the Visual Field	Monocular	Objects closer to the horizon in the visual field are perceived to be further away than those located below or above the horizon.
Light and Shadow	Monocular	Shading and shadow patterns can provide cues about the distances and contours of objects, making them appear more three-dimensional.
Kinetic Depth	Monocular	The phenomenon where the movement of an object provides cues about its three-dimensional shape. Rotating objects can reveal their structure.
Artificial Cues	Both	Cues created by humans to enhance depth perception in images, such as 3D glasses, VR environments, or techniques used in paintings and drawings.

Tbl. 2. Comparison between the depth cues. Cues highlighted in dark grey will be explored later in this thesis as part of the design and development process of iDaVIE-p

3.3. Selection in VR

Various techniques have been developed to facilitate selection in VR environments, each leveraging the spatial capabilities and immersive qualities of VR technology.

The most simple yet most likely intuitive method to select objects in VR is using a virtual hand. Often called “grab selection”, this method allows the user to use the controllers (and their virtual representation in the virtual environment as hands or controllers) as if they were their hands. This means that everything within reach of the user is selectable, like in real life. The issue with this method is that, unlike in real life, the user usually does not have room to walk towards the object in the virtual environment, as the user is constrained by the physical space, which is most likely a room. To reach these distant objects, the user must use a locomotion system within the virtual environment to move their digital avatar near the object, and only then can they grab or select them. Using locomotion whenever the user wants to select might cause frustration, fatigue and inefficiency.

To overcome this physical space limitation, researchers developed a method that can select far in the distance, namely the ray casting (Roth, 1982), where users point at objects with a virtual ray or laser beam emanating from a handheld controller or finger-tracking device. This technique allows for precise selections at a distance but can become physically taxing over time and may require practice to achieve accuracy. The process involves mathematically checking for intersections between the ray and the geometric shapes of objects in the environment, which can be computationally intensive depending on the complexity of the objects and the number of checks required. Optimisations and spatial partitioning algorithms, such as octrees (Meagher, 1980), are often used to make these calculations more efficient. Ray casting is particularly useful for selecting or interacting with objects that are out of reach, providing a way to bridge the physical gap between the user and virtual objects. A notable development in VR ray casting was the RayCursor (Baloup et al., 2019). This solution moves between targets automatically, taking into consideration occlusion and depth. It also allows the user to control the length to which the ray (beam) extends and adds some fidelity so the selection can happen when the ray is near a target. An example of RayCursor and ray casting is shown in Fig. 12. The drawback is that the user might experience fatigue. Ray casting on distant and small objects, such as particles, requires holding the controllers very precisely, sometimes for longer. In addition, the precision of the ray casting depends greatly on the controller itself and its accuracy.

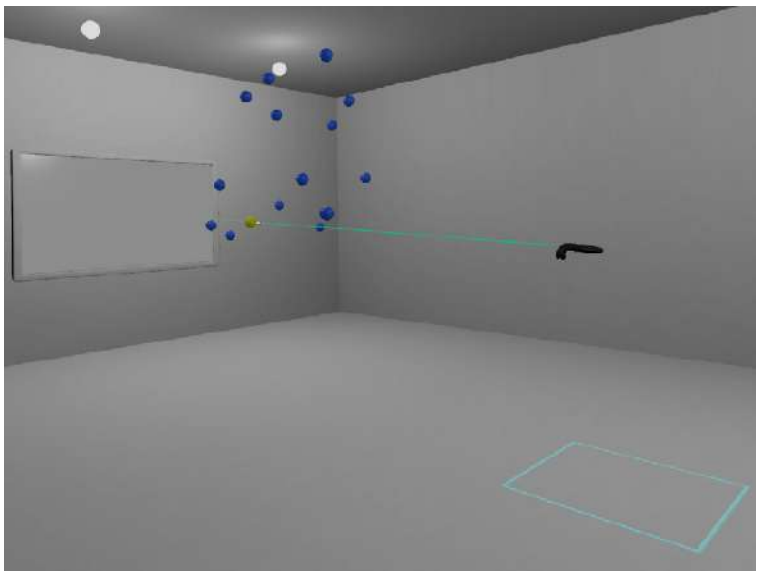


Fig. 12. RayCursor casting a ray at an object. The selected object is yellow (Baloup et al., 2019)

A method called cone ray casting, or cone selection, has been developed to mitigate the issue of selecting small items using ray casting and its precision (Steinicke et al., 2006). This method expands the ray into a cone shape to allow a threshold in the selection and the selection of smaller items in the distance. Sometimes, as shown by Steinicke et al., the selection cone can also be referred to as “sticky selection”, as the ray “sticks” to the closest element by bending towards it. While this method improves precision in the distance, it might harm precision when selecting items next to the user, as the cone might be too big in this case. In addition, ray casting may feel unnatural in VR, as it eventually mimics the mouse function from the 2D desktop experience.

The go-go selection was invented to overcome these issues and take the best of both worlds: ray casting and grab selection (Poupyrev et al., 1996). This selection method allows the user to use grab selection for objects near the user. Still, for objects in the distance, it allows the user to telescope the controller into the distance to grab the object. This method is theoretically perfect but takes a long time to master and might feel unnatural for most users due to the separation of the physical and virtual hands.

Another approach is head-gaze-based selection, which uses the direction of the user's gaze, tracked via a VR headset, to highlight and select items. While gaze-based methods are hands-free and can reduce physical effort, they might lead to accidental selections and require additional confirmation steps, such as blinking or button pressing, to finalise a choice. The selection of unwanted elements by looking at them is called the Midas Touch problem (Jacob, 1990).

Recent developments in VR technology also allow researchers to explore eye-gaze-based selection, using sensitive sensors that detect eye movement, saving the user the effort of moving their head and allowing a more accurate selection process (Blattgerste et al., 2018). Some research has proven that eye-gaze-based selection can be accurate and provide a better user experience by implementing three selection methods: Duo-Reticles (DR), Radial Pursuit (RP) and Nod and Roll (NR) (Piumsomboon et al., 2017). DR allows the user to gaze at an object, and when the object and the

eye's direction are aligned, and after some time passes (as a threshold), the object is selected. RP allows the user to select within a cluster of objects by looking at them. Once a radial threshold has been met, the objects within this cluster separate (in the VR environment), and then the user can gaze at a specific object. NR, sometimes called a Gaze-Dwell, allows the user to gaze at an object similar to DR, but the user has to approve the selection by nodding. Each method has its own advantages and disadvantages. The different methods for eye-gaze selection are shown in Fig. 13.

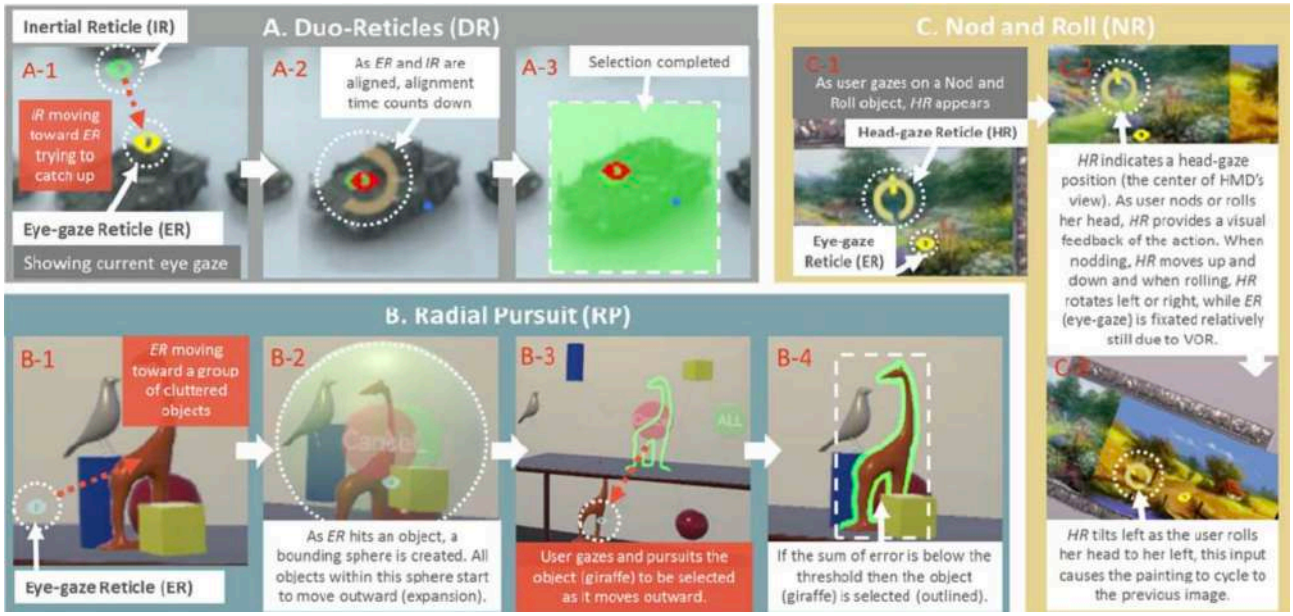


Fig. 13. DR, RP and NR eye-gaze-based selection methods (Piumsomboon et al., 2017)

This thesis focuses on standard grab (simple virtual hand). This method allows easy and fast implementation alongside an intuitive feeling for the user as it mimics real-life experience. The range limitation poses a challenge (which other selection methods do not have, such as go-go or ray casting). Still, that limitation can be mitigated due to iDaVIE's efficient locomotion system; unlike VR games where the space structure matters (walls, doors, etc.), in iDaVIE the user navigates empty space, practically moving the dataset around. This allow fast and efficient movement towards point of interest in the dataset. With an effective locomotion system, the standard grab selection suffers from little to no disadvantages. In addition, head-gaze-based selection was ruled out as small objects such as particles raise accuracy issues that we try to avoid. The hardware we used, supported by iDaVIE, does not support eye-gaze-based selection due to the lack of the relevant sensors. All selection methods are compared in Tbl. 3, based on their use cases, precision, disadvantages and ergonomics.

Selection Method	Input Method	Interaction Range	Precision	Ergonomics	Typical Use Cases	Disadvantages
Simple Virtual Hand (Grab)	Hand tracking or controllers	Limited by arm's reach	High, depends on tracking quality	Natural for close interactions	Object manipulation, UI interaction in close proximity	Limited reach, not suitable for distant interactions
Raycasting	Controllers or hand tracking	Long distance	Medium to high, depends on implementation	Can be tiring for prolonged use	Selecting distant objects, UI interaction	Can be imprecise for small targets, physically tiring over time
Cone Raycasting	Controllers or hand tracking	Long distance	Lower than raycasting, area-based selection	Similar to raycasting, potentially less precise	Selecting objects in a general area, useful when precision is less critical	Less precise in short distances, can result in accidental selections
Go-Go	Hand tracking or controllers	Extended beyond arm's reach using non-linear mapping	High within arm's reach, decreases with distance	Reduces physical strain for distant objects	Manipulating objects at varying distances without moving	Can feel unnatural, difficult to control at extended distances
Head-Gaze Selection	Head movement	As far as the user can turn their head	Low, broad selection area	Can be tiring for neck with prolonged use	Selecting large or distant objects when hands are occupied	Imprecise, can cause neck strain, slower selection process
Eye-Gaze Selection	Eye tracking	Limited by field of view	High, very precise	Low physical effort, but requires good calibration	Precise selection, accessibility features, analysis of user focus	Requires high-quality eye tracking, can be affected by calibration issues, accidental selections

Tbl. 3. Comparison between selection methods for VR

3.4. Parameterisation and typing in VR

Changing textual settings and typing in VR creates a challenge that does not exist on the desktop; The user must select the relevant setting or letter, namely a parameter, and click it. One solution for typing on VR is using a virtual keyboard. Using a keyboard in VR can be challenging, as selecting a letter to type is inherently a selection task. A research study conducted in 2018 looked into the options available for selecting letters over a virtual keyboard. It is divided into six methods: head pointing, controller pointing, controller tapping, free hand, discrete cursor, and continuous cursor (Speicher et al., 2018). The different methods are shown in Fig. 14.

Fig. 14. All six methods to select letters on a virtual keyboard in VR (Speicher et al., 2018)

Head Pointing allows the user to select letters using head-gaze-based selection and ray casting; in this case, the cursor is the ray itself. Controller Pointing allows users to select letters using ray casting from the controller as a source, similar to RayCursor, where the cursor is the ray itself. Controller Tapping allows the user to use the controller to tap the virtual keyboard to select specific letters. Free Hand allows the user to tap on the virtual keyboard as if it were a real keyboard. This method requires sufficient sensory data to track the user's hands. The Discrete Cursor uses the joystick on the controller to move between the letters. Clicking the joystick selects the relevant letter. The cursor in this method is a virtual highlight over the selected letter, as seen in Fig. 14. In this method, there is no connection between the controller's position in space and the selection on the keyboard; this is analogue to using the arrows on a real keyboard to select a folder instead of using the mouse. A Continuous Cursor is similar to the Discrete Cursor, with the only difference being that if the joystick is held in a relevant direction, the virtual cursor keeps moving in that direction. In contrast, the discrete method allows one cursor movement per one joystick movement, even if the joystick is held, hence its name: discrete. Speicher et al. found that Controller Pointing is the most accurate out of the six methods, while Discrete Cursor ranked as the lowest in terms of physical demand on the users that participated in the testing.

Even with these results, we chose to implement and use the Discrete Cursor due to the specific use case represented by this thesis. As iDaVIE-p deals with decimal numbers and fine-tuning that requires time, accuracy, and effort, we needed to find a method to balance these. While the Controller Pointing might be more accurate, editing long decimal numbers with it can cause extra physical demand, which is already prevalent in the main selection task: selecting the particles themselves. Therefore, minimising the physical demand for the secondary task, parameter tuning, was our priority.

Chapter 4 - Design and Implementation

This chapter offers a thorough description of the design specifications for the visualisation applications that have been developed, as well as the experimental designs employed to address the research questions presented in Chapter 1.2.

The main focus of this chapter is to design, choose and eventually develop the selection method that allows us to select particles in the iDaVIE-p VR version. We compare two 3D selection methods: a boolean operation sphere and a spatial deformation cylinder. The boolean operation sphere uses boolean operations to determine if a particle is contained within the selection sphere. As the name suggests, the spatial deformation cylinder uses a cylinder to select the particles. This cylinder's shape can be modified using spatial deformation, which this chapter discusses in detail.

Then, based on an analytical comparison based on our design goals, we compare the two methods and choose the boolean operation sphere. This chapter then details the development of this method. In addition, as we must have a desktop version of iDaVIE-p to answer the research questions, we discuss the design and development of a standard desktop selection method using the mouse pointer to select particles in this chapter.

4.1. Features Descriptions

In this section, we elaborate on the two main features that would be developed as part of the iDaVIE-p software: selection and parameterisation.

Selection (Manipulation)

In astronomy, selecting a subset of data for analysis is a critical and often complex task, given the vast and diverse nature of astronomical data. Astronomers use data from various sources, including telescopes capturing light across different wavelengths, space probes, and satellite-based observatories. The selection process in this context is essential for focusing on specific phenomena, testing hypotheses, and effectively managing the sheer volume of data.

One primary criterion for data selection in astronomy is the research objective or specific astronomical phenomena under investigation. For instance, an astronomer studying exoplanets might select data specifically from instruments capable of detecting planetary transits or radial velocity changes in stars.

Another important factor in data selection is the wavelength of the observed light or other electromagnetic radiation. Different astronomical phenomena are best observed at specific wavelengths. For example, an astronomer might select infrared data to study the formation of stars, which can penetrate dust clouds where stars form. Conversely, X-ray data would be selected to study high-energy phenomena like black holes or neutron stars.

Temporal selection is also crucial in astronomy. Certain phenomena require long-term observation data, such as monitoring variable stars or detecting subtle changes in a celestial object over time.

Other studies might focus on transient events, like supernovae or gamma-ray bursts, requiring data selection and analysis following these events in a short time window.

Moreover, the sheer volume of astronomical data often requires astronomers to employ sophisticated algorithms and data processing techniques to select and analyse data efficiently. Techniques such as machine learning and automated data pipelines are increasingly used to handle, filter, and process data, allowing astronomers to focus on the most relevant data for their research.

Finally, data selection in astronomy must consider data quality and calibration. Astronomical data can be affected by various sources of noise and bias, such as atmospheric interference for ground-based observations or instrument sensitivity. Selecting data often involves preprocessing steps to calibrate and normalise data, ensuring it is suitable for scientific analysis.

In summary, the selection of data subsets in astronomy is a multifaceted process guided by the specific goals of the research, the characteristics of the phenomena being studied, and the capabilities of observational instruments. It involves carefully considering the type of data, the wavelength observed, temporal factors, and data quality, all crucial for deriving meaningful and accurate insights from the data. Selecting individual sources, such as galaxies or stars, can help astronomers learn about their cosmic surroundings, past and future.

Parameterisation

Parameterisation, in a broad mathematical and computational context, refers to defining or expressing a system, model, or function in terms of parameters. Parameters are variables essential to the system's description, influencing its behaviour or outputs. This concept is widely used in mathematics, physics, computer science, and engineering to simplify complex systems, make them more understandable, or adapt them to specific situations.

Unlike command selection (Bailly & Malacria, 2020), which defines how a user interacts with an interface such as a menu or a button, parameterisation focuses solely on the change of a variable and its effects on the dataset as a result.

In computer science and engineering, parameterisation is often used to tailor algorithms or models to specific scenarios. In astronomy, and our specific case, parameterisation allows astronomers to change select sources such as galaxies and stars, based on parameters, to see how they behave under different conditions. For example, an astronomer might be interested in selecting the star based on its temperature, and therefore needs to isolate it with a different colour than colder stars. In addition, some parameters help the astronomer view the data in a simpler way; sources might be enlarged based on their velocity in space. Changing the size-to-velocity ratio might help the astronomer identify sources that were harder to identify before and select them with ease.

Overall, parameterisation is a fundamental concept that provides flexibility and specificity in various scientific and engineering disciplines, enabling the analysis, design, and optimisation of complex systems and models.

4.2. Design Goals

Before designing the selection methods in VR and desktop, the goals of the design were defined as follows. The iDaVIE team defined these goals, and as such, received priority from us.

1. Accuracy: The selection must ensure precision in picking the desired data points or objects within the virtual environment. For us, accuracy, in this context, is how many particles were selected that were supposed to be selected compared to those that were not supposed to be selected. We define the accuracy in mathematical terms in section 4.2.1.
2. Fatigue: The method should minimise physical and cognitive fatigue, allowing users to work efficiently without strain over extended periods.
3. Fidelity: High fidelity in the representation and interaction with data is essential for creating immersive and realistic virtual environments. This means that the virtual setting must closely mimic real-world scenarios and the behaviour of data within those contexts. Such accuracy ensures that users can engage with the virtual world in a manner that feels authentic and responsive. In the context of this thesis, fidelity would be tested with regard to the selection method, rather with the virtual world itself.
4. Speed (Efficiency): The process should be swift, enabling quick and efficient data selection without lag or delay, enhancing productivity.
5. Iteration: The method should support rapid iteration, allowing users to modify selections and experiment with different scenarios quickly. Speed and iteration are closely intertwined in data selection and analysis, yet they serve distinct roles that enhance the overall efficiency and flexibility of the process. It ensures that the initial selection and data processing happen quickly, allowing for a smoother workflow that saves time and resources. On the other hand, iteration complements speed by offering the flexibility to rapidly refine and adjust these selections through multiple cycles. This capacity for rapid iteration allows users to experiment with various scenarios, test hypotheses, and make adjustments based on the insights gained from each cycle. While speed ensures that each of these cycles is completed efficiently, iteration ensures that the process is fast but also adaptable and precise, enabling users to focus on the most relevant data and insights. Iteration stems from direct manipulation defined by Shneiderman (Shneiderman, 1983).
6. Dimensionality (DOF): The selection technique must effectively handle multiple degrees of freedom, allowing users to manipulate and explore complex, multi-dimensional datasets.
7. Required Expertise: The selection must be easy enough for researchers without VR experience to use. As most astronomers are still accustomed to desktop software, the transition to VR should be seamless, ensuring that newcomers are not deterred from its adoption.
8. Scalable: The selection method must work effectively on extensive datasets and be scalable for real-life production projects beyond the theoretical scope of this thesis.

9. Implementation Complexity (IC): To support scalability and ease of maintenance and to conform to dissertation deadlines, the implementation (coding) of the selection method must not be overly complex.
10. Undo/Adjust: The system should allow for easy undoing and adjustment of selections, providing flexibility and room for error correction, making the experience of the user adaptable. The connection between iteration and the ability to easily undo or adjust selections is foundational to creating an adaptable process. The ability to undo and adjust selections is crucial for supporting rapid iteration, allowing users to experiment and refine their data analysis without penalty for mistakes. This flexibility encourages a trial-and-error approach, enabling users to learn and improve their selections with each iteration. The system becomes adaptable by facilitating easy corrections and adjustments, enhancing efficiency and creativity in the data selection process. Essentially, this feature ensures that the iterative process is not just about repeating steps but also about refining outcomes with each cycle, leading to more accurate and tailored results.

Before we continue to describe the design methods and solutions, accuracy and efficiency must be described mathematically to measure the design's success in fulfilling these goals.

4.2.1. Accuracy

Given the formula based on (Metz, 1978; Stehman, 1997):

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{N} \quad (1)$$

True Positives (TP): The particles that were supposed to be selected and were correctly selected.

True Negatives (TN): The particles that were not supposed to be selected and were not selected.

False Positives (FP): The particles that were not supposed to be selected but were selected.

False Negatives (FN): The particles that were supposed to be selected but were not selected.

Notice that $TN = N - (TP + FP + FN)$, as long as N is finite.

This calculation methodology considers both the particles correctly identified and those correctly left unselected, offering a comprehensive insight into the user's overall performance. However, it harbours a significant drawback: Given the characteristics of the available data and the limited scope of the selection sets, the value of N amounts to several hundreds of thousands of particles.

Meanwhile, a real-life-like selection set comprises a few hundred to thousands of particles. This disproportionality quickly overshadows all the other variables involved in the calculation, potentially yielding a misleading accuracy score. In essence, it may portray the user as highly accurate, even in cases where they might have selected zero particles, thus not truly reflecting the precision or the skill exercised during the selection process. This flaw necessitates revising or adjusting the calculation approach to attain a more accurate representation of user performance.

We modify the formula as follows to meet our needs: $Acc' = \frac{TP}{TP + FP + FN}$ (2)

This formula omits the true negatives (TN) entirely, as it serves as an abstract component that does not align with the primary objective of the selection process. Our focus is not dispersed across the entirety of the particles in the dataset; instead, we are solely concerned with the particles the user has selected. By omitting the TN component, the formula intends to offer an accuracy measure more relevant to the targeted selection task, centring its attention predominantly on the positive class - the category that holds substantial significance for us in this context. This adjustment in the formula aims to facilitate a more precise evaluation of user performance, highlighting their skill in distinguishing and selecting the correct particles.

4.2.2. Efficiency

In our case, efficiency can be described as the time it took to select the particles. A selection is finished when the user selects all the particles they intended to select.

Efficiency and accuracy offer a trade-off; Opting for the swiftest, albeit most simplistic method, such as selecting the entire dataset, would result in a highly efficient yet inaccurate outcome, as can be quantified using the specified accuracy formula. Given that all particles are selected, $FN = 0$ as there are no false negatives and $FP + TP = N$, which can be rewritten as: $FP = N - TP$.

$$Acc' = \frac{TP}{TP + FP + FN} = \frac{TP}{TP + (N - TP) + 0} = \frac{TP}{N} \quad (3)$$

As TP can be at most a few thousand, but N is hundreds of thousands, the accuracy will be close to 0. This trade-off operates in both directions; if the user opts to meticulously select particles individually, rectifying errors by deselecting inaccurately chosen particles (thereby reducing the false positives and false negatives) and ensuring each selection is precise, they could achieve flawless accuracy. However, this approach compromises efficiency, necessitating a considerably longer selection time.

4.3. Software Specifications

To realise the design goals, answer the research questions in Chapter 1.2 and evaluate it scientifically, we would need to develop VR and desktop applications. The reasoning behind this is to be able to compare the two and draw conclusions. Therefore, the applications must look and feel the same, but each should utilise the best practice of its domain features. While on desktop, the process is quite straightforward as the development of 3D visualisation on desktop has spanned a few decades, with prominent software like Blender leading the market. The technology is still quite novel in VR. It requires detailed planning and design before implementation. Therefore, the next two chapters detail the design process for each platform and each feature, selection and parametrisation.

4.4. Virtual Reality Application

4.4.1. Locomotion

Developing and designing the locomotion system of iDaVIE-p was not part of this work, as the iDaVIE team already developed it. Nevertheless, locomotion in VR is extremely important for the selection process, as it is an integral part of it; the user must first find the particles before selecting them.

iDaVIE-p locomotion is unique in its mechanics and requires some time to master. Pressing *one* of the grip triggers allows the user to translate the dataset in space. Pressing *both* grip triggers and opening the arms allows the user to zoom into the dataset. The more zoomed the user is, the slower the movement of the dataset is. If both grip triggers are pressed and the user closes their arms, it will cause the dataset to zoom out. If both grip triggers are pressed, and one controller is in place, but the other is moving around the first controller, it will cause the dataset to rotate.

Effectively, only grip triggers are relevant for the locomotion in VR.

4.4.2. Selection Design

A mock particle dataset was created using Blender to test different design ideas. This dataset is small but is made to mimic three different selection scenarios based on difficulty: easy, medium and hard, as seen in Fig. 15.

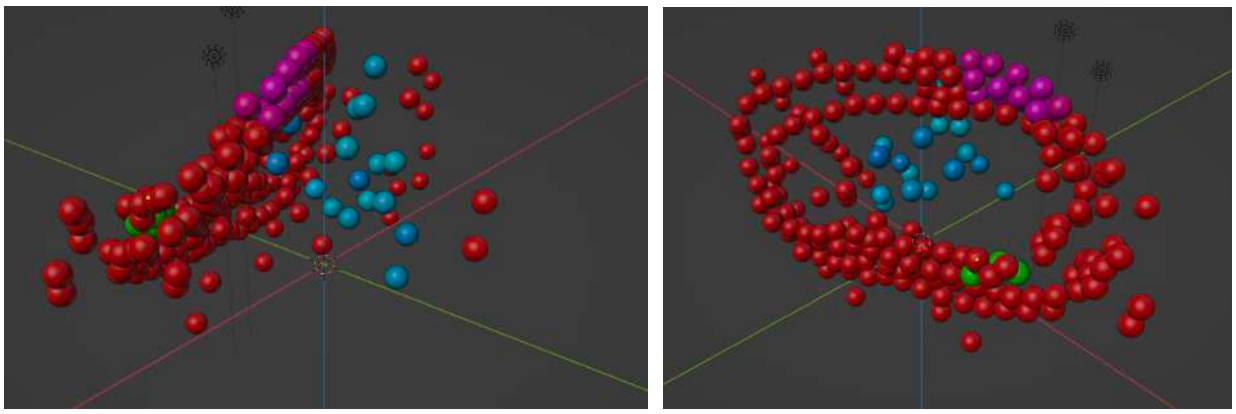


Fig. 15. Different perspectives of the mock dataset; red are particles that should not be selected, purple particles are easy in selection difficulty, blue particles are medium in selection difficulty, and green particles are hard in selection difficulty

An extensive subset of the dataset, marked in red in Fig. 15, should not be selected. This represents the real-life scenario where the researcher does not want to select most of the dataset, as it is irrelevant to them. The purple-coloured particles are easy to select. This is due to their location in space; they are positioned on the outer side of the "doughnut-shaped" data, which makes it relatively easy to select them without the need to view the data from different angles and without selecting red particles by mistake. The blue particles represent particles that are neither easy nor hard to select; we call this "medium difficulty". These particles are not easy to select as they are within the centre of the "doughnut", meaning that if the user wants to select, they need to rotate the data and view it from different angles to ensure all blue particles are selected. On the other hand, they are not hard to select, as they are not behind or blocked by any red particle. Finally, the green particles are hard to select; red particles obscure them, so the user needs to rotate the dataset to view it from different angles and be extremely careful when selecting so no red particles are accidentally selected.

4.4.3. Boolean Operation Spheres

We use the notion of a select "brush" to select particles. Like a real brush that paints over a canvas or a 2D brush in painter software, a 3D select brush allows the user to hover over the particles and "paint" or create a shape. Any particle that is contained within this shape will be selected. We use boolean operations (see Appendix F) to determine if the particle is contained within the shape. Many shapes are a good fit to use as the select brush, but the sphere was the most intuitive in our testing, consisting of a cube and a sphere as brushes. In the future, allowing the user to changing between brush shapes can be considered. A selection sphere allows fine selection due to its curvature, conforming to the accuracy and fidelity goal. Its unique shape in the 3D space also allows it to perform the three Boolean set operations easily and intuitively: union and difference. Most graphical visualisations of these three boolean operations use the 2D version of a sphere: a circle, as demonstrated in Appendix F. Transforming this into a 3D world is relatively easy. It would be familiar to the user as many drawing applications use a sphere (or a circle) as the brush, reducing

the required expertise as per the design goal. A sphere also operates as an atomic unit. Combined, it can create the union operation or, when subtracted from each other, the difference operation. This atomic behaviour can allow the undo/adjust goal by reverting unwanted spheres and the iteration goal to allow the rapid creation of more spheres. Implementing a sphere-like brush is also quite simple, which aligns well with the implementation complexity and scalability goals.

To realise the union operation in the 3D space, we use two or more spheres that share surface area. Any particle within the spheres or their shared surface area is considered “selected”. As shown in Fig. 16, the two spheres combined select all the “easy” particles coloured in pink.

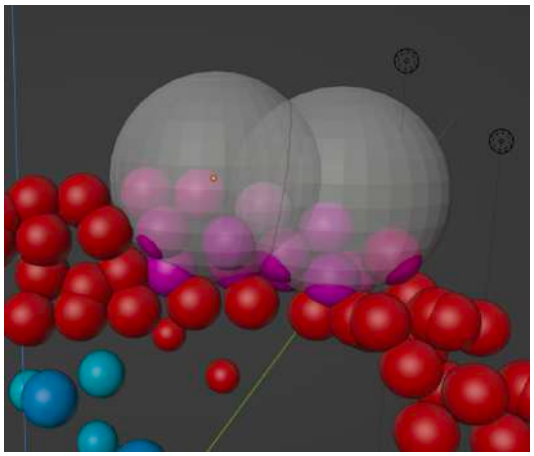


Fig. 16. Two spheres create a union operation to select all particles within them or their intersection

Of course, if the selection is complicated, more than one sphere can create different shapes and unions. For example, as seen in Fig. 17, more than two spheres are required to select the blue (“medium”) particles.

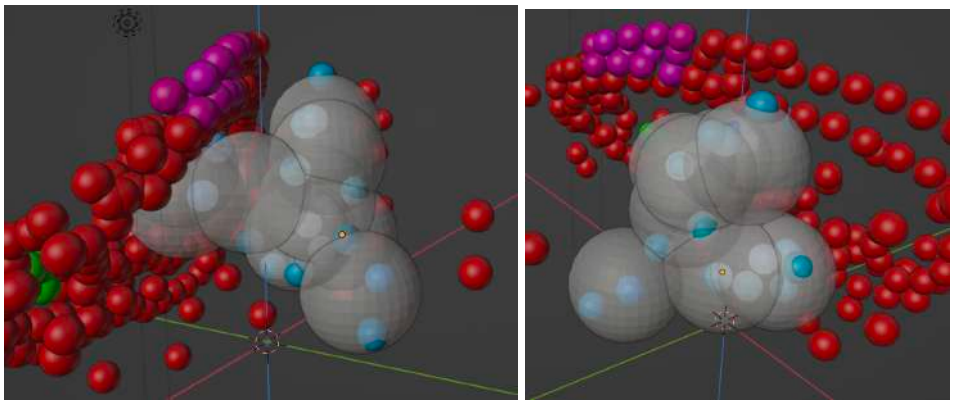


Fig. 17. Multiple spheres create a union operation along different axes

In the original Blender design, all the selection spheres have the same radius. This radius should and must be changeable, allowing a finer or bulkier selection, affecting speed and accuracy goals; a

finer selection would be accurate and slow, while a bulkier selection would be inaccurate and fast. This tradeoff between efficiency (speed) and accuracy is discussed in section 4.2.

Indeed, a user will want not only to *select* but also to *deselect* particles. For that, we introduce another mode to the selection sphere: deselection. This sphere deselects all particles that are within it, and it takes precedence over the selection sphere. Practically, it means that if the user selected particle A and the same particle is within a deselection sphere, the particle will not be selected. When a deselection sphere interacts with a selection sphere, we modify the selection sphere shape using the difference operation. This can create interesting shapes that allow the user to increase their DOF, fidelity and accuracy per design goals.

Deselection spheres also serve as part of the undo/adjust design goal, as they allow the user to undo their selection or use the difference operation to modify a current selection, as seen in Fig 18.

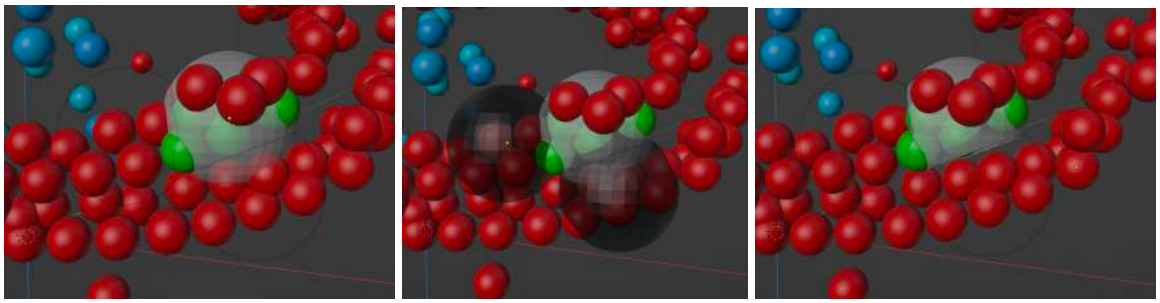


Fig. 18. From left to right: a selection sphere is applied over the "hard" particles but is inaccurate and contains unwanted particles. Deselection spheres are added to modify the selection using the difference operation, and a new selection volume is formed containing only the wanted particles

4.4.4. Spatial Deformation Cylinder

Unlike the boolean operations, a spatial deformation technique would involve shaping and adjusting the space around the objects, causing the objects to change shape. This indirect modelling technique does not manipulate the space directly but uses points, surfaces, curves and volumes as control mechanisms to allow the deformation (Gain & Bechmann, 2008), as seen in Fig. 19.

Fig. 19. From left to right: Spatial deformation based on surface, points and curves as dimensions (Gain & Bechmann, 2008)

This allows local and global deformation, making it a versatile and intuitive method. Many users use spatial deformations without calling it such; for example, Blender cylinder mode uses points as

a dimension to manipulate the space around the cylinder and adjust its curvature. Due to its popularity and accessibility, we consider an approach similar to Blender. Using a cylinder, we allow the user to select all particles that contained within it. The cylinder is an intuitive shape that most users have encountered in 3D editing. A box could be an alternative to the cylinder but has flexibility limitations, as a cylinder has round corners, allowing it a more precise selection. As seen in Fig. 20, a simple cylinder is sufficient to select the easy particles.

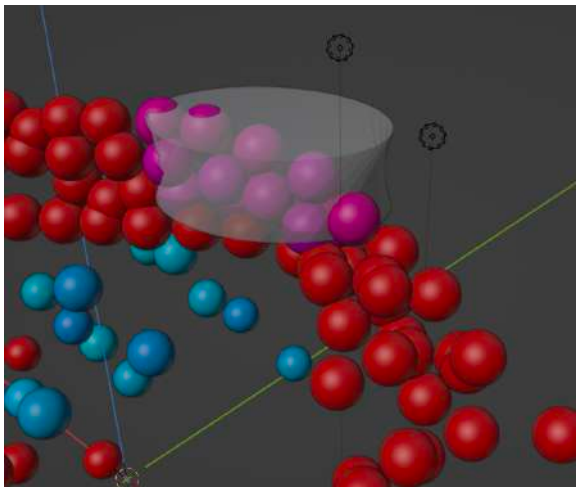


Fig. 20. A simple cylinder with no curvature is enough to select the purple “easy” particles.

If the user wants to select a set of more complex particles, for example, the blue particles, they have to use spatial deformation to change the cylinder’s shape. While spheres represent a single atomic object that, together with the boolean operations, can form a more complex unified object, an object such as a cylinder that is adjustable using spatial deformation points is a single object that does not require other cylinders to change its shape; a simple deformation of the space around the cylinder can suffice. As already mentioned, we use points as handles for the deformation. Each point is automatically created every time the user wants to start a new curved segment of the cylinder by clicking a button, as seen in Fig. 21. In geometry terms this is referred to as a generalised cylinder.

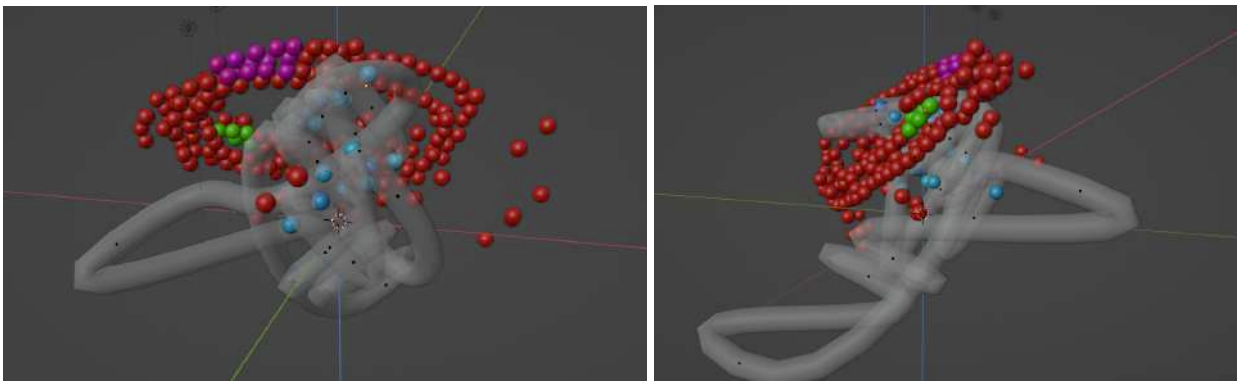


Fig. 21. A complex cylinder using spatial deformation points (denoted as black points) to select all the blue “medium” particles.

This selection method offers excellent scalability and few iterations, as even with complex datasets, all that is needed for selection is a single cylinder. This low number of iterations can also affect fatigue directly and positively. This method would also be accurate, as this cylinder can be deformed as many times as needed to select all the wanted particles (as seen in Fig. 21). Here lies its main drawbacks: speed and the ability to undo and adjust. Adjusting and deforming the cylinder takes time, as each segment modification must be thoroughly thought out before implementation. Undoing a deformation is possible, as they are created discretely through (point) placement, but it might require undoing many points, depending on where the curvature went wrong. Implementing this method is also relatively simple, as there are not a lot of objects to deal with, just a single cylinder. The expertise needed for this method is extensive from the user perspective, as this might be unfamiliar to most users without a 3D design background and might pose some learning curve. As for the degree of freedom, this method ranks high as there are almost no limitations to space deformation. The only limitation comes from where this deformation occurs, and that is the auto-generated points along the cylinder.

4.4.5. Comparison

Before implementing the desired design, we chose to consider which design scores higher regarding the design goals. This comparison is not objective; it is impossible to have an objective user-experience comparison, as it inherently depends on the user experience. To make the comparison more robust, we met with the iDaVIE team and supervisors from the computer science department of UCT. Each design goal was then ranked from 1 to 5, with 1 being the lowest (worst) and 5 the highest (best).

- Accuracy: Both methods rank with 5, as both can offer considerable accuracy; the boolean operation sphere can become extremely accurate the more spheres the user adds, while the spatial deformation cylinder can be very accurate if deformed in a way that encapsulates all the particles.
- Fatigue: The boolean operations sphere ranks as 4, as creating spheres is quite simple. The user must keep the button pressed, and more spheres will be created in union or difference mode. This is very similar to painting software, where the user holds the button to use the brush. On the other hand, the spatial deformation cylinder ranks low at 2, as it is quite a cumbersome process to deform it to the exact shape needed. We know it from our day-to-day activity with images on our phones: drawing a shape is much easier than cropping an image into a shape.
- Fidelity: Again, the boolean operant sphere ranks higher with 4, closely mimicking a natural selection action. The user must “paint” over the particles they want to select, and as long they are within the sphere, they will be selected. The spatial deformation cylinder, on the other hand, is less intuitive and has lower fidelity. Hence, it ranks as a 2. The user must deform the cylinder

until particles lie within it, a process that is not as natural to most users and is mostly reserved for graphic designers or people with similar expertise.

- Speed (Efficiency): As before, the boolean operations sphere ranks higher with a 5. It is extremely fast: the user clicks, and a sphere appears. The user holds the button, and spheres appear until the button is released. Because the spheres are atomic, the selection process is as quick as creating a sphere. The spatial deformation cylinder ranks low with a 2, as the deformation process is slow and prone to errors due to the many points created on a cylinder and the need to reposition them.
- Iteration: The spatial deformation cylinder is the clear winner here with 5, as it requires one single iteration to create the cylinder. The boolean operations sphere can also require one single iteration in the best case, but in the worst case, it can require N iterations, where N is the number of particles in the dataset. This ranks the sphere with a very low iteration score of 1.
- Dimensionality (DOF): Both selection methods rank with a 5 as both allow the user the same freedom of selection on complex datasets. The dimensionality of the set is represented through the parameters on the particles (such as colour or size) and has no effect over the selection methods, which use the particle's position in space to select.
- Required Expertise: Being an atomic process, the boolean operations sphere is much easier to adapt to and to learn, requiring low expertise on the part of the user. It is very straightforward and intuitive, ranking as a 5. On the other hand, the spatial deformation cylinder requires some modifications from the user. After modifying, the user must also use the locomotion to inspect the cylinder from different viewpoints and control it on three axes. This process is cumbersome and requires a process of learning for users who are not experts in 3D editing, ranking it with a low score of 2.
- Scalable: Both selection methods are easily scalable. The boolean operations sphere is atomic and can scale very easily. The spatial deformation cylinder can also scale to big datasets easily as it can be modified to any shape and size the user requires. Both methods receive a score of 5.
- Implementation Complexity (IC): The boolean operation sphere is easy to implement. The code, being an atomic operation, makes its complexity very low; implementing a boolean sphere operation for 10,000 particles is the same as implementing it for one particle - the logic is the same, and therefore, it scores 5. The spatial deformation cylinder is not complex to implement but requires more code complexity due to the axis handles and the deformation part. It is scalable, which makes the code simple but more complex than the atomic boolean operations sphere. Therefore, we ranked it with a 4.
- Undo/Adjust: Undoing and adjusting the boolean operations sphere is very simple, as it is an atomic operation. The creation of a selection sphere involves an entry in the list of operations. If the user wants to revert the last action, we revert the last sphere created and select or deselect the particles within it. This can be done for every sphere created with the same code. Because of that, this selection method ranks as a 5. The spatial deformation cylinder, on the other hand, ranks

very low with 1. This is because it is not atomic; once a cylinder is created, it can only be adjusted and rescaled. It requires many iterations until the user gets the perfect shape that selects all the desired particles.

Based on the scores, which can be compared in Tbl. 4, we have chosen to implement and integrate boolean operation spheres in the VR version of iDaVIE-p.

	Accuracy	Fatigue	Fidelity	Speed	Iteration	DOF	Expertise	IC	Scalable	Undo/ adjust
Boolean Operations	5	4	4	5	1	5	5	5	5	5
Spatial Deformation	5	2	2	2	5	5	2	4	5	1

Tbl. 4. Comparison of scores across design goals between the two selection methods. Green cells show superiority, red shows inferiority, and yellow shows no clear winner

4.4.6. Selection Depth Cues

Depth cues, especially in VR, are extremely important when selecting particles. We face a dilemma of which depth cues to use; too many can saturate the view and confuse the user, and too few can make it harder for the user to understand which particles are selected and their relative spatial position. Because our background is black, the perception of a particle's depth heavily depends on its depth cues. Therefore, we introduce a few ideas and implementations of depth cues into our software.

The particles, being small discrete objects, can benefit from kinetic depth (Ware, 2013). Therefore, animation of the particles, once selected, could be considered. We tested a gentle moving animation that moves the particle on the X-axis a few millimetres back and forth. This worked well but was very confusing when the user zoomed out and looked from afar; many particles moving together created fatigue and confusion. In addition, we tested proximity luminance covariance. In other words, particles closer to the user have higher contrast and opacity than those further away, giving them more depth. This worked quite well and was implemented in the final product. Finally, as shown in the Related Work chapter, one of the most effective ways to estimate a particle's position and size in space is with regard to the ground plane. For that, an artificial spatial cue in the form of lines from the ground to the particles was considered. We did not implement this method, as the user usually selects thousands of particles and giving all of them ground lines can clutter the view of the user. As mentioned in the Related Work chapter, point clouds, or particle clouds, do not benefit from perspective cues but can benefit from stereoscopic depth and structure-from motion cues. We use this structure-from-motion cue in our VR design as the locomotion system moves and spins the whole dataset, practically giving the particles kinetic depth. In addition, we also added

shape-from-shading, as suggested in the Related Work chapter. This can be achieved by treating each particle as a flat, oriented object. If a particle lies near the boundary of the cloud, it can be shaded to increase the overall depth perception of the cloud. We found this method efficient and easy to implement, so it was included in the final product. Of course, the size of the particles also creates depth, as mentioned in the Related Work chapter; particles that are further away from the user are smaller. In addition, in future work, a dynamic rendering can be developed that allows particles that are far away to be dense, and particles that are close by to be sparse by changing their size.

4.4.7. Selection Implementation

To integrate the selection method, we needed to build it onto the existing iDaVIE-p code and apply some core modifications to allow it to work seamlessly. All of the code is written in C# and tested through Unity; since, as already mentioned, iDaVIE is developed on Unity. The most important aspect of the implementation is that the particles are rendered through the GPU directly via HLSL shaders. The particles are rendered that way mostly for performance reasons: if done directly on the CPU, the software can have significant slowdowns and lag when rendering a large dataset.

First, we read the data (either FITS or IPAC format) into buffer memory using ComputeBuffer [9], which serves as a GPU buffer that allows us to send data from the CPU to the shaders. The particle's ID, position (X, Y, Z), colour, size and shape are saved to the buffer. On top of these parameters, we also added two new parameters: a "selected" flag and a "test" flag. The "selected" parameter can have the values of 1 (selected), 0.5 (deselected), or 0 (original state). We chose to work with three states instead of a boolean state (1 or 0) because the original state is not equivalent to the deselected state due to the depth cues; the original state is affected by different transparency settings as per depth cues in the previous section. When all the particles are deselected, they are in their original state (opaque), but when selection begins, all the other particles move into a transparent state to allow the user to focus on the selected particles.

For example, as we start, all particles are in their original state (0) and with full opacity. Then, if we select a particle, it goes from state 0 to 1, making all other particles in state 0 move to state 0.5 (deselected) so they are dimmed (lower opacity). Then, if we deselect the particle we just selected, it would move from state 1 to 0.5 and be dimmed. If all particles are in state 0.5, we can safely move them back to state 0, so opacity would be reverted to full luminosity. This is demonstrated in Fig. 22. The "test" parameter is boolean, either 0 or 1, indicating if the particle is part of the evaluation process detailed in Chapter 5.

[9] https://docs.unity3d.com/ScriptReference/ComputeBuffer.html

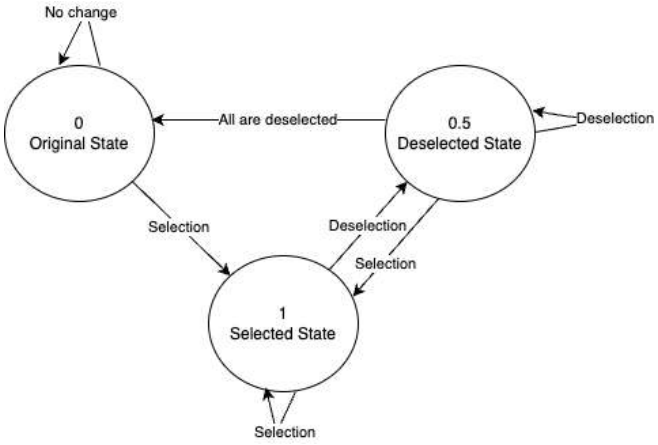


Fig. 22. State diagram of selection states

After we read the data, we initialise the controllers using the SteamVR Unity Plugin [10]. This plugin allows us to use different VR devices and is not developed specifically for Meta Quest. Both controllers have their logic, which can be swapped depending on whether the user is left-handed or right-handed. For the sake of simplicity, we consider the selection logic to be on the right controller for the rest of this chapter. The right controller has a "preview" sphere attached to the end of it, showing the user where the sphere will be spawned once they create it, as can be seen in Fig. 23.

Fig. 23. A preview sphere attached to the VR controller, next to a particle that needs to be selected

The user can switch between the union sphere and the difference sphere by clicking a button on the controller. The preview sphere will change its colour accordingly, based on the mode selected: green for union and red for difference. The implementation of the two modes is the same; the only difference is that we deselect the particles already selected in the difference mode.

To select a particle, the sphere must be in union mode, and the user must click a button to create "real" spheres rather than the preview one. When a sphere is created, we create a temporary

[10] https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647

GameObject [11] in Unity in the shape of a sphere. This GameObject has a material attached to it (either green or red, depending on the mode) and a radius of 0.5. The radius can be changed by manoeuvring the joystick upwards or downwards. There is a minimum value, to prevent the sphere from having too small a radius. There is also a threshold in place to prevent the sphere from scaling on every small joystick movement.

This GameObject sphere is created, but using a Coroutine [12], we fade out the material and destroy the GameObject, making the sphere last for only one second. This is done for two reasons:

1. UI/UX: the user expects the sphere to disappear and not stay and clutter the view, especially when there are many spheres. The user cares only for the selected particles, not the spheres.
2. Every GameObject allocates more resources on the CPU, taking up memory. Spheres are created in the thousands at a time and can potentially cause serious slowdown. Therefore, destroying the GameObjects after the selection process is necessary.

During this one second when the GameObject is active, we loop over all the particles in the cache and run the following formula:

First, we multiply the sphere radius (r) with a tolerance value 0.5. This is done to ensure fidelity. The tolerance value can be changed, but we found that 0.5 works best. We define $S_r = 0.5r$. Then, we calculate the offset vector between the sphere position S^p and the particle position P_i^p. This offset, $\overrightarrow{O_i}$, is a vector representing the distance between the sphere position and the particle position. We define it as $\overrightarrow{O_i} = S^p - P_i^p$.

Finally, we compare the squared magnitude of the vector $\overrightarrow{O_i}$, which is defined as $|\overrightarrow{O_i}| = \sqrt{O_{ix}^2 + O_{iy}^2 + O_{iz}^2}$ and the squared value of S_r which is defined as $S_r^2 = (0.5r)^2 = 0.25r^2$.

The implementation of this algorithm is summarised in Fig. 24.

Algorithm 1: Boolean Operation Sphere Selection

```
BooleanSphereSelect(r: sphere radius, S^p: sphere position, P: particle list):
L ← {}
for i=1 to |P| do
    P_i^p ← p_i //position
    O_i = S^p − P^p
    |O_i| ← sqrt(O_ix^2 + O_iy^2 + O_iz^2)
    if S_r^2 ≤ |O_i| then
        L.insert(p_i)
    else
        continue
    end
end
SelectParticles(L)
```

Fig. 24. A pseudocode of the selection process using the boolean operation sphere

[11] https://docs.unity3d.com/ScriptReference/GameObject.html

[12] https://docs.unity3d.com/ScriptReference/Coroutine.html

If $S_r^2 \leq |\overrightarrow{O_i}|$ we know that the particle P_i lies within the selection sphere, we add that particle to a list of particles, which is processed by a function called “SelectParticles”. Notice that S_r^2 is a constant and can be compared to all particles and calculated outside the loop, which helps with performance, given that it is impossible to change the sphere's radius while spawning new spheres (holding the button to select); the user can change the sphere’s radius only after a selection batch is done. As its name suggests, the function “SelectParticles” selects or deselects (depending on the mode) the particles passed to it as a list from the previous formula. Based on these particle indices, it fetches them from the ComputeBuffer and changes the selected flag based on the mode of the sphere. If the sphere is in union mode, it will always change the flag to 1. If the sphere’s mode is difference and the flag is already on 1, it will change the flag to 0.5. If the mode is difference and the flag is not 1, it will keep the flag intact in its current state. The logic behind this is that if we deselect a selected particle, we change it into the deselected state, 0.5. But, if the particle is already deselected or was never selected, we must keep it either in state 0.5 or state 0, depending on which state it was before. By updating the ComputeBuffer directly, we allow “real-time” feedback that immediately changes the particles' states (and their depth cues) through the GPU.

One noticeable drawback of this method is that we loop *over every* particle in the dataset every time a sphere is created, and as mentioned, they can be created by the thousands every few seconds. This loop is relatively fast as it reads the cache data. In our testing on the 2MRS dataset, this loop took a few milliseconds on the minimum hardware requirements. While other and bigger datasets exist, this simplification was sufficient for the sake of this work. The Future Work & Improvements chapter explores more complex implementation ideas to make this loop method more efficient, mainly focusing on different search algorithms in the cache table. Once a particle has been selected via the shader, we change its opacity and potentially also its shape. This is per the depth cues chosen in the previous chapter.

Unity Implementation

Our shader pipeline, IDIA/CatalogPoint, consists of three sub-shaders: vsPointBillboard, gsBillboard and fsSprite. vsPointBillboard takes a particle and calculates its world position by scaling and multiplying its position with the dataset matrix. Then, depending on whether the particle has a uniform point size, the shader loads the relevant size from either a settings JSON file or from the passed point size in the ComputeBuffer. Then, it calculates the up and right vector based on the cross-product of the camera direction and up vector. As with the point size, the shader changes the colour, opacity and shape. It is worth noting that the colour, if not uniform, is based on a colour map file provided by the user.

The next sub-shader, gsBillboard, creates a sprite with four vertices representing the particle in 3D space. The offsets are 0.5 in each direction, creating a small sprite. Depending on the particle’s opacity and position from the previous sub-shader, the sprite changes its colour and opacity. The sprite billboard origin is calculated based on the right and up vectors, multiplied by the X and Y offsets, respectively.

Lastly, the fsSprite sub-shader creates the 2D texture and a vignette effect, so the particle's edges will not be pixelated. The final result of the rendering can be seen in Fig 25.

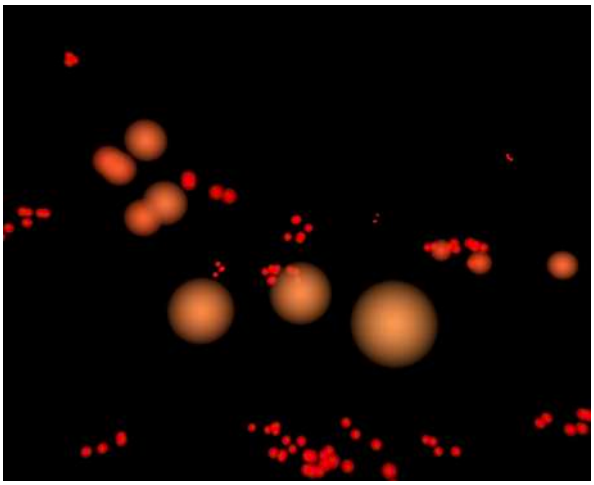


Fig. 25. Particles rendered through the GPU with a vignette effect

It is worth mentioning that our implementation supports only a cartesian coordinates system for the particles. The possibility of supporting spherical coordinates is reserved for future work and is mentioned in the Future Work & Improvements chapter.

4.4.8. Parameterisation Design

Before designing the parameter menu, we considered previous work (see Related Work chapter), especially computer games such as Half-Life: Alyx, which already utilise complex in-game menus that allow users to change parameters "on the fly" without exiting the game. Of course, there are many interaction interfaces to consider when investigating VR designs (see Appendix A), and there is no one solution that fits all in this case (Thériault et al., 2004). An interface that works perfectly for Half-Life: Alyx might not work well in our scientific context. We conducted an analysis based on internal testing and user experience. As part of future work, we plan to investigate more interfaces and compare them for accuracy and efficiency. We focused on the Speech Command Interface (SCI) and Manual Command Interface (MCI) designs. This section elaborates on why these two interfaces were chosen and their respective design in the iDaVIE ecosystem.

Speech Command Interface (SCI)

iDaVIE's main characteristic is its big data handling; it processes and displays large volumes of data in VR. Most big data, but specifically astronomical data, requires the modification and reading of decimal numbers, known as floating point numbers in Computer Science.

Editing floating point numbers poses a challenge in VR, as it requires very fine editing, which can cause fatigue or inaccuracies. One obvious solution is to allow the user to edit the floating number

via a virtual keyboard, similar to what we are used to with our phones. But this is a poor solution due to prolonged mid-air gestures, causing fatigue and the "gorilla-arm" effect due to the arms being suspended in the air while typing (Hansberger et al., 2017). We want to avoid implementing desktop-like solutions into VR, and we would rather utilise the power and novelty of VR to our benefit. For these reasons, we chose to use the SCI. SCI is very efficient and has recently become more accurate with the help of artificial intelligence (AI). Powered by Microsoft Azure Speech to Text recognition AI [13], we designed a Speech Command Interface that allows the user to speak into the HMD's microphone as input and receive the textual representation of the decimal number as an output, e.g. "Change size to one dot zero five four (1.054)". In addition, we designed the interface to support simple voice commands such as "Increase size by zero dot zero one (0.01)". We planned to extend the design to support ambiguous, more human commands such as "Increase size by a little", but these commands require defining the ambiguous words, such as "little", and therefore, due to time constraints and most likely low usage within the scientific use of iDaVIE, were dropped out. The idea behind designing the SCI was to allow the user to enter the initial decimal value without using their arms and to prevent the "gorilla arm" effect. After using the SCI, the user can modify the number via the MCI in cases where the AI interprets the number wrongly.

Manual Command Interface (MCI)

This interface was designed and inspired by VR games and virtual menus used by companies like Apple and Meta, as well as research that has been done in the field (Bowman & Wingrave, 2001). The menu, showing all the settings and numbers, is attached to the user's controller hand object, as shown in a mock in Fig. 26.

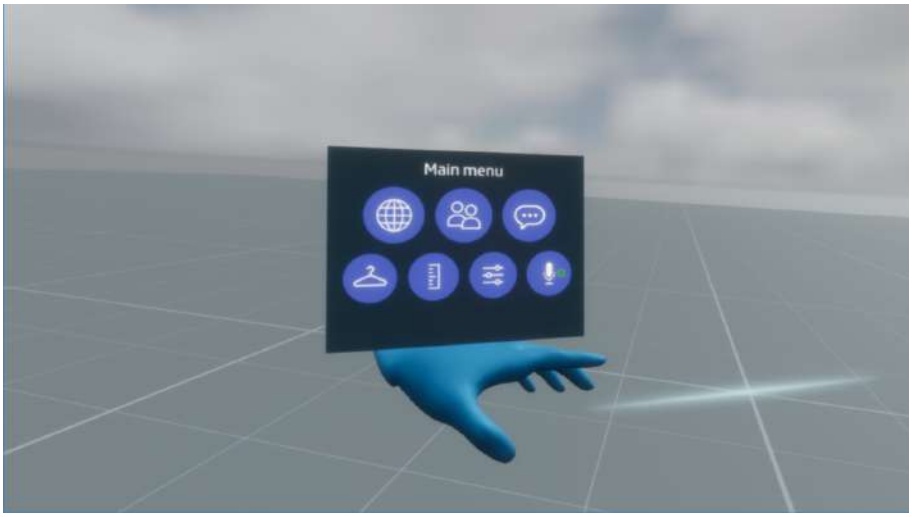


Fig. 26. A mockup of a menu attached to a VR hand

[13] https://learn.microsoft.com/en-us/azure/ai-services/speech-service/rest-speech-to-text

By doing this, we allow the user to see the menu at all times with intuitive control over its position; all the user needs to do is look at their hand as if they wore a hand-watch. This simple design allows us to support both right-handed and left-handed users alike, as all we need to do is switch the controller to the interface to which it is attached. We introduce a manual editing mechanism using the joystick to support editing and fine-tuning of decimal numbers (parameters). This also supports a bi-modal interaction, which can be more accurate. Apple's digit selector is an inspiration for this design. The user can select the digit they want to edit by moving left or right using the joystick, and then by moving up or down, they can change the individual digits. This allows the user to edit the decimal number digit by digit, making it extremely accurate but not as efficient. Combining the MCI with the SCI as an initial input can introduce balance and help make the process more rapid.

4.4.9. Parameterisation Implementation

To integrate the parameter menu, we used Unity Canvas [14] and attached it to the controller object as a child object so that it would always follow the controller. Then, in the C# code, we detect the movement of the joystick. Movement left or right (on the X-axis of the joystick) is interpreted as switching between parameter settings. Once on a parameter, if the user clicks the trigger, the parameter is selected for editing. We implemented this to prevent editing parameters by mistake, which often happened in early testing. After the parameter is selected for editing, the user can move the joystick up and down (on the Y-axis of the joystick) to change between settings. If the parameter is of a floating number type, we allow the user to move left and right using the joystick to select a specific digit, then click the trigger to edit this digit, and then up and down using the joystick to move between 0 and 9. Once editing is completed, the user can click the trigger again, and the parameter is changed. At this point, we update the ComputeBuffer directly and send the update to the GPU shader, causing the dataset to update immediately with the new settings and allowing the user to effect real-time updates.

4.4.10. Controls

For simplicity, we use a right-handed control scheme. As mentioned, the controls can be switched easily to a left-handed scheme by activating a setting in Unity. Since we are developing on top of existing software (iDaVIE), the locomotion system was left untouched, and the controls are limited to it (mainly the grip triggers). We tested different layouts of controls, and this layout, as seen in Fig. 27, emerged as the most intuitive one. Comparison of different control layouts is beyond the scope of this work and is mentioned in the Future Work & Improvements chapter.

[14] https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UICanvas.html

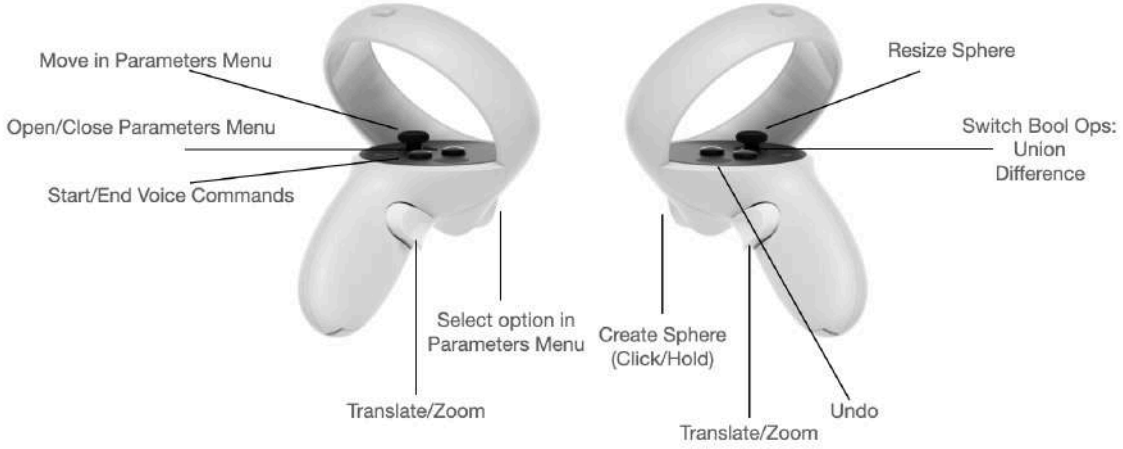


Fig. 27. The controls in right-mode mapped on a pair of Meta Quest 2 Controllers, in left-mode they would be mirrored

4.5. Desktop Application

4.5.1. Locomotion

The locomotion on the iDaVIE-p desktop is inspired by computer games, which also allow the user to move through 3D space, although they deal with different non-scientific contexts. The user can move forward, backwards and sideways using the arrow buttons. Using the W,S,A,D keys, the user can pan the camera upwards, downwards and sideways. However, in our context, panning sideways and moving sideways are the same.

The user can rotate the viewpoint by holding the SHIFT key and dragging using the mouse. The user can pan around by holding the SPACE key and dragging using the mouse (similar to W,S,A,D keys). Scrolling using the scroll wheel allows the user to move forward or backwards, similar to the arrow keys. This redundancy in controls is common in computer software, and we decided to adhere to the standard. As this locomotion system is very similar to computer games, most users who play games should be familiar with it, making the evaluation process easier.

4.5.2. Selection Design

Similar to the process in VR, we design selection methods for the desktop counterpart of iDaVIE-p. Unlike the VR design, this design is much less novel and is inspired by 3D editing software, like Blender. The most simple and obvious selection method for desktop applications is using the mouse pointer. A mouse click, which is used by every operating system (OS) to select, can be used to select a single particle. Similar to selection, deselection would also use the mouse click but with a different pre-defined mode. For simplicity, every “selection” mentioned in this chapter is also considered as “deselection” if the deselection mode is enabled. Unlike VR, where we use the sphere’s colour as feedback for the user on which mode they are on, on desktop, this setting is

highlighted in the settings box on the bottom left of the screen with a different icon, as modes are usually highlighted by a different icon in editing software such as Blender or Photoshop.

The big difference between this selection method and the one we designed for VR is that the user does not need to move close to the particle to select it; there is no meaning to the distance, as the mouse pointer uses only X and Y coordinates (screen coordinates), while the Z coordinate of the 3D space is irrelevant. This makes this selection method very efficient and accurate, as long as the user selects particles within view one by one (single selection). This, of course, can be hindered if there is another particle in front of the particle we want to select; in other words, this does not consider occlusion. In the case of occlusion, the user will still have to use the locomotion system. Of course, the user might want to select more than just a single particle at a time, similar to the selection sphere in VR. In this case, desktop software offers poor solutions, as there is a lack of depth perception, unlike in VR. Like Blender, we developed a 3D selection box that the user can translate and scale, as seen in Fig. 28.

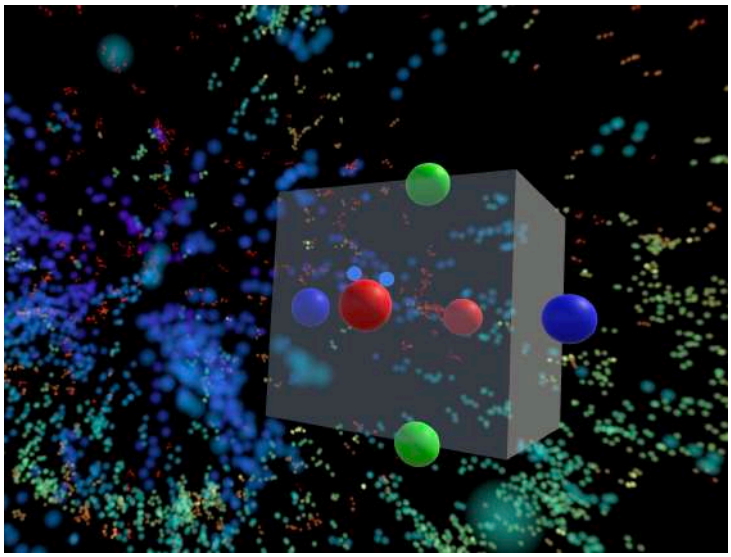


Fig. 28. The selection box has two particles selected within it, and the translation handles are on each side

This box is created by clicking on CTRL + Mouse click. It is semi-transparent so that the particles can be seen through it, and once a particle intersects it, it is highlighted via colour or opacity change. This gives the user a preview of what particles are to be selected. Once the user is ready to select, they can click ENTER and select based on the chosen mode (selection or deselection) or BACKSPACE to cancel the selection box. As mentioned, the selection box's size can be modified via scaling of each axis (X, Y and Z) or translated on each axis. This means the user has to position the box initially and then scale and move it around. Users must change their perspective when moving the box around to see how it selects different particles. This makes the process of selecting multiple particles rather slow and inefficient.

4.5.3. Selection Implementation

The implementation of the iDaVIE-p desktop version is much simpler than for VR. We developed the iDaVIE-p desktop version mainly for the evaluation part of this work and to be able to answer the research questions in Chapter 1.2. We do not assign too much importance to scalability for the desktop software at this point; this can be later developed (see Future Work in Chapter 7).

Unlike in the VR version, we load each particle as a GameObject and not through the GPU. This is not optimal for performance, as GameObjects use CPU power as well, but because the desktop version is used only for the evaluation process on specific pre-defined datasets, this drawback does not affect our use case. After the initial load, we use event listeners in Unity to listen to mouse clicks. Once the mouse is clicked, we use a Collider [15] to detect collision between the pointer X and Y position and the particle. If there is a collision, we select the particle and add it to a list of selected particles. If the selection mode is deselection, we deselect and remove the particle from the selected particle list. The user can switch between selection modes using a dropdown menu on the bottom left of the screen, as can be seen in Fig 29.

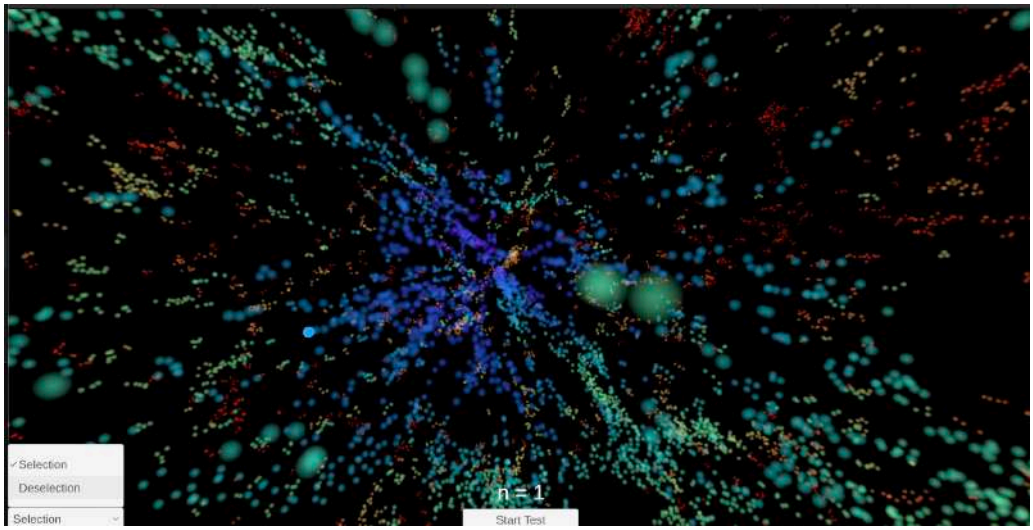


Fig. 29. A single particle is selected, changing its opacity and the selection mode menu open

The selection box is also a GameObject of type Cube, which operates similarly. Once created, we check which particles collide with the box (through its collider layer), and we select or deselect these particles based on the mode. This check is done on every frame update (Update() function) so we can highlight, using depth cues, the particles that are within the cube. Keeping the selection implementation simple and using only GameObjects with colliders and not GPU shaders allowed faster development time. While it is not scalable to millions of particles, it is "good enough" for the sake of this work and for the evaluation chapter, where we compare the desktop user experience to the VR user experience.

[15] https://docs.unity3d.com/ScriptReference/Collider.html

4.5.4. Parametrisation Design

Parameterisation on a desktop is a more straightforward process than VR, as it is a core feature of every OS. The design of this feature is already saturated and well-known to most users: a simple input field where the user can enter the desired decimal number using the keyboard. We will use checkboxes to represent boolean values (uniform colour on/off) and select inputs to represent predefined choices like axis column selection.

4.5.5. Parameterisation Implementation

Implementation, as with the desktop design, is straightforward. We use Unity Input Field [16] for text and select (predefined choices) inputs and Unity Toggle [17] as checkboxes for boolean inputs. Whenever the user changes a setting and clicks ENTER, we re-render the GameObjects with the new chosen parameters.

4.5.6. Controls

The controls, as mentioned, are inspired by computer games so that they will be familiar to users. On desktop, unlike VR, novelty, as far as controls go, is much harder to achieve. The control layout is demonstrated in Tbl. 5.

Control	Action
Mouse Click	Select or deselect a single particle
Mouse Click + SHIFT	Rotate the view
Mouse Click + SPACE BAR	Pan the view
Scroll Wheel	Move forwards or backwards
Up Arrow	Move forwards
Down Arrow	Move backwards
Left Arrow	Move left
Right Arrow	Move right
W	Pan up
S	Pan down
A	Pan left
D	Pan right
Mouse Click + CTRL	Create a selection box
BACKSPACE	Deselect all particles, or if the selection box is active, cancel it

Tbl. 5. The control layout for the iDaVIE-p desktop version

[16] https://docs.unity3d.com/Packages/com.unity.ugui@2.0/manual/script-InputField.html

[17] https://docs.unity3d.com/Packages/com.unity.ugui@2.0/manual/script-Toggle.html

Chapter 5 - Experiment Design

This project's evaluation and experiment design process is structured around a testing and feedback cycle. This cycle involves participants undergoing interviews centred on the usability, workload, and flow questionnaires to have deeper insights into their experiences with the final product.

User evaluation primarily compares the user experience between the standalone iDaVIE-p VR software, tested using an Oculus HMD, and the standalone iDaVIE-p desktop version, assessed through a standard desktop monitor with keyboard and mouse inputs. Initially, users are segregated into two groups: one tasked with testing the selection method and one tasked with testing the selection method after tuning some parameters, which we call “testing combined selection”, as it combines both selection and parameter tuning. These parameters can be, for example, the colour or size of the particles. Subsequently, participants will complete the System Usability Scale (SUS) (Brooke, 1996), NASA Task Load Index (NASA-TLX) (Hart & Staveland, 1988) and Flow State (Jackson & Marsh, 1996) questionnaires as outlined in detail in this chapter.

Each test is repeated twice but with a different set of particles to select. During the second round, the participants switched platforms, exploring the version they had not previously evaluated (Desktop or VR). Then, they revisit the questionnaires. During this evaluation process, users engage in selection and combined selection tasks, described in greater detail later in this document, on both the desktop and VR iterations of the iDaVIE-p system. This, therefore, constitutes a within-subject design with respect to desktop and VR but between subjects when considering the question of parameter tuning.

5.1. Hypotheses

The following hypotheses must be checked for both types of tests: selection and combined selection. Each test must fulfil the research question and, hence, the null hypothesis.

- Null Hypothesis (H0): There is no difference between the desktop and VR platforms in terms of (a) questionnaire scores, (b) time to complete tasks (efficiency), and (c) accuracy of task completion.
- Alternative Hypothesis (H1): There is a significant difference between the desktop and VR platforms in terms of (a) questionnaire scores, (b) time to complete tasks (efficiency), and (c) accuracy of task completion.
- Null Hypothesis (H2): There is no difference between the selection test and combined selection test in terms of (a) questionnaire scores, (b) time to complete tasks (efficiency), and (c) accuracy of task completion.
- Alternative Hypothesis (H3): There is a significant difference between the selection test and combined selection test in terms of (a) questionnaire scores, (b) time to complete tasks (efficiency), and (c) accuracy of task completion.

The difference between H0/H1 and H2/H3 is that the latter compares the aggregated data (VR and desktop) of the two tests, selection and combined selection. The reason for that is that we need H2/H3 to answer Research Question 2. As per the iDaVIE team, the impact of parameter tuning on selection is interesting from a broader perspective, not just VR/Desktop. In addition, if we compare VR and Desktop between the different tests, the sample size is too small to be significant. Therefore, H2/H3 are designed to compare the tests as a whole in an aggregated manner, while H0/H1 is designed to compare VR and Desktop, disregarding if it was in the selection or combined selection test.

5.2. Methodology

5.2.1. Treatment

A treatment for this experiment is using the developed selection method (boolean operations sphere) and the parameter tuning menu to select pre-chosen particles within a certain amount of time. We create four subsets of pre-selected particles from the whole dataset, each with 90 particles. 80 of which are in clusters (easier to bulk select), and 10 are scattered at the edges of the dataset (requiring locomotion and harder to select). Initially, we wanted to use 180 particles for testing, but after internal testing with the iDaVIE team, we noticed it was almost impossible to finish the test successfully within the given time. We decreased the number of particles until we reached 90, showing a good balance between difficulty and successful completion. The particles are selected at random. Each of the four pre-selected subsets of particles is a testing set: two for VR and two for desktop. The first subset is used as a training set, where the participant can train ("playground") on, as later discussed in section 5.4.2. The second subset is for the actual testing round. Having different sets of particles to select on desktop and VR eliminates any learning bias and familiarity with the dataset that can happen when the participant alternates between the systems. This allows us to compare the accuracy, efficiency and user experience objectively. As previously mentioned, the tests are timed, with an upper time limit (10 minutes). The participants can end the test beforehand once they have selected all the particles. The participant has to choose their strategy: either to go for accuracy and select particles carefully or efficiency and select as quickly as possible, sacrificing accuracy. Of course, the participant can choose an "in-between" strategy that balances accuracy and efficiency.

Upon successful selection, the selected particle is highlighted so the participant knows the particle is selected. In the combined selection test, we perform the same test (on the same dataset with the same preselected particles) but add another layer of complexity: parameter fine-tuning. The participant *must* edit at least one parameter through the parameters menu (in VR and desktop). This can be, for example, a change in the particles' size, which can help select the clusters. By changing the parameters, but keeping the rest of the test the same, we can compare later on the differences in accuracy and efficiency between the selection test and the combined selection test.

5.2.2. Participants

VR presents an exciting and novel experience, especially for users who are relatively inexperienced with this technology. To counteract any bias that could emerge in favour VR due to this novelty, we provide standard video tutorials and offer a specialised demo testing round within the VR setup, commonly called the "VR playground". This playground enables newcomers to test their skills, controls and locomotion of the iDaVIE system before embarking on the actual test.

Moreover, we select 18 users per test to gather balanced and informative feedback. The plan was to have 18 participants for the selection test and 18 for the combined selection test. That is, 36 users in total, broadly categorised into three distinct groups. There is no preference for astronomy students and researchers, as the selection tasks present no direct relationship to astronomy; it is purely a selection task. Rather, the focus is on the user's familiarity with VR, and they are first to be divided into three groups:

- Group A: Users with no or little VR experience (use VR zero to twice a year)
- Group B: Users who have tried VR multiple times (use VR twice to 12 times a year)
- Group C: Users who use VR extensively (use VR more than 12 times a year)

These categories are designed to temper the initial "excitement" new users often experience with VR, enabling a concentrated focus on the tasks. To ensure a balanced and fair analysis, we planned to mix non-professional users (students) with professional users (researchers) to mix diverse perspectives.

The three groups (A, B and C) are reshuffled into Groups 1 and 2. Each new group contains three researchers and three students from Groups A, B, and C. This ensures that the new groups, 1 and 2, are balanced regarding the experience of their participants so that no group can outperform another due to the prior experience of its participants. The balance between researchers and students is also kept intact to keep the previous knowledge within the group balanced. Fig. 30 shows the assignment process.

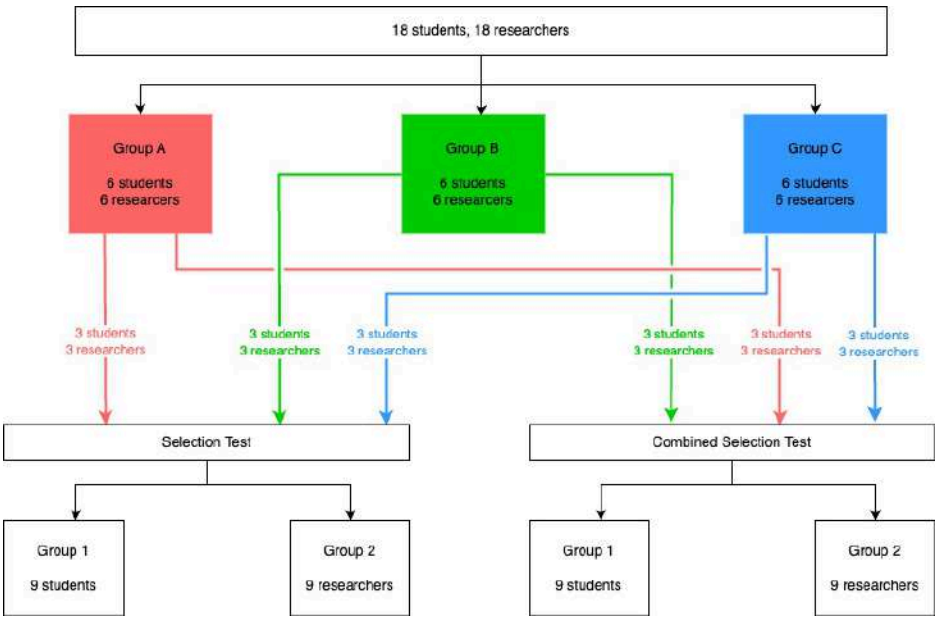


Fig. 30. Group shuffling (Plan)

Users who participate in one test cannot participate in the other test, so the shuffling of the groups is unique and performed only once. In other words, the two tests (with and without parameter tuning) are distinct and independent (between subjects).

To initiate the testing phase, Group 1 engages first with the iDaVIE desktop version, while Group 2 starts with the iDaVIE VR version. After the test, each group revisits the testing environment, exploring the platform version they have not yet experienced but with a different set of particles to eliminate the learning effect. Tbl. 6 shows the testing matrix.

	Group 1	**Group 2**
VR	Particle Set 1	Particle Set 2
Desktop	Particle Set 2	Particle Set 1

Tbl. 6. Testing Matrix

This alternating approach ensures that we gather comprehensive feedback, considering the experiences of both interfaces and users with varying levels of familiarity and expertise.

Because of different constraints and fewer users participating than expected, this original plan of having 36 participants was slightly modified, as shown in section 5.4.1, but the shuffling, the testing methods and the alternate testing matrix stayed the same, so the structure of the testing methodology did not change.

5.3. Measures

In this chapter, we introduce the measures that we used to evaluate the user's performance. These measures have been chosen carefully to allow us to answer the research questions thoroughly and meaningfully.

Accuracy

We use the following formula from Chapter 4:

$$Acc' = \frac{TP}{TP + FP + FN} \quad (2)$$

Where True Positives (TP) are particles correctly selected as intended, False Positives (FP) are particles selected despite not being intended, and False Negatives (FN) are particles not selected despite being intended. As noted in Chapter 4, we do not use True Negatives (TN) in our accuracy formulation.

In the best case, TP = 90, FP = 0 and FN = 0. This would result in 1.0 accuracy, or 100%.

Efficiency

As mentioned in Chapter 3, efficiency is the time it takes to complete the selection or, in our case, the test. The worst case would be 10 minutes (600 seconds). The quicker the participant is, the better. But as mentioned in Chapter 4 and demonstrated in formula (3), it is a delicate balance between speed and accuracy.

Usability

The System Usability Scale (SUS) (Brooke, 1996) is a simple yet effective tool used to assess the usability of various systems, including hardware, software, mobile applications, websites, and other products. Developed by John Brooke, the SUS has become a widely adopted standard in the usability industry due to its ease of use and broad applicability. The scale consists of a 10-item questionnaire with five response options for respondents, from "Strongly agree" to "Strongly disagree" on a Likert Scale. This structure allows it to gather subjective assessments of usability from users quickly and efficiently.

The beauty of the SUS lies in its versatility and simplicity. It can be applied at any stage of the design and development process, from early prototypes to finished products, providing insights into how users perceive the usability of a system. The questionnaire covers various usability aspects, including the effectiveness, efficiency, and satisfaction with which users can achieve their goals. After users complete the questionnaire, their responses are converted into a single usability score ranging from 0 to 100. Although this score does not represent a percentage, it indicates a product's

usability. Generally, a SUS score above 68 is considered above average, and scores below 68 suggest that a product's usability could be improved. The average SUS score of 68 was determined by aggregating and analysing SUS scores from numerous studies and systems, establishing it as a benchmark for average usability, though this figure is an empirical finding rather than an intrinsic part of the SUS methodology.

However, the SUS has several weaknesses. One primary concern is its lack of diagnostic capability; it provides an overall usability score but does not specify the nature or location of usability issues, making it challenging for designers to identify specific areas for improvement. Additionally, its one-size-fits-all approach may not capture the nuances of different user interfaces or experiences, potentially oversimplifying complex usability aspects. Finally, its scoring system, which translates to a 0-100 scale not directly correlating to percentages, can be confusing and misleading for those unfamiliar with its calculation and interpretation.

Despite its simplicity, the SUS has been validated through extensive research and is regarded as a reliable tool for measuring usability. It provides a quick, user-experience means for evaluating the usability of products and services, making it invaluable for designers, developers, and researchers aiming to create user-centred solutions.

Workload

The NASA Task Load Index (NASA-TLX) (Hart & Staveland, 1988) is a widely recognised tool for assessing the perceived workload of individuals performing various tasks. Developed by the Human Performance Group at NASA's Ames Research Center in the 1980s, NASA-TLX provides a multi-dimensional measure of workload, taking into account various factors that contribute to the overall load experienced by a person. The tool is designed to evaluate the workload of tasks across different domains, including but not limited to aviation, healthcare, and software engineering, making it versatile for research and practical applications.

NASA-TLX assesses workload through six key dimensions: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration Level. Each dimension is rated by the individual on a scale, typically from low to high, to reflect their perceived workload during or after task completion. The unique aspect of NASA-TLX is its ability to weigh each dimension according to its importance to the overall task, allowing for a more nuanced and accurate workload measurement. After the ratings and weightings are completed, the scores are combined into an overall workload score, providing a comprehensive view of the task's demands on the user. In our case, we gave all dimensions the same weight, as they all hold the same importance for iDaVIE-p further development.

This questionnaire is particularly useful in design and evaluation processes where the human factor is crucial. Designers and engineers can make informed decisions to improve efficiency, safety, and user satisfaction by identifying tasks or aspects of a system that impose a high workload. Its

application extends beyond traditional fields, proving valuable in any scenario where it is critical to assess and mitigate the impact of task load on human performance.

Flow State Scale

The Flow State Scale (FSS) is a psychological assessment tool specifically designed to measure the flow experience of individuals during particular activities. Flow, a concept introduced by Mihaly Csikszentmihalyi (Csikszentmihalyi, 1990), represents a state of peak enjoyment, deep concentration, and complete absorption in an activity where individuals lose their sense of time and self-consciousness, achieving high performance and engagement. The FSS aims to capture this optimal experience by quantifying the subjective feelings and perceptions of being in a flow state.

Developed for use in research and applied settings, the FSS includes several dimensions essential to the flow experience. These dimensions typically encompass aspects such as challenge-skill balance (feeling that one's skills are well-suited to the task's challenges), action-awareness merging (feeling that the actions performed are being done almost automatically without too much awareness), clear goals (understanding what needs to be achieved), unambiguous feedback (receiving immediate and clear feedback about one's performance), concentration on the task at hand (focusing deeply without distraction), sense of control (feeling in command of the activity), loss of self-consciousness (forgetting oneself and personal worries), transformation of time (sense of time speeding up or slowing down), and autotelic experience (engaging in the activity for its own sake, rather than for some external reward).

Respondents of the FSS rate their agreement with statements related to each flow dimension on a Likert scale, typically ranging from "strongly disagree" to "strongly agree." The FSS is highly regarded for its ability to provide insights into the conditions under which individuals achieve flow, facilitating strategies to enhance performance, enjoyment, and overall satisfaction in both personal and professional settings.

5.4. Procedure

5.4.1. Pre-experiment Phase

The pre-experiment phase included an online form sent to the astronomy and computer science departments of UCT. This form includes a short textual explanation of the test procedure and a video showing a sample test. The participants have to fill in their name, phone number, email address, birthdate, academic level (undergraduate, graduate, staff, etc.), UCT department, experience level in VR (as defined in chapter 5.2.2), frequency of playing computer games (weekly, monthly or rarely/never), if they suffer from epilepsy or are colour blind. The test could not be conducted if the participant has epilepsy or is colour-blind. None of our participants had either, so no screening had to be made.

In total, 39 participants completed the form. Of these, 11 participants did not attend the test and cancelled their participation. We were left with 28 active participants. As described in section 5.2.2, the participants are divided into three groups based on their experience in VR. Because we had a lot of cancellations, we decided to merge Groups B and C into what is now called Group B. Group A is for inexperienced participants. Group B is for participants with *some or extensive* experience in VR. Group A had 17 participants, and Group B had 11 participants. The two groups were shuffled together and re-divided into two new groups: normal and combo tests. The normal test group conducted a normal selection test as described in Chapter 5.2.1, while the combined selection test group conducted a selection and parameter fine-tuning test. The selection and combined selection groups are then divided again into two groups: 1 and 2. These groups alternate between the sets to to eliminate any VR bias and familiarity with the particle sets, as mentioned in Chapter 5.2.2 and demonstrated in Fig. 36. The new group divide, which differs from the plan, is shown in Fig. 31. Notice, that with the new divide, we do not give too much weight to the user's academic level, as it was nearly impossible to find a perfectly balanced participant group as was originally planned.

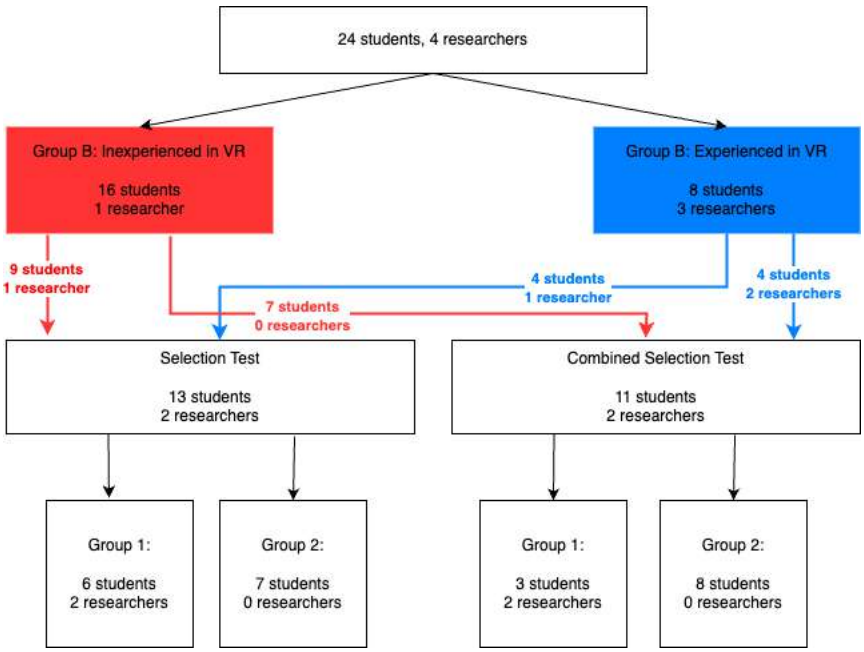


Fig. 31. Revised group shuffling

The obvious imbalance between researchers and students across groups 1 and 2 is less important than the overall number of people between these groups, which is more important for balanced testing with the alternate testing sets. The number of experienced and inexperienced participants, as well as students and researchers, is balanced between the two tests (selection and combined selection).

The testing sets were created randomly out of the 2MRS dataset. Four sets were created: two for the tutorial/playground (normal and combined test tutorial) and two for the actual testing. We chose 80 particles in clusters and 10 scattered particles to create the sets. To choose these particles, we first

tried to implement a Gaussian Distribution method to select particles close to the centre of the data, but this method was not fruitful as the differences between the four sets were minimal. The clusters ended up being very similar, with the same particles chosen every time. We needed the sets to be distinct to minimise the user's learning curve when moving between VR and desktop (or vice versa). Instead, we implemented a method that utilises the k-means algorithm. We calculate the global centroid of the dataset and then calculate the distance to this centre from every particle. Using these distances, we sort the particles and divide them into subsets based on their distance from the centroid. A clustering operation is performed for each subset using K-Means clustering, with k=5. We chose the number of clusters using the heuristic Elbow Method, as seen in Fig. 32.

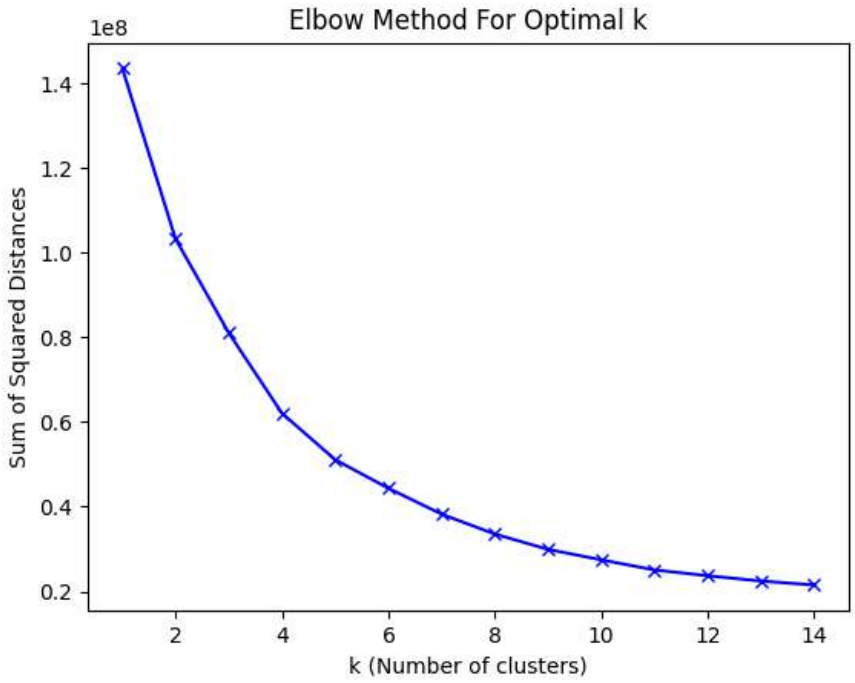


Fig. 32. Elbow Method for Optimal K using the 2MRS dataset

This way, unlike the Gaussian Distribution method, each subset, or cluster, is different from the one before and is based on the distance from the centre. This allows us to have four distinct sets of particles that users cannot remember, hence keeping the learning curve to a minimum. The script was written in Python3, and libraries such as sklearn and numpy were used. To choose the scattered particles, we chose the 1000 particles furthest from the centroid of the dataset, and from them, we randomly chose 10 particles.

5.4.2. Experiment Phase

VR Playground

As noted earlier, many users might be unfamiliar with VR in general or find iDaVIE's distinctive locomotion system challenging. To bridge this gap, we allow users to practice using a mock dataset to select and deselect particles and adjust various parameters according to their preferences. This practice session, although time-bound, does not impose any specific objectives, allowing users to explore and familiarise themselves with the system at their own pace. During this preparatory phase, users are encouraged to consult with the researcher, posing any questions regarding the locomotion system and controls, thereby acclimatising themselves to the VR experience.

Selection Test Tutorial

In this phase, the user receives a tutorial video illustrating the proper task-completion procedure. This instructional video, which showcases a successful completion of the selection test during an easier round, is displayed on the appropriate platform corresponding to the user's setup, whether VR or Desktop. Throughout and following the viewing of this video, time permitting, the user is encouraged to pose any questions they may have.

Selection Test

The objective of the test is to select a subset of particles from the entire dataset accurately and efficiently. The particles that are available for selection are distinguished by an opaque white box surrounding them. Once selected, the box will turn opaque green. Auditory signals accompany successful and unsuccessful selections, facilitating immediate feedback on whether a particle is part of the predefined set. In VR, the controllers will vibrate on selection. Once all requisite particles have been selected, a distinctive sound will signal the conclusion of the test round. The user can then finish the test by pressing a button or modify their selection by deselecting irrelevant particles and then end the test by pressing the button. As mentioned, the set of particles to select is predetermined beforehand. The selection test will have two sets, each containing 90 randomly chosen particles: 80 particles in clusters and 10 particles at the edges of the dataset (lower or higher end of the distribution). The clusters will allow users to select many particles simultaneously but risk selecting unwanted particles. The particles at the edges of the dataset will force the user to navigate and select particles individually, as some might be obscured behind non-selectable particles. While the sets are predetermined randomly, they are still fixed and identical for all users, allowing us to compare them.

Combined Test Tutorial

This tutorial includes the selection test tutorial and an extra tutorial that illustrates a successful execution of parameter tuning. Much like the selection test tutorial, this tutorial is displayed on the platform the user is utilising, whether VR or Desktop. Again, throughout and following the video,

time permitting, users are encouraged to ask questions to gain a deeper understanding and clarification regarding the task at hand.

Combined Test

This test is the same as the selection test, with the same set of particles to select, set 1 and set 2. The only difference between the tests is that this time, the user must change the default parameters to a new set of parameters. This change must involve a minimum of one parameter. The user is able to change digits (such as colour string, scale factors, etc.) and select settings from pre-defined lists such as particle shape type (halo, circle, square) and axes (x,y,z). In addition, the user can change the boolean values of certain settings (true, false).

Changing these parameters might help the user with selection. For example, if the particles are clustered, the user can decrease their size and select them separately, as they will have more space between them. Another example is the particles that are on the edge of the dataset. If the user changes the value of axes, e.g., axis Y can represent velocity instead of luminosity, these particles might suddenly be at the centre of the dataset, making them much easier to select.

The accuracy and efficiency definitions we used in the selection test are also the same for this test. Like the selection test, this test encourages users to strike an optimal balance between accuracy and efficiency, compelling them to navigate and make choices that lean towards meticulousness or expediency during the testing period. This balance would test their ability to adapt and optimise their interaction with both versions of the iDaVIE-p system within the stipulated time frame. We introduce an interesting comparison metric between the tests and the two environments by allowing us to change the parameters in real-time (both on desktop and VR). This would allow us to answer the research question: "Does tuning the parameters lead to more accurate and efficient selection?”.

Test Schedule

The schedule, as shown in Tbl. 7, is strict and must be respected to allow participation in an orderly fashion and to avoid the tiredness of the users, which could affect the results. Both tests share the same schedule.

Task	Time Desktop (minutes)	Time VR (minutes)	Total Time (minutes)
Playground	5	10	15
Tutorial	4	2	6
Test	10	10	20
Questionnaires	10	10	20

Tbl. 7. Testing Schedule

Feedback and Questionnaires

Participants complete three online questionnaires - SUS (System Usability Scale), NASA-TLX (Task Load Index), and FSS (Flow State Scale). These are collected individually, along with their accuracy and efficiency scores obtained automatically. Although the responses are anonymous, each questionnaire includes crucial information about the user, including their assigned group and VR experience. The questionnaires are to be completed after the finalised test, and the same questions apply to the standard and the combined selection test.

5.4.3. Post-experiment Phase

After testing is finished, we collect the data into folders. Each participant has a separate folder, with a JSON file created automatically using Python code. This JSON file, called meta.json, consists of the scores this participant received in each domain: VR and desktop. The scores are for accuracy, efficiency, and each of the three questionnaires. Finally, all the JSON files are collected through a Python script that calculates the accuracy, efficiency mean, and standard deviation across all participants in VR and desktop. This calculation is done three times: one time for all participants across all tests, one time for the normal selection test and one time for the combined selection test.

Chapter 6 - Results

In this chapter, we analyse the results collected from the experiment. The results are analysed twice: one time to compare VR and Desktop, and another time to compare the selection and combined selection tests. As these comparisons are unrelated to each other to answer the research questions in Chapter 1.2, our analysis is separate. In the methodology section, we explain which methods we use to compare and analyse; we systematically analyse every metric defined by the research questions: accuracy, efficiency, usability, workload and flow. While accuracy and efficiency are based on auto-generated results, the usability, workload and flow are based on the questionnaire scores produced by the SUS, NASA-TLX and FSS questionnaires, respectively.

Lastly, we discuss the results and their interpretation and explore if these results are sufficient to reject our null hypothesis (H0 and H2) in favour of our alternative hypothesis (H1 and H2) so we can answer the research questions positively.

6.1. Methodology

To answer the first research question defined in Chapter 1.2, we compare the accuracy, efficiency and the questionnaires' (SUS, NASA-TLX and FSS) scores between VR and Desktop. This comparison aggregates both tests (selection and combined) and gives priority only to the system used. This is due to the relatively low number of participants on each test alone and since the research question focuses primarily on the difference between VR and desktop, not selection and combined selection.

To answer the second research question defined in Chapter 1.2, we do the same as the first question but compare selection and combined selection tests. This time, the comparison aggregates VR and Desktop, and we focus primarily on the differences between the two tests, disregarding differences between VR and desktop, for the same reasons stated before. In addition, the research question focuses on how parameter tuning affects selection, regardless of the system used, as it has little scientific value, as per the iDaVIE team.

Before comparing and answering the research question, we check if the scores received in the two groups (VR/Desktop or Selection/Combined Selection) are distributed normally (Gaussian). This will allow us to use a paired t-test if it is normally distributed or a Wilcoxon Signed-Rank test in case it is not distributed normally. To check the distribution, we use four methods: looking at the histograms, comparing the p-value of Shapiro-Wilk and D'Agostino's K-squared tests, and looking at the Q-Q plots. Looking at the histograms offers a first glance at whether the data looks normally distributed. We do not weigh this method in our decision if the data is normally distributed, as it is prone to human error, but it gives a good "first indication" of the data.

Then, we run the Shapiro-Wilk and D'Agostino's K-squared tests. If the p-value of these tests is below or equal to 0.05 (the common 5% alpha), it indicates that there is significant evidence to reject the null hypothesis of the test and that the data is *not* normally distributed. Due to the low sample quantity (28 participants), we give the Shapiro-Wilk test more weight than the D'Agostino's

K-squared in our decision for normality. The Shapiro-Wilk test is known for its sensitivity and power in detecting departures from normality, especially in small sample sizes. It assesses how well the data conforms to a normal distribution by comparing the order statistics (the observed data values sorted in ascending order) to the expected values of a normal distribution. This sensitivity makes it a preferred choice when the sample size is limited, as it can more accurately detect non-normality. On the other hand, the D'Agostino's K-squared test, which examines the skewness and kurtosis of the distribution to determine how much it deviates from normality, tends to perform better with larger sample sizes. While useful, its power to detect non-normality in small sample sizes is generally lower than the Shapiro-Wilk test.

Lastly, we plot the Q-Q (Quantile-Quantile) plot to verify our results. Again, this plot does not receive as much weight as the analytical tests performed before, but it gives us an indication of the data and supports our decision for normality. A Q-Q plot is an insightful graphical technique used to determine whether a dataset follows a particular theoretical distribution, commonly the normal distribution. By plotting the quantiles of the dataset against the quantiles of the theoretical distribution, the Q-Q plot reveals how well the data conforms to the expected distribution. A linear pattern suggests a good match, indicating that the dataset may be drawn from the theoretical distribution. Deviations from this linearity, such as a curved or S-shaped pattern, signal discrepancies in distribution characteristics like skewness or kurtosis.

After determining whether the data is normally distributed, we run the relevant test, which is either a paired t-test in case of normality or a Wilcoxon Signed-Rank test in case it is not normally distributed. In addition, we check the effect size using Cohen’s d coefficient. This process is repeated separately for accuracy, efficiency, and SUS, NASA-TLX and FSS questionnaires. It is done twice: first, comparing VR results to Desktop results and, secondly, comparing selection results to combined selection test results.

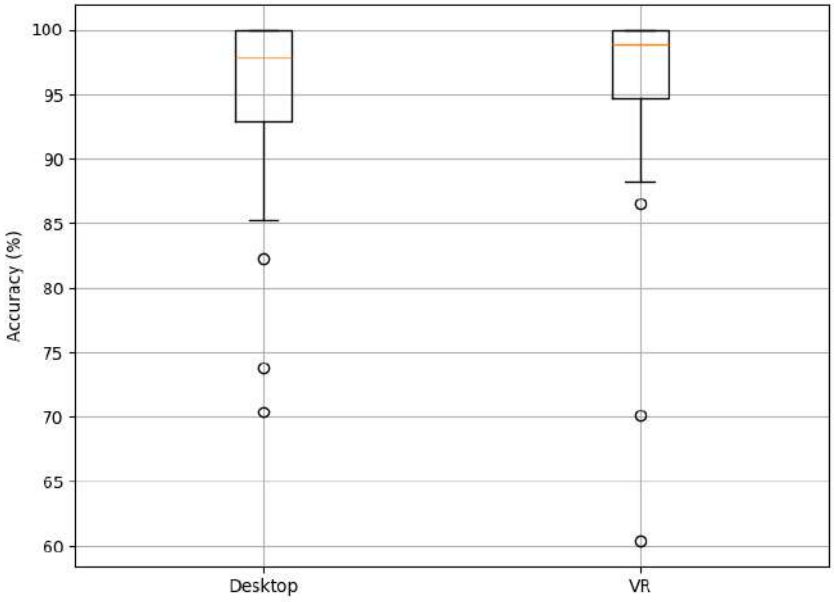
When we compare VR and Desktop, we have the same sample size for both, 28, as all participants tested both. When we compare selection and combined selection tests, our sample size differs; 15 participants and 13 participants, respectively. In this case, we run the Mann-Whitney U Test, suited for different sample sizes. If the p-value of the tests (Wilcoxon Signed-Rank or Mann-Whitney U) is smaller than 0.05 (the common 5% alpha), we can reject the null hypothesis in question (either H0 or H2) in favour of the alternate hypothesis (either H1 or H3). Otherwise, we accept the null hypothesis in question. The effect size indicates how much the difference between the tests is significant. An effect size d = 0.2 is small, d = 0.5 is medium, and d = 0.8 is large. A positive value of Cohen's d indicates that the mean of the first group is higher than the mean of the second group. A negative value indicates that the mean of the first group is lower than the mean of the second group. Lastly, we also compare the means, medians and standard deviation (SD) between the groups. All plots Y-axis do *not* start at 0, to allow an in-depth inspection of the graph without losing resolution.

6.2. Accuracy

The summarised accuracy results for the first research question are shown in Tbl. 8. The effect size is considered very small, with VR having a slightly better but negligible mean. The Wilcoxon Signed-Rank p-value is significantly higher than 0.05; therefore, we accept H0 regarding accuracy. The histograms and the Q-Q plots are in Appendix G. In addition, the median and SD are compared in Tbl. 9. The acceptance of H0 is supported by the results shown, as there is little to no difference between the accuracy scores, which is supported by the very small effect size.

	Desktop	VR
Mean	94.62	94.90
SD	7.73	9.12
Is the histogram normally distributed?	No	No
Shapiro-Wilk p-value	0.00	0.00
D'Agostino's K-squared p-value	0.00	0.00
Is the Q-Q plot showing normality?	No	No
Is normally distributed?	No	No
Wilcoxon Signed-Rank p-value	**0.449**	
Cohen's d effect size	**-0.032**	

Tbl. 8. Accuracy results of Desktop and VR, the better mean is marked in green

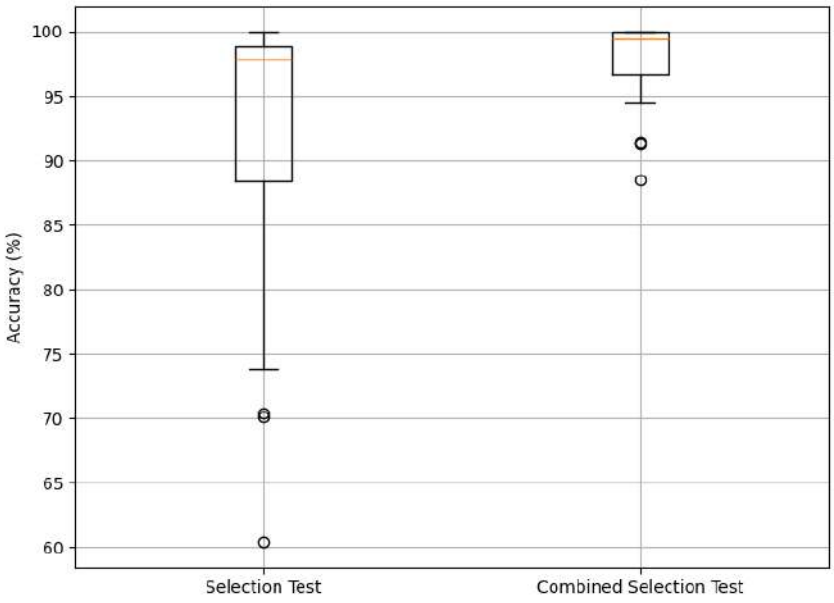


Tbl. 9. Comparison of accuracy between VR and Desktop

The summarised accuracy results for the second research question are shown in Tbl. 10. The effect size is considered big, with the combined selection test having a better mean. The Mann-Whitney U Test p-value is lower than 0.05; therefore, we reject H2 and accept H3 regarding accuracy. The histograms and the Q-Q plots are in Appendix G. In addition, the median and SD are compared in Tbl. 11. The rejection of H2 in favour of H3 agrees with the results, as the difference between the means is roughly 6%, which is supported by the big effect size, which is significant enough to show a difference between the tests.

	Selection Test	Combined Selection Test
Mean	92.13	97.8
SD	10.5	3.0
Is the histogram normally distributed?	No	No
Shapiro-Wilk p-value	0.00	0.00
D'Agostino's K-squared p-value	0.001	0.001
Is the Q-Q plot showing normality?	No	No
Is normally distributed?	No	No
Mann-Whitney U Test p-value	**0.01**	
Cohen's d effect size	**-0.69**	

Tbl. 10. Accuracy results of selection and combined selection tests, the better mean is marked in green



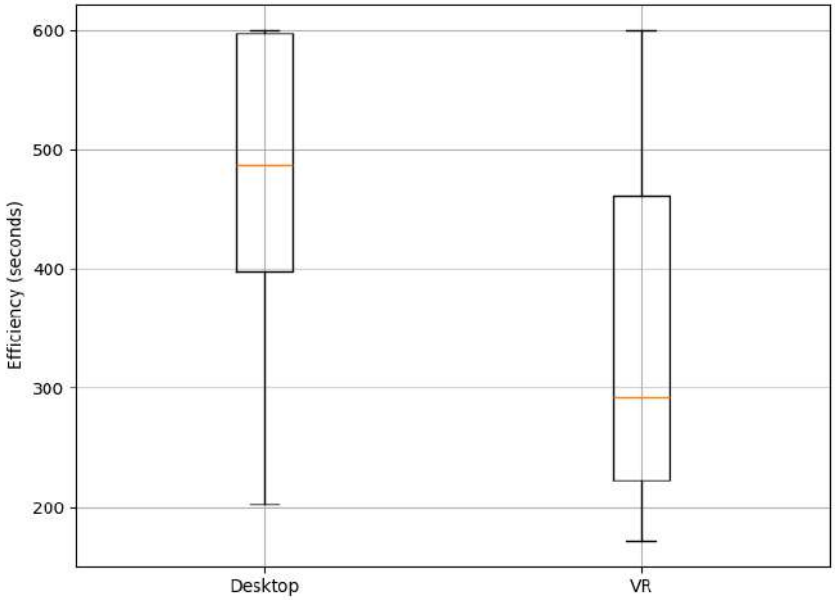
Tbl. 11. Comparison of accuracy between selection and combined selection tests

6.3. Efficiency

The summarised efficiency results for the first research question are shown in Tbl. 12. Unlike accuracy, a smaller mean means better results, as the mean is in seconds. The effect size is considered very large, with VR having a much smaller mean; hence, VR scored much better. The Wilcoxon Signed-Rank p-value is significantly smaller than 0.05; therefore, we reject H0 in favour of H1 regarding efficiency. The histograms and the Q-Q plots are in Appendix H. In addition, the median and SD are compared in Tbl. 13.

	Desktop	VR
Mean	476.75	343
SD	114.57	139.91
Is the histogram normally distributed?	No	No
Shapiro-Wilk p-value	0.01	0.014
D'Agostino's K-squared p-value	0.282	0.110
Is the Q-Q plot showing normality?	No	No
Is normally distributed?	No	No
Wilcoxon Signed-Rank p-value	**0.001**	
Cohen's d effect size	**1.02**	

Tbl. 12. Efficiency results of Desktop and VR, the better mean is marked in green

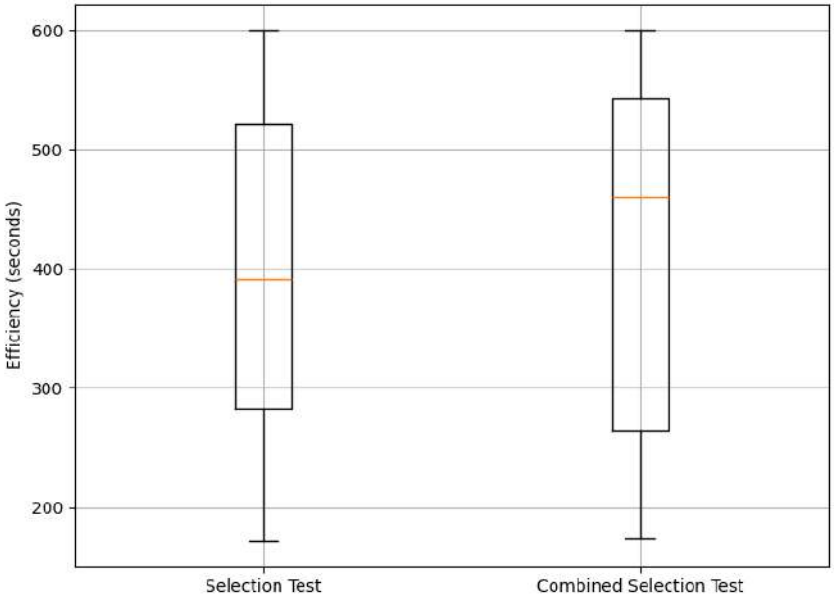


Tbl. 13. Comparison of efficiency between VR and Desktop. The lower median is faster, hence better.

The summarised efficiency results for the second research question are shown in Tbl. 14. As before, a smaller mean (seconds) means a better score, as the test was faster and more efficient. The effect size is small, with the selection test having a better mean. The Mann-Whitney U Test p-value is higher than 0.05; therefore, we accept H2 regarding efficiency. The histograms and the Q-Q plots are in Appendix H. In addition, the median and SD are compared in Tbl. 15.

	Selection Test	Combined Selection Test
Mean	397.01	424.72
SD	143.03	144.33
Is the histogram normally distributed?	No	No
Shapiro-Wilk p-value	0.031	0.011
D'Agostino's K-squared p-value	0.057	0.021
Is the Q-Q plot showing normality?	No	No
Is normally distributed?	No	No
Mann-Whitney U Test p-value	**0.37**	
Cohen's d effect size	**-0.1**	

Tbl. 14. Efficiency results of selection and combined selection tests, the better mean is marked in green



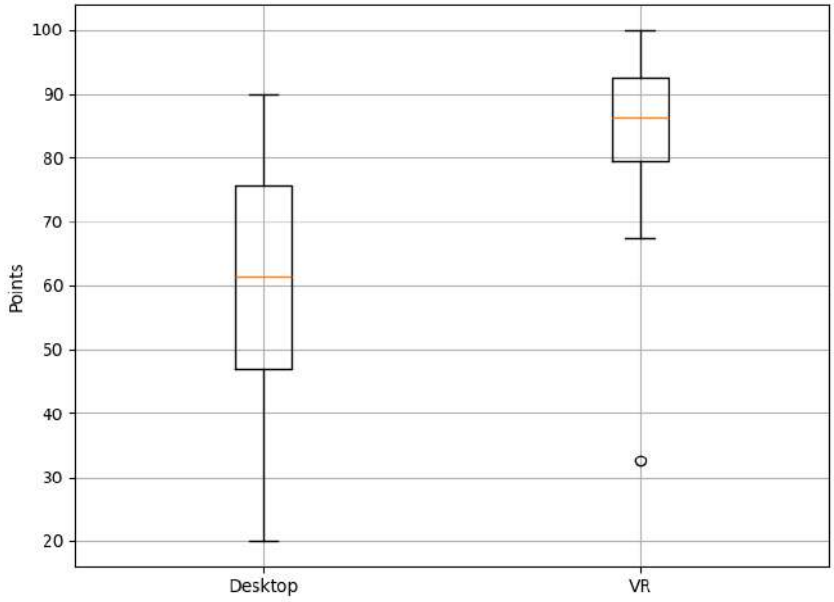
Tbl. 15. Comparison of efficiency between combined selection tests in desktop

6.4. System Usability Scale (SUS)

The summarised SUS results for the first research question are shown in Tbl. 16. The median and SD are compared in Tbl. 17. The desktop data is distributed normally, but the VR is not. Therefore, we used the Wilcoxon Signed Rank test. The effect size is very big, with VR having a much bigger, and hence, better score; Users found VR far more usable than Desktop. The Wilcoxon Signed-Rank p-value is 0.00; therefore, we reject H0 in favour of H1 regarding usability. The histograms and the Q-Q plots are in Appendix I.

	Desktop	VR
Mean	61.34	84.73
SD	19.83	13.52
Is the histogram normally distributed?	Yes	No
Shapiro-Wilk p-value	0.309	0.00
D'Agostino's K-squared p-value	0.395	0.00
Is the Q-Q plot showing normality?	Yes	No
Is normally distributed?	Yes	No
Wilcoxon Signed-Rank p-value	**0.00**	
Cohen's d effect size	**-1.35**	

Tbl. 16. SUS results of Desktop and VR, the better mean is marked in green

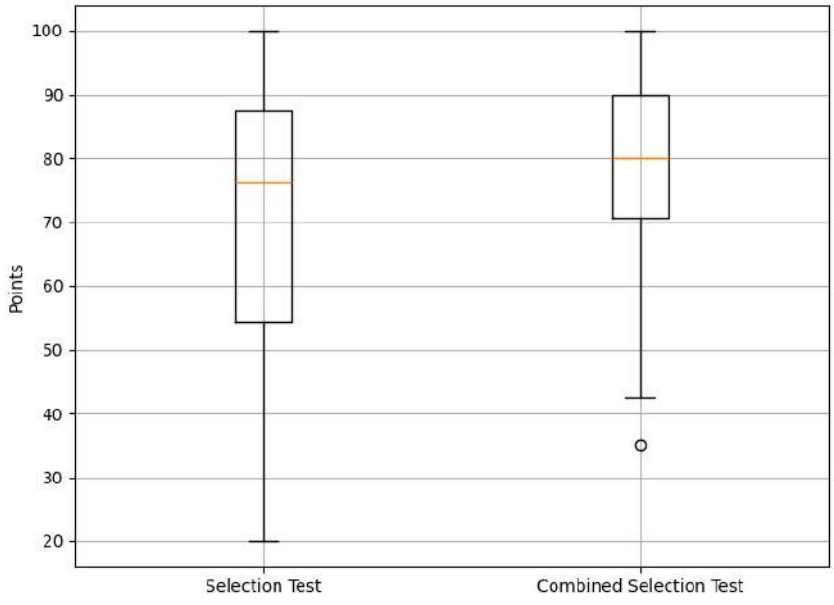


Tbl. 17. Comparison of SUS between VR and Desktop

The summarised SUS results for the second research question are shown in Tbl. 18. The effect size is small, with the selection test having a better mean. Although D'Agostino's K-squared resulted in $p > 0.05$, we give the Shapiro-Wilk test more weight and decide that neither test is normally distributed. The Mann-Whitney U Test p-value is higher than 0.05; therefore, we accept H2 regarding usability. The histograms and the Q-Q plots are in Appendix I. In addition, the median and SD are compared in Tbl. 19.

	Selection Test	Combined Selection Test
Mean	70.0	76.54
SD	22.87	16.97
Is the histogram normally distributed?	No	No
Shapiro-Wilk p-value	0.033	0.042
D'Agostino's K-squared p-value	0.215	0.118
Is the Q-Q plot showing normality?	No	No
Is normally distributed?	No	No
Mann-Whitney U Test p-value	**0.415**	
Cohen's d effect size	**-0.31**	

Tbl. 18. Efficiency results of selection and combined selection tests, the better mean is marked in green



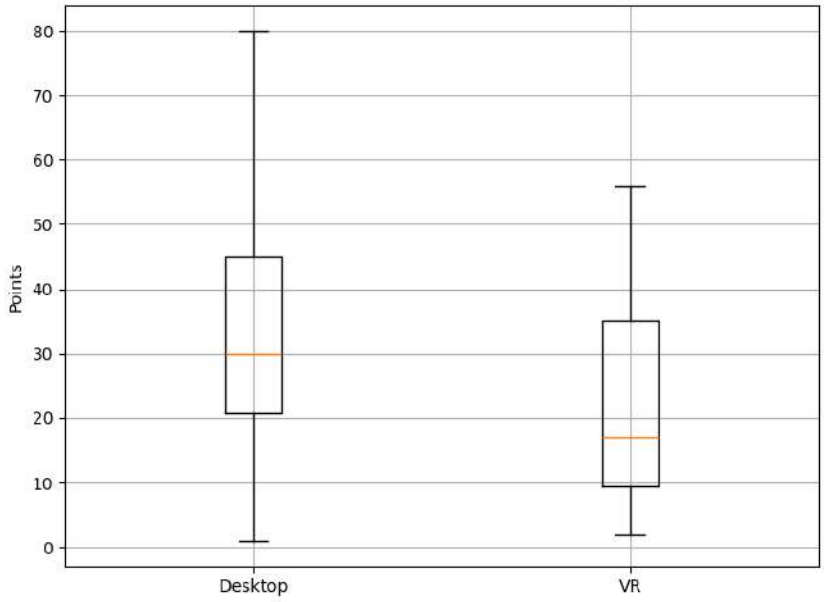
Tbl. 19. Comparison of SUS between combined selection tests in desktop

6.5. NASA-TLX

The summarised efficiency results for the first research question are shown in Tbl. 20. The mean and SD are compared in Tbl. 21. The desktop data is distributed normally, but the VR is not. Therefore, we used the Wilcoxon Signed Rank test. The effect size is big, with VR having a much bigger, and hence, better score; Users in VR felt less workload when using VR. In this questionnaire, a lower score (less workload) means a better score. The Wilcoxon Signed-Rank p-value is 0.01; therefore, we reject H0 in favour of H1 regarding workload. The histograms and the Q-Q plots are in Appendix J.

	Desktop	VR
Mean	33.61	21.57
SD	19.63	15.53
Is the histogram normally distributed?	Yes	No
Shapiro-Wilk p-value	0.669	0.035
D'Agostino's K-squared p-value	0.644	0.157
Is the Q-Q plot showing normality?	Yes	No
Is normally distributed?	Yes	No
Wilcoxon Signed-Rank p-value	**0.010**	
Cohen's d effect size	**0.667**	

Tbl. 20. NASA-TLX results of Desktop and VR, the better mean is marked in green

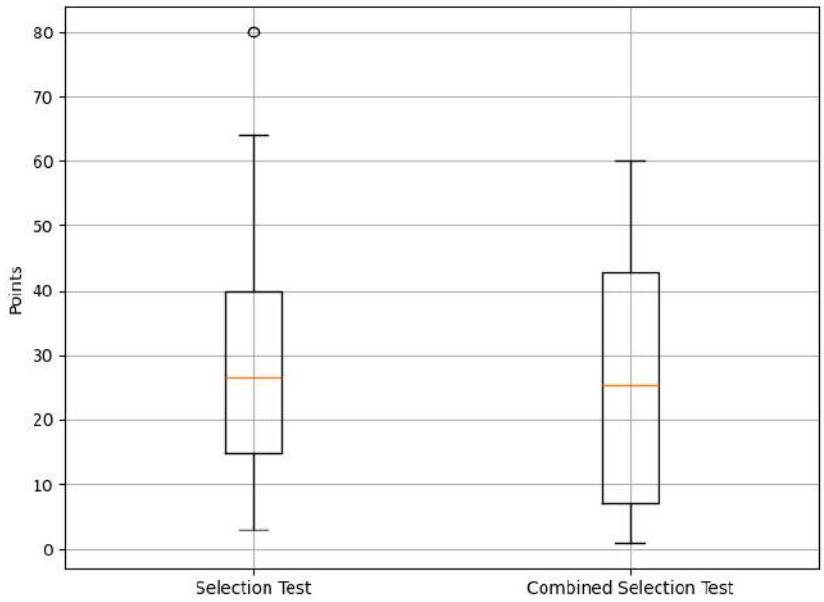


Tbl. 21. Comparison of NASA-TLX between Desktop and VR

The summarised NASA-TLX results for the second research question are shown in Tbl. 22. While the selection test is normally distributed, the combined selection is not. The effect size is small, with the selection test having a better mean. The Mann-Whitney U Test p-value is higher than 0.05; therefore, we accept H2 regarding workload. The histograms and the Q-Q plots are in Appendix J. In addition, the median and SD are compared in Tbl. 23.

	Selection Test	Combined Selection Test
Mean	29.43	25.46
SD	18.50	18.68
Is the histogram normally distributed?	Unclear	Unclear
Shapiro-Wilk p-value	0.105	0.031
D'Agostino's K-squared p-value	0.106	0.027
Is the Q-Q plot showing normality?	Yes	No
Is normally distributed?	Yes	No
Mann-Whitney U Test p-value	**0.450**	
Cohen's d effect size	**0.20**	

Tbl. 22. NASA-TLX results of selection and combined selection tests, the better mean is marked in green



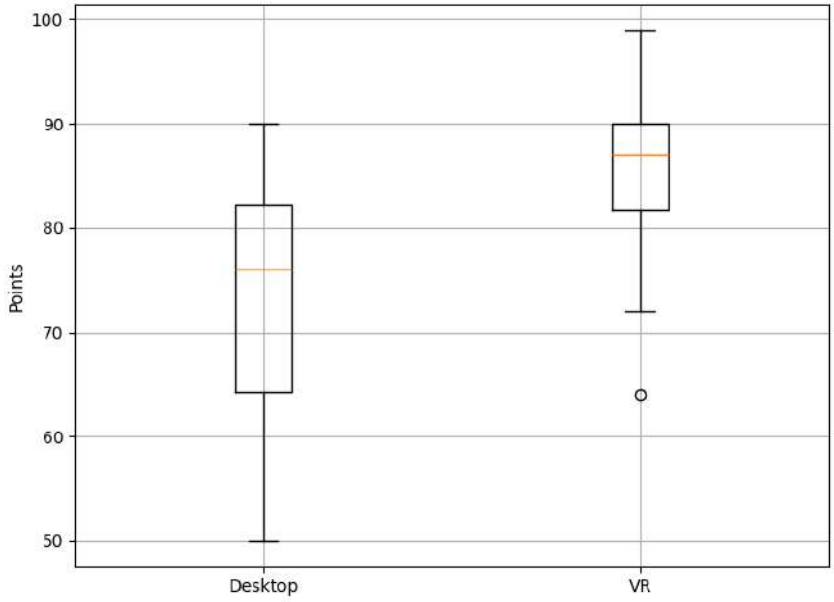
Tbl. 23. Comparison of NASA-TLX between combined selection tests in desktop

6.6. Flow State Scale (FSS)

The summarised FSS results for the first research question are shown in Tbl. 24. The mean and SD are compared in Tbl. 25. The VR data is distributed normally, but the desktop is not. Therefore, we used the Wilcoxon Signed Rank test. The effect size is very big, with VR having a much bigger and better score. The Wilcoxon Signed-Rank p-value is 0.00; therefore, we reject H0 in favour of H1 regarding flow. The histograms and the Q-Q plots are in Appendix K, alongside an in-depth analysis of the means between VR and Desktop on the specific dimensions of the FSS questionnaire.

	Desktop	VR
Mean	72.82	85.57
SD	12.36	7.48
Is the histogram normally distributed?	Unclear	Yes
Shapiro-Wilk p-value	0.037	0.123
D'Agostino's K-squared p-value	0.191	0.043
Is the Q-Q plot showing normality?	No	Yes
Is normally distributed?	No	Yes
Wilcoxon Signed-Rank p-value	**0.00**	
Cohen's d effect size	**-1.22**	

Tbl. 24. FSS results of Desktop and VR, the better mean is marked in green

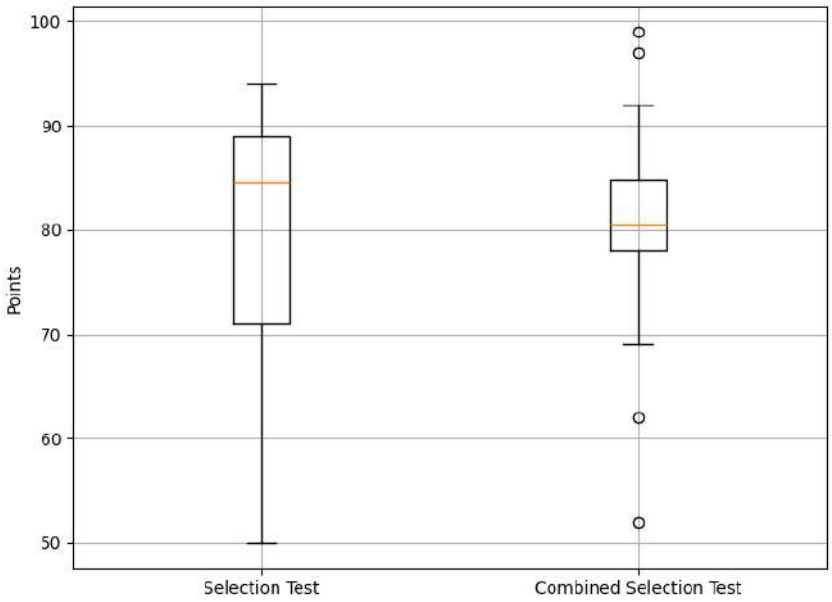


Tbl. 25. Comparison of FSS between VR and Desktop

The summarised FSS results for the second research question are shown in Tbl. 26. While the combined selection test is normally distributed, the selection test is not. The effect size is small, with the selection test having a better mean. The Mann-Whitney U Test p-value is higher than 0.05; therefore, we accept H2 regarding flow. The histograms and the Q-Q plots are in Appendix K. In addition, the median and SD are compared in Tbl. 27.

	Selection Test	Combined Selection Test
Mean	78.03	80.54
SD	13.47	9.96
Is the histogram normally distributed?	No	Yes
Shapiro-Wilk p-value	0.00	0.211
D'Agostino's K-squared p-value	0.097	0.075
Is the Q-Q plot showing normality?	No	Yes
Is normally distributed?	No	Yes
Mann-Whitney U Test p-value	**0.902**	
Cohen's d effect size	**-0.2**	

Tbl. 26. FSS results of selection and combined selection tests, the better mean is marked in green



Tbl. 27. Comparison of FSS between combined selection tests in desktop

6.7. Discussion

In this section, we discuss the results of the first research question and the second one, as both questions are unrelated. To answer the first research question from Chapter 1.2, we aim to reject the null hypothesis H0. By rejecting it, we accept the alternative hypothesis H1, which states that there is a significant difference across all metrics between the systems. To reject H0, we must reject all aspects, including the questionnaire scores, efficiency and accuracy. We perform an analysis of the metrics, as shown in Tbl. 28.

	Null-Hypothesis H0	VR Improvement over Desktop (Mean)
Accuracy	Accepted	1%
Efficiency	Rejected	28%
Usability	Rejected	38%
Workload	Rejected	55%
Flow	Rejected	17%

Tbl. 28. Comparison of results across all metrics for VR and Desktop

The average accuracy for both VR and desktop is comparable, each achieving around 94%. For desktop usage, the smaller standard deviation indicates a closer grouping of participants' results around the mean. This variability in standard deviation is attributed to the mixed experience levels with VR among participants, influencing the range of results. In contrast, desktop users typically exhibit more uniform experience levels, leading to less variation. With the negligible accuracy difference, confirmed by the Wilcoxon Signed-Rank Test with a p-value of 0.0 and effect size of -0.032, our findings suggest acceptance of the null hypothesis, indicating no significant difference between VR and desktop in terms of accuracy. However, efficiency and user experience metrics reveal a different narrative.

Efficiency comparisons, highlighted through histograms, show VR's superior speed and effectiveness over desktop. On average, VR users completed tasks 133 seconds faster than desktop tasks, with mean times of 343 seconds for VR and 476.75 seconds for desktop. Although VR again shows slightly greater variability (with a standard deviation of 139.91 versus 114.57 on desktop), the significance of these results leads us to reject the null hypothesis in favour of the alternative, based on a Wilcoxon Signed-Rank Test result with a p-value smaller than 0.05 and effect size of 1.02. Alongside accuracy, these metrics illustrate that VR maintains comparable accuracy to desktop and enhances test completion speed by an average of 28%, which any researcher would want when working with data. User experience, assessed through usability, workload, and flow metrics, further strengthens the case for VR. VR outperforms desktop across all aspects, with usability scores of 84.73 versus 61.34 on desktop — a 38% improvement. Workload scores also favour VR, showing a 55% reduction compared to desktop, and flow state scores are 17% higher in VR. Every flow individual dimension also outperformed desktop (see Appendix K). These improvements, verified

by the Wilcoxon Signed-Rank Test with p-values smaller than 0.05 and effect sizes bigger than 0.6, underscore the enhanced efficiency and user experience in VR despite many participants' lack of prior VR experience. Thus, in answering the first research question regarding the comparative effectiveness of particle selection in VR versus desktop, we reject the null hypothesis in favour of the alternative hypothesis, even though the accuracy was accepted in regards to H0. We reject the hypothesis regardless, as the evidence of the difference in efficiency and user experience cannot be ignored. As we aggregate the results, we have a clear picture that the VR system has significant differences and improvements compared to the desktop, highlighting the significant benefits of VR in efficiency and user experience while maintaining the same accuracy.

This result is interesting, especially concerning how conclusive it is. We would expect users who have no prior experience in VR to perform poorly on VR, especially compared to desktop. Seventeen of the participants were inexperienced with VR, which is 60% of the total participants. Despite their lack of VR experience, the results show that the selection task on VR was faster, more enjoyable, and usable and resulted in less workload. This disparity between the results and the participants' experience can be explained by looking at the task and the dataset itself.

Upon examining the task before us, specifically the selection of particles, it becomes evident that the nature of this task is inherently 3D. Selecting particles involves navigating through a space with depth, necessitating a capability for depth perception to identify a specific particle accurately. Desktop interfaces, however, are fundamentally limited in this regard; they lack the innate depth perception required for this task, namely binocular cues (see Chapter 3). Despite advancements in technology aimed at enhancing the selection process, the inherent limitations of a monitor prevent the realisation of genuine depth perception. In stark contrast, VR technology is designed with depth perception as a fundamental aspect. The experience of selecting a particle within a VR environment feels intuitive and natural because the user is immersed in a space that genuinely possesses depth. This is not achieved through the illusion of depth cues on a monitor but through the binocular depth perception enabled by the dual screens of the HMD, similar to how the eyes function. This fundamental difference profoundly influences the particle selection task, enabling the user to engage in a 3D selection within an authentic 3D environment.

The structure of the dataset further supports this observation. The dataset presents a compilation of sources that collectively depict a 3D object—the universe. Each source, representing a galaxy, occupies a specific position in space, mirroring its true 3D coordinates within the cosmos. Rendering these sources and their positions on a 2D device, such as a monitor, results in a 2D image that suffers from a lack of depth, a limitation imposed by the use of artificial depth cues. Conversely, presenting these sources within a VR environment employs the 3D positions of the sources with significance; the data envelops the user, presented in its actual celestial coordinates. Representing a 3D dataset within a 3D environment and attempting to execute a 3D selection task aligns well. This direct 3D to 3D mapping proves to be more intuitive and effective, even for novice users, compared to the traditional 3D to 2D mapping. While many tasks within VR are essentially 2D tasks adapted for a 3D context (such as typing on a virtual keyboard), our scenario presents a

unique reversal of this trend. We are confronted with a genuinely 3D task that naturally excels in a 3D environment, and our challenge lies in adapting it to function effectively in a 2D context on a desktop. This inversion highlights the reasons behind the enhanced task efficiency and user experience stemming from our development. Furthermore, this highlights the importance of developing software for VR that can truly benefit from its 3D abilities rather than trying to adapt 2D solutions into VR.

To answer the second research question from Chapter 1.2, we aim to reject the null hypothesis H2. By rejecting it, we accept the alternative hypothesis H3, which states that there is a significant difference across all metrics between the tests. To reject H2, we must reject all aspects, including the questionnaire scores, efficiency and accuracy. We perform an analysis of the metrics, as shown in Tbl. 29.

	Null-Hypothesis H2	Combined Selection Test Improvement over Selection Test (Mean)
Accuracy	Rejected	6%
Efficiency	Accepted	-7%
Usability	Accepted	9%
Workload	Accepted	15%
Flow	Accepted	3%

Tbl. 29. Comparison of results across all metrics for the selection and combined selection tests

The results this time are *opposite* to the results from the VR/Desktop comparison. This time, the only metric that is rejected in regards to H2 is accuracy. Unlike H0, the rejection or acceptation of H2 is based on much more diverse results. From a user experience perspective, considering the three questionnaires and their result, H2 was accepted. But, looking at the mean, all three (usability, workload and flow) performed better. As expected, accuracy increased due to the parameter tuning. Of the 13 participants who participated in the combined selection test, 11 changed one parameter, and 2 changed two parameters. This makes the result much more surprising, as only two parameters at most were changed, which caused a non-negligible difference (therefore, accuracy was rejected regarding H2). On the other hand, efficiency decreased in an almost linear relationship to the increase in accuracy. This, too, is expected. When users change parameters, they also lose time to select particles. We hoped that the relationship would not be 1:1 and that the increase in accuracy would overshadow the decrease in efficiency, but this is *not* the case.

Based on our results, all metrics were accepted regarding H2, except accuracy. On the other hand, all metrics performed better regarding mean, except efficiency. These results are mixed, and we cannot reject H2 based on them. Therefore, the answer to the second research question is negative: there is *no* significant difference in the metrics when using parameter tuning.

These findings present deviation from anticipated outcome, predicated on the presumption that changing parameters would yield improvements in task execution, specifically enhancing accuracy and expediting selection processes, even when taking into consideration the time it takes to change a parameter. Contrary to expectations, such differences between the tests were not decisively evidenced within the results obtained. A factor potentially contributing to this may be attributed to the demographic composition of the participants; notably, 11 out of 13 participants engaged in the combined selection test were students, comprising approximately 84% of the sample with no prior familiarity with the dataset at hand.

The significance of context in this scenario cannot be underestimated, given that the modification of parameters requires prior knowledge of the dataset. Some parameter adjustments might facilitate faster particle selection than others. A majority of 11 participants chose to modify solely the particle size parameter, thereby enhancing the precision of particle cluster selection. Conversely, a minority of two participants changed both particle size and one spatial dimension. The strategic alteration of spatial dimensions, such as substituting the Y axis with velocity instead of spatial positioning, effectively restructures the dataset into a cone-shaped structure, positioning scattered particles at the base of the cone and clusters at the apex. Only researchers who have prior knowledge of this dataset have this nuanced understanding.

The empirical evidence does indeed indicate a small of improvement in selection accuracy, aligning with theoretical expectations. However, this is matched by a similar decline in efficiency, which is unexpected. Still, improvements in usability, workload, and flow observed in the combined selection test—signal a positive trajectory, especially with the demographics of the test. Even minimalistic parameter changes, such as adjustments to particle size, have been demonstrated to improve the user experience.

While these findings do not conclusively validate the initial hypothesis nor fully address the posed research question, they underscore the importance of sharing domain-specific knowledge in contexts where task performance depends upon such understanding. Moreover, the results advocate for the development of adaptable software interfaces, wherein even marginal adjustments, as in our case of particle size alteration, can yield improvements.

Chapter 7 - Conclusions

In this chapter, we analyse the findings presented in the preceding chapter, juxtaposing our interpretations and the empirical outcomes against this thesis's established hypotheses and research inquiries. Furthermore, we explore further software implementations, allowing scalability.

7.1. Results and Interpretation

The experiment conducted highlights a comparison between VR and desktop interfaces in terms of accuracy, efficiency, and user experience in performing selection tasks. With an average accuracy rate of approximately 94% for both VR and desktop, the findings suggest no significant difference in accuracy between the two. However, the desktop interface demonstrated a smaller standard deviation, indicating a more consistent performance among participants, likely due to their more uniform experience levels with desktop usage compared to the mixed experience levels with VR. Efficiency metrics, however, revealed a distinct advantage for VR, with VR users completing tasks an average of 133 seconds faster than their desktop counterparts. This significant difference in task completion speed, supported by statistical analysis (p-value < 0.05, effect size 1.02), suggests that VR not only maintains comparable accuracy to desktop interfaces but also enhances efficiency by 28%. This finding leads to the rejection of the null hypothesis in favour of VR's superiority in efficiency.

User experience metrics further bolster the case for VR, showing significant improvements across usability, workload, and flow states compared to the desktop. VR outperformed desktop in usability by 38%, reduced workload by 55%, and increased flow state scores by 17%, with statistical tests confirming these improvements as significant. These results underscore VR's potential to offer a more engaging and less taxing experience despite participants' varying levels of familiarity with VR technology. This is a surprising outcome, particularly due to the fact that the experiment was conducted mostly by participants inexperienced with VR. Despite 60% of the participants lacking prior VR experience, the findings indicate that the task was completed more swiftly, was more enjoyable, and was deemed more usable in VR, also resulting in a reduced workload for the users. This contrast between expected outcomes based on user experience and the actual results suggests that the nature of the task and the characteristics of the dataset used might offer explanations for this outcome. Upon closer examination, the essence of the selection task—which involved selecting particles in a three-dimensional space—reveals why VR provided a superior platform for task execution. Desktop interfaces, by their very nature, are constrained by their inability to provide genuine depth perception, a critical requirement for accurately navigating and selecting particles in a 3D space. VR technology, on the other hand, is designed at its core to simulate depth perception, mimicking the natural function of the human eyes through binocular depth cues provided by the HMD. This fundamental difference in how depth is presented and perceived accounts for the intuitive and efficient interaction with the 3D environment in VR, making the particle selection task more natural and effective compared to the traditional 2D display limitations of desktop interfaces. Furthermore, the structure of the dataset, which represents a compilation of galaxies, each

occupying a unique position in a model of the universe, underscores the advantages of VR for this specific task. While rendering this inherently 3D data on a 2D monitor results in a significant loss of depth perception even with the help of artificial depth cues, presenting it in VR allows for a direct mapping of 3D data in a 3D space. This enables users to interact with the dataset as it exists in reality, enveloping them in a true representation of celestial coordinates. This direct 3D-to-3D mapping enhances the intuitiveness and effectiveness of the selection task for even novice users, demonstrating the potential of VR technology to significantly improve the efficiency and user experience of inherently 3D tasks. This contrast not only highlights the limitations of adapting 3D tasks to 2D interfaces, such as virtual keyboards in VR, but also emphasises the importance of developing VR applications that leverage the technology's 3D capabilities to their fullest extent using 3D tasks.

In addressing a second research question regarding the impact of parameter tuning on task performance, the study presents mixed results. While some metrics improved with parameter adjustments, notably accuracy (+6%), efficiency suffered a proportional decline (-7%). This outcome suggests that contrary to expectations, parameter tuning did not enhance task execution. Most participants opted to adjust parameters like particle size, affecting the task performance in nuanced ways. An interesting aspect of the experiment that can explain these results was the demographic composition of the participants, where a majority were students with no prior experience with the dataset, which could have influenced the outcomes. The assumption was that parameter tuning could lead to more efficient task execution, as the task should become easier to finish, yet this was not strongly evidenced in the results.

The experiment highlights the importance of domain-specific knowledge for context-based tasks, especially when it comes to adjusting parameters like particle size or spatial dimensions. Most participants opted to alter only the particle size, which was thought to enhance precision in selecting particle clusters. A smaller group experimented with changing both particle size and spatial dimensions, a strategy that shows a deeper understanding of the dataset structure. This approach suggests that having prior knowledge of the parameters could potentially allow for better selection in the dataset.

Despite the anticipated improvements in accuracy and efficiency not being fully realized, the experiment did observe some positive outcomes regarding user experience. Improvements in usability (+9%), workload (+15%), and flow (+3%) were noted. These findings suggest that even without having prior knowledge about the parameters, a small change, such as particle size, affected the overall satisfaction with the task, and especially affected the workload, effectively reducing the workload and improving the usability. This aligns well with how a smaller particle size allowed an easier, hence less workload, in selecting particles within clusters.

7.2. Future Work & Improvements

This research is, by design, limited in duration and scope, necessitating the postponement or complete abandonment of certain implementations and ideas. These concepts, which were not pursued within the scope of this timeframe, are listed in this section.

Avoid looping the data and better search algorithms: our VR and Desktop implementations currently face a significant limitation due to the inefficient method of generating a boolean operation selection sphere, which involves iterating over each particle in the cache to check its inclusion within the sphere. This approach was initially adequate for smaller datasets, like the 2MRS Group Galaxies, but struggles with the scale of datasets containing billions of particles, especially when spheres are created frequently. Adopting spatial partitioning algorithms, such as the k-d tree algorithm, could improve efficiency and performance by reducing the amount of particles considered in the search (Bentley, 1975).

Explore other interfaces for parameter fine-tuning: we have successfully integrated both Speech and Manual Command Interfaces into our system, with room for further enhancements. The Eye Movement Location Interface (ELI) is a notable addition, leveraging Oculus Quest 3's sensors for precise eye movement tracking (this thesis used Oculus Quest 2, which does not support ELI). This allows users to select parameters hands-free, improving efficiency and enabling multitasking with navigation. Incorporating ELI could potentially boost efficiency (Tanriverdi & Jacob, 2000).

Explore different control layouts: the controllers' control layout was designed around the pre-existing locomotion system's need for both grip triggers, guided by "common sense" and industry best practices without analytical comparison. Our chosen layout may not have been optimal, potentially influencing our findings. Investigating' more effective control schemes could lead to more definitive results and insights.

Modify render code for iDaVIE-p desktop version to support GPU rendering: the desktop version of iDaVIE-p struggles with scalability due to its method of rendering each particle as a GameObject, a technique that works for smaller datasets but falters with millions or billions of particles because of the heavy demand on both CPU and GPU, with the CPU bottleneck becoming a significant issue. In contrast, the VR version uses GPU-based rendering. Moving to GPU rendering for the desktop could enhance performance.

Add support for spherical coordinate systems: the current version of iDaVIE-p only accommodates cartesian coordinate systems, excluding datasets in spherical coordinates. Minor adjustments in particle position calculations and selection sphere within the GPU shaders are needed to support spherical coordinates.

Acknowledgement

This work made use of the Inter-University Institute for Data Intensive Astronomy (IDIA) visualisation lab https://vislab.idia.ac.za. IDIA is a partnership of the University of Cape Town, the University of Pretoria and the University of the Western Cape.

This work made use of the iDaVIE-p (immersive Data Visualisation Interactive Explorer for Particle Rendering) software (DOI – 10.5281/zenodo.4614116 – https://idavie.readthedocs.io).

References

Abbott, B. P., Emmart, C. B., Levy, S., & Liu, C. T. (2004). Visualizing and Analyzing Massive Astronomical Datasets with Partiview. *ESO ASTROPHYSICS SYMPOSIA*, pp. 57-61.

Arcand, K., Price, S., & Watzke, M. (2020). Holding the Cosmos in Your Hand: Developing 3D Modeling and Printing Pipelines for Communications and Research. *Frontiers in Earth Science*, *8*(590295), pp. 1-8.

B.H. McCormick, T. A. D., M.D. Brown. (1987). Visualization in scientific computing. *Computer Graphics and Image Processing*, *21*, pp. 1-14.

Bailly, G., & Malacria, S. (2020). Command Selection. *Handbook of Human Computer Interaction*, pp. 1-35.

Baloup, M., Pietrzak, T., & Casiez, G. (2019). RayCursor: A 3D Pointing Facilitation Technique based on Raycasting. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, *101*, pp. 1-12.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Commun. ACM*, *18*(9), pp. 509–517.

Blattgerste, J., Renner, P., & Pfeiffer, T. (2018). Advantages of eye-gaze over head-gaze-based selection in virtual and augmented reality under varying field of views.

Bowman, D., & Wingrave, C. (2001). Design and evaluation of menu systems for immersive virtual environments. *Proceedings - Virtual Reality Annual International Symposium*, pp. 149-156.

Bowman, D. A. a. R. P. M. (2007). Virtual Reality: How Much Immersion Is Enough? *Computer 40.7*, pp. 36-43.

Brooke, J. (1996). SUS - A quick and dirty usability scale. *Usability Evaluation in Industry.*, pp. 189-194.

Csikszentmihalyi, M. (1990). Flow: The Psychology of Optimal Experience. pp. 1-8.

de Sitter, W. (1934). On distance, magnitude, and related quantities in an expanding universe. *Bulletin of the Astronomical Institutes of the Netherlands*, *7*(261), pp. 206-216.

Delgado, F. J., & Noyes, M. (2017). NASA's Hybrid Reality Lab: One Giant Leap for Full Dive. *GPU Technology Conference (GTC)*, pp. 1-54.

Dhamodharavadhani, s., Rajasekaran, G., & Ramalingam, R. (2018). Unlock Different V's of Big Data for Analytics. *First International Conference on Computer Vision,Networks and Informatics*, pp. 183-190.

Djorgovski, S., Hut, P., Knop, R., Longo, G., McMillan, S., Vesperini, E., Donalek, C., Graham, M., Mahabal, A., Sauer, F., White, C., Lopes, C. (2013). The MICA experiment: Astrophysics in virtual worlds. *International Research Conference on Virtual Worlds*, pp. 49-58.

Dosher, B. A., Sperling, G., & Wurst, S. A. (1986). Tradeoffs between stereopsis and proximity luminance covariance as determinants of perceived 3D structure. *Vision Research*, *26*(6), pp. 973-990.

Evans, I., Primini, F., Glotfelty, K., Anderson, C., Bonaventura, N., Chen, J., Davis, J., Doe, S., Evans, J., Fabbiano, G., Galle, E., Gibbs, D., Grier, J., Hain, R., Hall, D., Harbo, P., He, X., Houck, J., Karovska, M., & Zografou, P. (2010). The Chandra Source Catalog. *The Astrophysical Journal Supplement Series*, *189*, pp. 38.

Feigelson, E. (2012). Classification in Astronomy: Past and Present. *Advances in Machine Learning and Data Mining for Astronomy*, pp. 3-10.

Ferrand, G., English, J., Irani, P. (2016). 3D visualization of astronomy data cubes using immersive displays. *Canadian Astronomical Society Conference*, pp. 1-8.

Fisher, S. K., & Cuiffreda, K. J. (1980). Adaptation to optically increased interocular separation under naturalistic viewing conditions. *Perception*, pp. 171-180.

Fluke, C., Barnes, D. (2018). Immersive virtual reality experiences for all-sky data. *Publications of the Astronomical Society of Australia*, *35*, pp. 1-11.

Friendly, M. (1995). Milestones in the history of thematic cartography, statistical graphics, and data visualization. *13th International Conference on Database and Expert Systems applications*, pp. 59-66.

Gain, J., & Bechmann, D. (2008). A survey of spatial deformation from a user-centered perspective. *ACM Trans. Graph.*, *27*, pp. 1-21.

Gibson, J. J. (1986). *The ecological approach to visual perception*. Houghton, Mifflin and Company.

Hansberger, J. T., Peng, C., Mathis, S. L., Areyur Shanthakumar, V., Meacham, S. C., Cao, L., & Blakely, V. R. (2017). Dispelling the Gorilla Arm Syndrome: The Viability of Prolonged Gesture Interactions. *International Conference on Virtual, Augmented and Mixed Reality*, pp. 505-520.

Harris, D., Arthur, T., Kearse, J., Olonilua, M., Hassan, E., Burgh, T., Wilson, M., & Vine, S. (2023). Exploring the role of virtual reality in military decision training. *Frontiers in Virtual Reality*, *4*, pp. 1-11.

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*, *52*, pp. 139-183.

Herschel, J. F. W. (1863). A General Catalogue of Nebulae and Clusters of Stars for the Year 1860.0, with Precessions for 1880.0. [Abstract]. *Proceedings of the Royal Society of London*, *13*, pp. 1-3. http://www.jstor.org/stable/111986

Huchra, J., Macri, L., Masters, K., Jarrett, T., Berlind, P., Calkins, M., Crook, A., Cutri, R., Erdogdu, P., Falco, E., George, T., Hutcheson, C., Lahav, O., Mader, J., Mink, J., Martimbeau, N., Schneider, S., Skrutskie, M., Tokarz, S., & Westover, M. (2011). The 2MASS Redshift Survey - Description and data release. *The Astrophysical Journal Supplement Series*, *199*, pp. 1-39. https://doi.org/10.1088/0067-0049/199/2/26

Jackson, S. A., & Marsh, H. W. (1996). Development and Validation of a Scale to Measure Optimal Experience: The Flow State Scale. *Journal of Sport and Exercise Psychology*, *18*(1), pp. 17-35.

Jacob, R. J. K. (1990). What you look at is what you get: eye movement-based interaction techniques. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 11–18.

Jarrett, T. (2004). Large Scale Structure in the Local Universe - The 2MASS Galaxy Catalog. *Publications of the Astronomical Society of Australia*, *21*, pp. 396-403.

Jarrett, T. H., Comrie, A., Marchetti, L., Sivitilli, A., Macfarlane, S., Vitello, F., Becciani, U., Taylor, A. R., van der Hulst, J. M., Serra, P., Katz, N., & Cluver, M. E. (2021). Exploring and interrogating astrophysical data in virtual reality. *Astronomy and Computing*, *37*, 100502.

Kaufman, A. e. a. (1994). Research issues in volume visualization. *IEEE Computer Graphics and Applications*, pp. 63-67.

Kavanagh, S., Luxton-Reilly, A., Wuensche, B., & Plimmer, B. (2017). A systematic review of Virtual Reality in education. *Themes in Science and Technology Education*, *10*(2), pp. 85-119.

Lambert, T., Kraan-Korteweg, R., Jarrett, T., & Macri, L. (2020). The 2MASS redshift survey galaxy group catalogue derived from a graph-theory based friends-of-friends algorithm. *Monthly Notices of the Royal Astronomical Society*, *497*, pp. 2954-2973.

Levy, S. (2003). Interactive 3-D visualization of particle systems with Partiview. *Astrophysical Supercomputing using Particle Simulations*, *208*, pp. 343-348.

Li, Y. (1997). Oriented particles for scientific visualization. *Master's thesis, University of New Brunswick, Canada*.

MacKenzie, I., Sellen, A., & Buxton, W. (1991). A comparison of input devices in element pointing and dragging tasks. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 161-166. https://doi.org/10.1145/108844.108868

Marchetti, L., Jarrett, T.H. (2018). The data2dome initiative at the Iziko planetarium & the idia visualisation lab. *BigSkyEarth Conference: AstroGeoInformatics.*, pp. 1-12.

Marchetti, L., Jarrett, T.H., Comrie, A., Sivitilli, A.K., Vitello, F., Becciani, U., Taylor, A. (2020). IDAVIE-V: immersive data visualisation interactive explorer for volumetric rendering. *arXiv:2012.11553*, pp. 1-4.

Mashey, J. R. (1999). Big Data and the Next Wave of InfraStress. *1999 USENIX Annual Technical Conference (USENIX ATC 99)*.

Meagher, D. (1980). *Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer*.

Metz, C. E. (1978). Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, *8*(4), pp. 283-298.

Mridha, M. F., Das, S. C., Kabir, M. M., Lima, A. A., Islam, M. R., & Watanobe, Y. (2021). Brain-Computer Interface: Advancement and Challenges. *Sensors*, *21*(17), pp. 5746.

NASA. (2020). Goddard's Emerging Technologies. *Cutting Edge*, *16*(2), pp. 2-3.

Pence, W., Chiappetti, L., & Shaw, R. (2010). Definition of the Flexible Image Transport System (FITS), version 3.0. *Astronomy & Astrophysics (A&A)*, *524*, pp. 1-40.

Pillai, A. S., & Mathew, P. S. (2019). Impact of Virtual Reality in Healthcare: A Review. *Virtual and Augmented Reality in Mental Health Treatment*, pp. 17-31.

Piumsomboon, T., Lee, G., Lindeman, R., & Billinghurst, M. (2017). Exploring natural eye-gaze-based interaction for immersive virtual reality.

Poole, A., & Ball, L. J. (2006). Encyclopedia of Human Computer Interaction. pp. 211-219.

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

Roth, S. D. (1982). Ray casting for modeling solids. *Computer Graphics and Image Processing*, *18*(2), pp. 109-144.

Sellen, A., Kurtenbach, G., & Buxton, W. (1992). The Prevention of Mode Errors Through Sensory Feedback. *Human-Computer Interaction*, *7*, pp. 141-164.

Shneiderman, B. (1983). Direct Manipulation: A Step Beyond Programming Languages. *Computer*, *16*, 57-69.

Skrutskie, M., Cutri, R., Stiening, R., Weinberg, M., Schneider, S., Carpenter, J., Beichman, C., M, C., Chester, T., Huchra, J., Liebert, J., Lonsdale, C., Monet, D., Price, S., Seitzer, P., Jarrett, T., Kirkpatrick, J., Gizis, J., & Wheelock, S. (2007). The two micron all sky survey (2MASS). *The Astronomical Journal*, *131*, pp. 1163.

Speicher, M., Feit, A., Ziegler, P., & Krüger, A. (2018). Selection-based Text Entry in Virtual Reality. *The 2018 CHI Conference*, pp. 1-13.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, *62*(1), pp. 77-89.

Steinicke, F., Ropinski, T., & Hinrichs, K. (2006). Object selection in virtual environments using an improved virtual pointer metaphor. *32*, pp. 320-326.

Sunyaev, R. A. (1974). The Thermal History of the Universe and the Spectrum of Relic Radiation. *Confrontation of Cosmological Theories with Observational Data*, pp. 167-173.

Sutherland, I. E. (1965). The ultimate display.

Sutherland, I. E. (1968). A head-mounted three dimensional display. *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pp. 757–764.

T.H. Jarrett, A. C., L. Marchetti, A. Sivitilli, S. Macfarlane, F. Vitello, U. Becciani, A.R. Taylor, J.M. van der Hulst, P. Serra, N. Katz, M.E. Cluver,. (2021). Exploring and interrogating astrophysical data in virtual reality. *Astronomy and Computing*, *37:100502*(5), pp. 1-23.

Tanriverdi, V., & Jacob, R. J. K. (2000). Interacting with eye movements in virtual environments. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 265–272.

Taylor, M. (2017). TOPCAT: Desktop Exploration of Tabular Data for Astronomy and Beyond. *Informatics*, *4*(3), pp. 18.

Thériault, L., Robert, J.-M., & Baron, L. (2004). Virtual Reality Interfaces for Virtual Environments. *6th Virtual Reality International Conference (IEEE VRIC 2004)*, pp. 1-6.

Viguier, A., Clément, G., & Trotter, Y. (2001). Distance Perception within near Visual Space. *Perception*, *30*, pp. 115-124.

Wang K.-S, C. A., Harris P., Moraghan A., Hsu S. -C, Pinska A., Chiang C.-C, Jan H., Simmonds R., Pang Q.,5 Brandt P., Taylor R., Ott J., Rosolowsky E., Lee C.-F, Raba R., Kirkham K. (2020). CARTA: Cube Analysis and Rendering Tool for Astronomy. *Astronomical Data Analysis Software and Systems XXIX*, *527*, pp. 1-213.

Ware, C. (2013). Information Visualization: Perception for Design: Third Edition. pp. 239-290.

Wells, D. C., Greisen, E. W., & Harten, R. H. (1981). FITS - a Flexible Image Transport System. *Astronomy and Astrophysics Supplement Series*, *44*, pp. 363.

Wheatstone, C. (1843). Contributions to the physiology of vision. On some remarkable and hitherto unobserved phenomena of binocular vision. *Abstracts of the Papers Printed in the Philosophical Transactions of the Royal Society of London*, *4*, pp. 76-77.

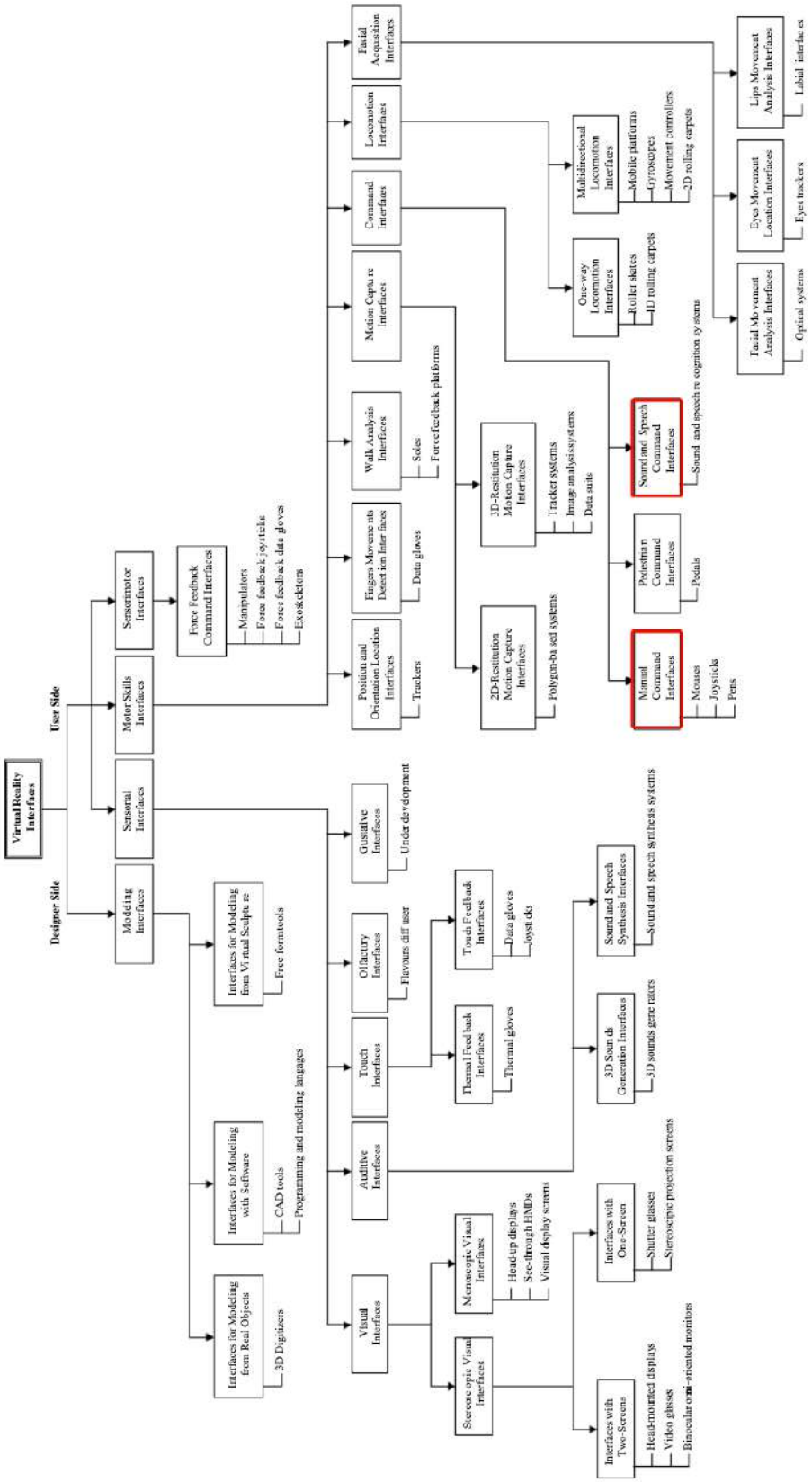
Wolfe, J. M., & Gancarz, G. G. (1996). Guided Search 3.0: A model of visual search. *Basic and clinical application of visual science*, pp. 189–192.

Wright, E. (2010). WISE, The Wide-field Infrared Survey Explorer: Status and Early Results. *American Astronomical Society, Bulletin of the American Astronomical Society*, *41*, pp. 854.

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. pp. 246-253.

Zheng, J. M., Chan, K. W., & Gibson, I. (1998). Virtual reality. *Potentials, IEEE*, *17*, pp. 20-23.

Appendix A: Interaction Interfaces For VR



Appendix B: Hardware Requirements

The hardware requirements for this development derive directly from the hardware requirements of iDaVIE, which can be found on the official website [18]. To operate the software successfully, a system with SteamVR compatibility is necessary. Detailed system requirements can be found on the SteamVR store page. We advise using an NVIDIA GTX 1070, a comparable AMD GPU, or a more advanced model for optimal performance.

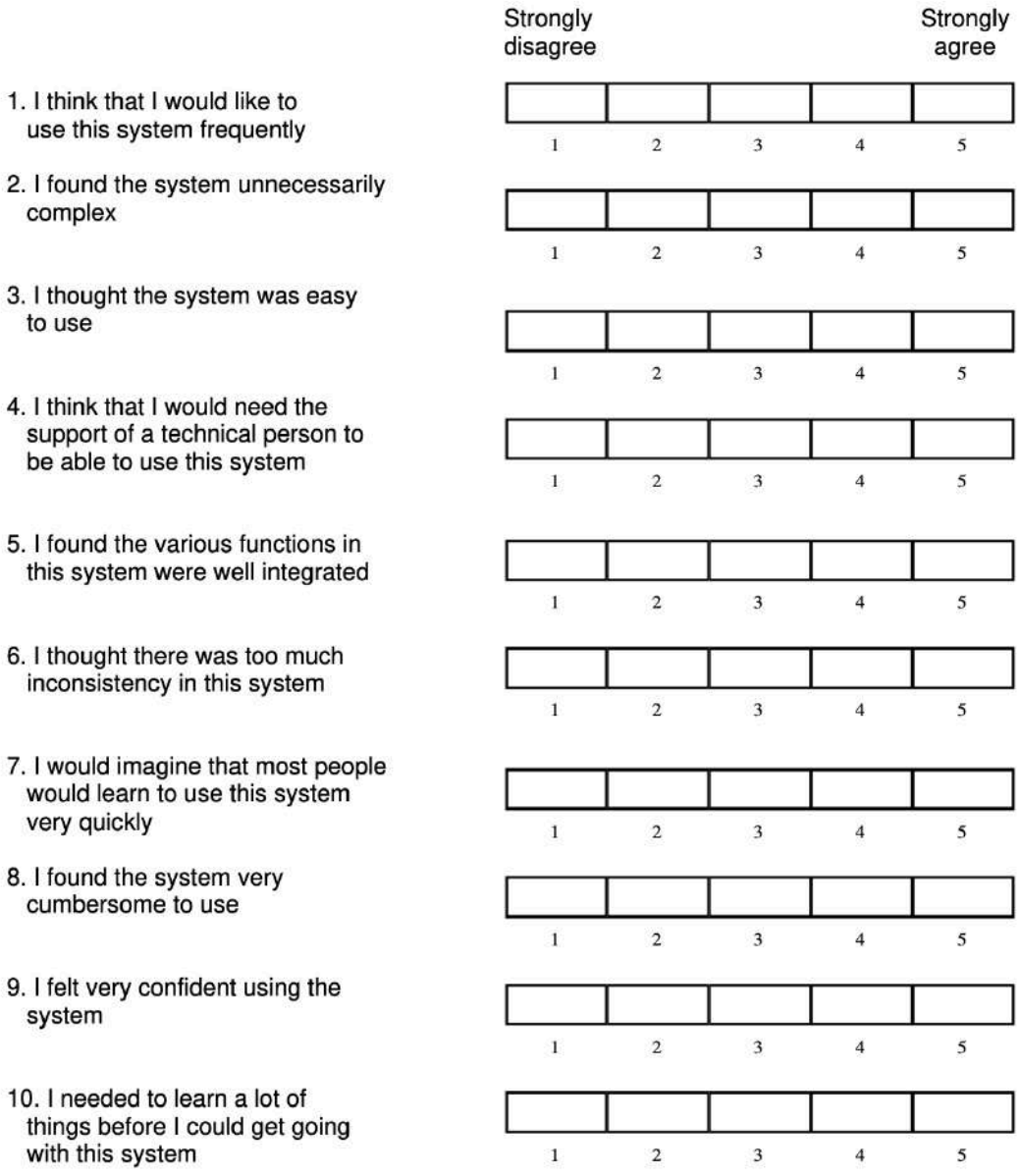
We suggest a CPU with at least four cores. A minimum of 8 GB of RAM is essential, but the available system memory significantly influences the manageable size of data cubes. For enhanced performance, especially with large data cubes, 16 GB is recommended, and for the best experience, 32 GB.

The software is designed to be compatible with any VR headset that supports SteamVR. We have conducted tests with the following VR headsets to ensure compatibility: Meta Rift, Meta Rift S, Meta Quest 2, HTC Vive, HTC Vive Pro, Valve Index and Samsung Odyssey

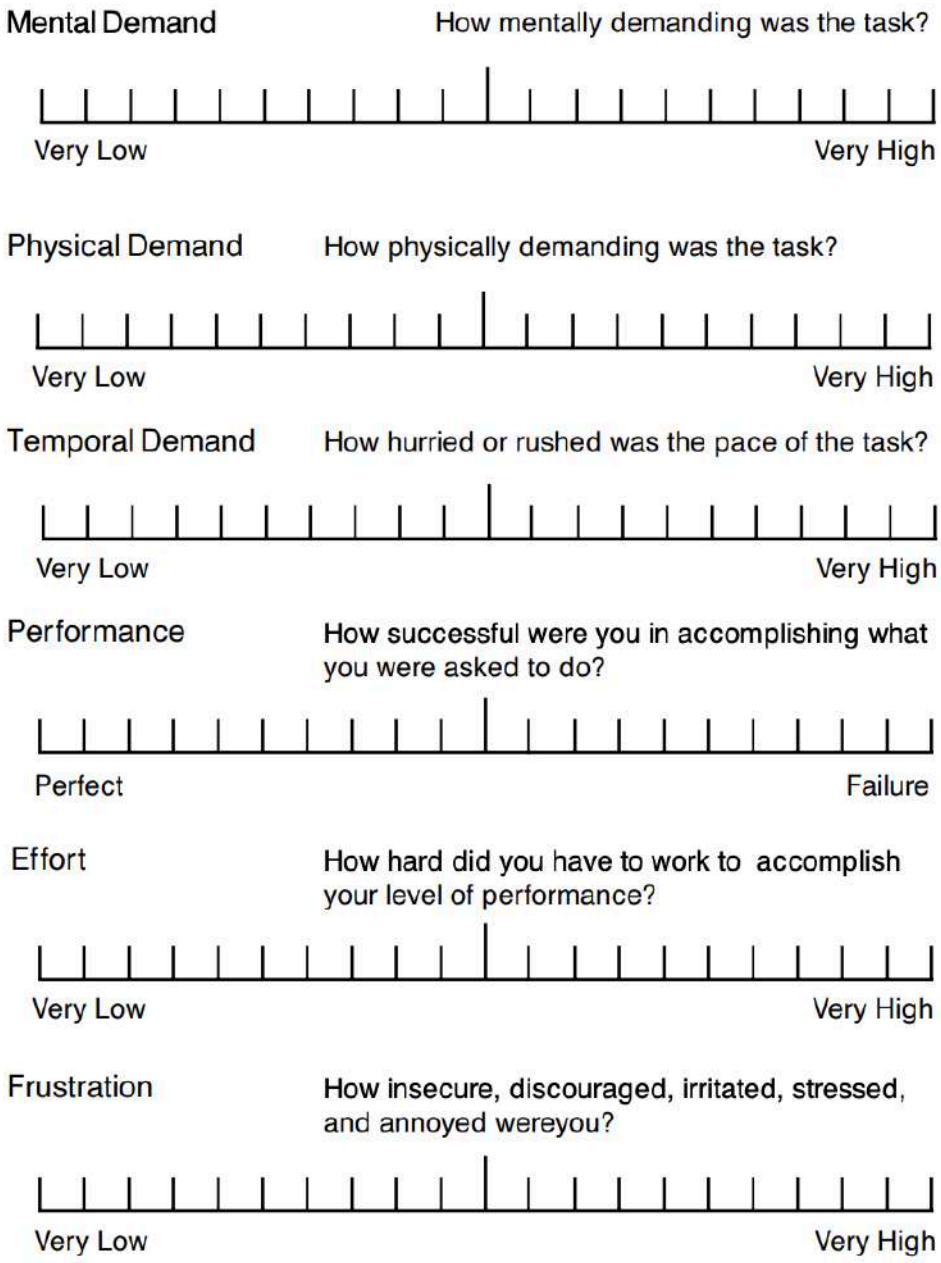
These headsets have been verified to work effectively with the software.

[18] https://idavie.readthedocs.io/en/latest/

Appendix C: SUS Questionnaire



Appendix D: NASA-TLX Questionnaire

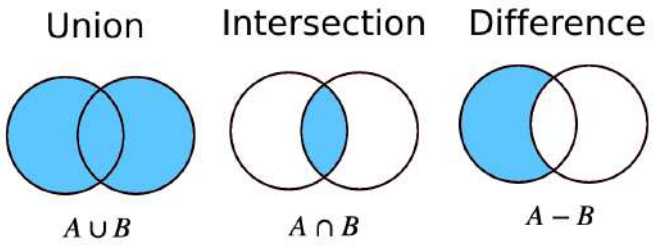


Appendix E: FSS Questionnaire

Strongly disagree 1	Disagree 2	Neither agree nor disagree 3	Agree 4			Strongly agree 5
			Strongly disagree			Strongly agree
1. I was challenged, but I believed my skills would allow me to meet the challenge.			1	2	3	4	5
2. I made the correct movements without thinking about trying to do so.			1	2	3	4	5
3. I knew clearly what I wanted to do.			1	2	3	4	5
4. It was really clear to me that I was doing well.			1	2	3	4	5
5. My attention was focused entirely on what I was doing.			1	2	3	4	5
6. I felt in total control of what I was doing.			1	2	3	4	5
7. I was not concerned with what others may have been thinking of me.			1	2	3	4	5
8. Time seemed to alter (either slowed down or speeded up).			1	2	3	4	5
9. I really enjoyed the experience.			1	2	3	4	5
10. My abilities matched the high challenge of the situation.			1	2	3	4	5
11. Things just seemed to be happening automatically.			1	2	3	4	5
12. I had a strong sense of what I wanted to do.			1	2	3	4	5
13. I was aware of how well I was performing.			1	2	3	4	5
14. It was no effort to keep my mind on what was happening.			1	2	3	4	5
15. I felt like I could control what I was doing.			1	2	3	4	5
16. I was not worried about my performance during the event.			1	2	3	4	5
17. The way time passed seemed to be different from normal.			1	2	3	4	5
18. I loved the feeling of that performance and want to capture it again.			1	2	3	4	5
19. I felt I was competent enough to meet the high demands of the situation.			1	2	3	4	5
20. I performed automatically.			1	2	3	4	5
21. I knew what I wanted to achieve.			1	2	3	4	5
22. I had a good idea while I was performing about how well I was doing.			1	2	3	4	5
23. I had total concentration.			1	2	3	4	5
24. I had a feeling of total control.			1	2	3	4	5
25. I was not concerned with how I was presenting myself.			1	2	3	4	5
26. It felt like time stopped while I was performing.			1	2	3	4	5
27. The experience left me feeling great.			1	2	3	4	5
28. The challenge and my skills were at an equally high level.			1	2	3	4	5
29. I did things spontaneously and automatically without having to think.			1	2	3	4	5
30. My goals were clearly defined.			1	2	3	4	5
31. I could tell by the way I was performing how well I was doing.			1	2	3	4	5
32. I was completely focused on the task at hand.			1	2	3	4	5
33. I felt in total control of my body.			1	2	3	4	5
34. I was not worried about what others may have been thinking of me.			1	2	3	4	5
35. At times, it almost seemed like things were happening in slow motion.			1	2	3	4	5
36. I found the experience extremely rewarding.			1	2	3	4	5

Appendix F: Boolean Operations

This appendix describes each of the operations that would be the baseline for the feature development of the selection methods.



A 2D visualisation of the three boolean operations: union, intersection and difference

Union

In the context of set theory, Boolean operations are fundamental tools for manipulating sets, which are collections of distinct objects or elements. The primary Boolean operations are union, intersection, and difference. Here, we focus on the union operation, which can be mathematically expressed using sets of items. The union of two sets, denoted as $A \cup B$, represents all the elements in A, B, or both. In terms of arrays, consider A and B as two distinct arrays. The union of these arrays consists of a new array that includes every distinct element from A and B. It is important to note that each element in a set is unique, so any duplicates are removed from the union. If $A = \{a_1, a_2, \ldots, a_n\}$and $B = \{b_1, b_2, \ldots, b_m\}$ then $A \cup B = \{a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m\}$, not repeating elements. This operation is commutative, meaning $A \cup B = B \cup A$.

Intersection

The intersection of sets is another crucial concept in set theory and Boolean operations, playing a vital role in various mathematical and computer science applications. The intersection operation is used to identify common elements between sets. Mathematically, the intersection of two sets, A and B, denoted as $A \cap B$, is the set containing all the elements that appear in both A and B. The intersection is analogous to finding the common elements between two arrays. Each element is unique in a set, and this uniqueness is also maintained in the intersection.

Formally, if $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_m\}$ then, the intersection $A \cap B$ is defined as the set of elements common to both A and B. This can be expressed as $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$. For example, consider two arrays $A = \{1,2,3\}$ and $B = \{3,4,5\}$. Then the intersection $A \cap B$ is $\{3\}$, as 3 is the only element common to both A and B. As with union, it is a commutative operation.

The concept of intersection is not just limited to two sets; it can be extended to any number of sets. The intersection of multiple sets contains only elements common to all the sets involved.

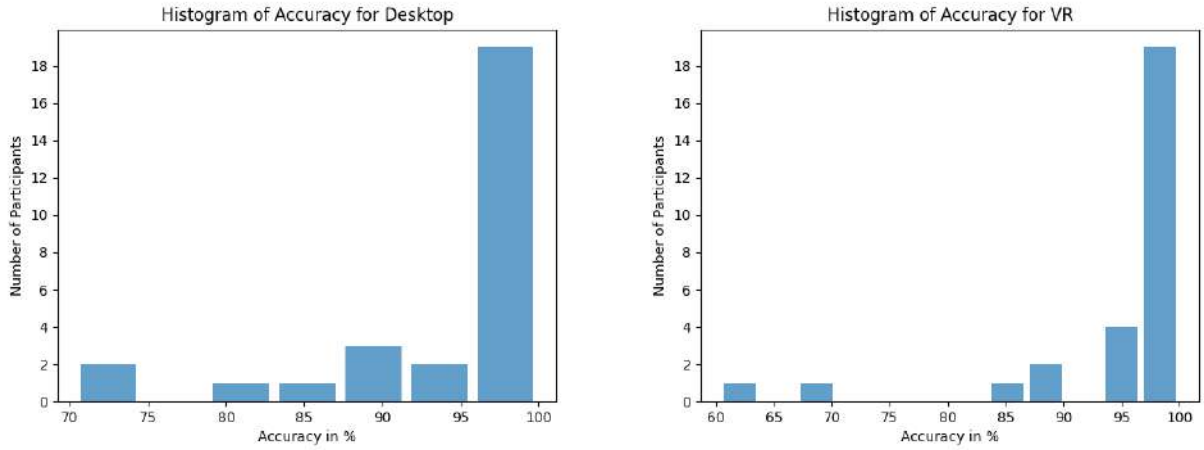
Difference

The difference of sets, also known as the set difference or relative complement, is another fundamental operation in set theory and Boolean algebra. It describes the elements that belong to one set but not another, highlighting the distinct elements in a set compared to another. Mathematically, the difference between two sets, A and B, denoted as $A - B$, is the set containing all the elements in A but not in B. When applied to arrays, the set difference corresponds to removing elements in the second array from the first array.

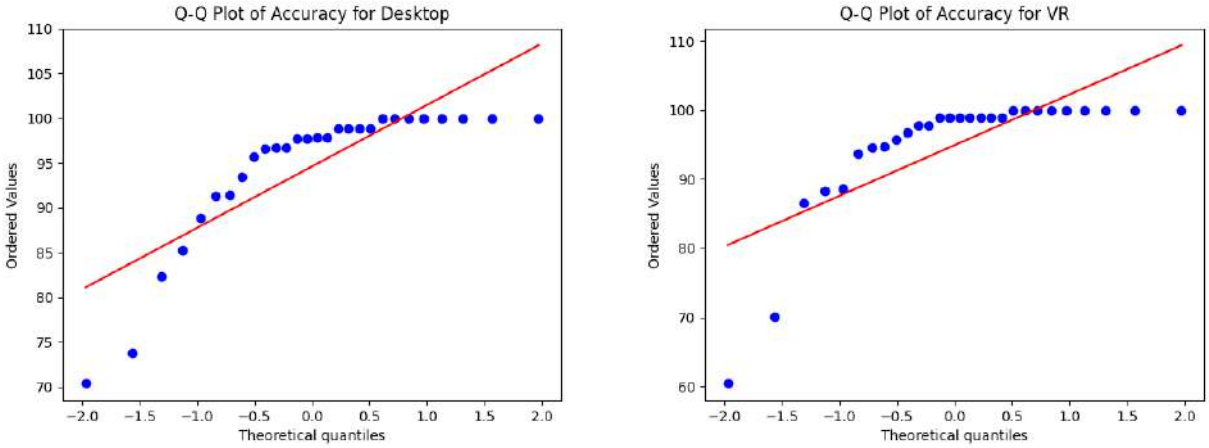
Formally, if $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_m\}$, then the difference $A - B$ is defined as $A - B = \{x \mid x \in A \text{ and } x \notin B\}$. It is important to note that set difference is not a commutative operation; $A - B$ it generally differs from $B - A$.

For example, consider two arrays $A = \{1,2,3\}$ and $B = \{3,4,5\}$. The difference $A - B$ is $\{1,2\}$. Conversely, $B - A$ is $\{4,5\}$, illustrating the non-commutative nature of this operation.

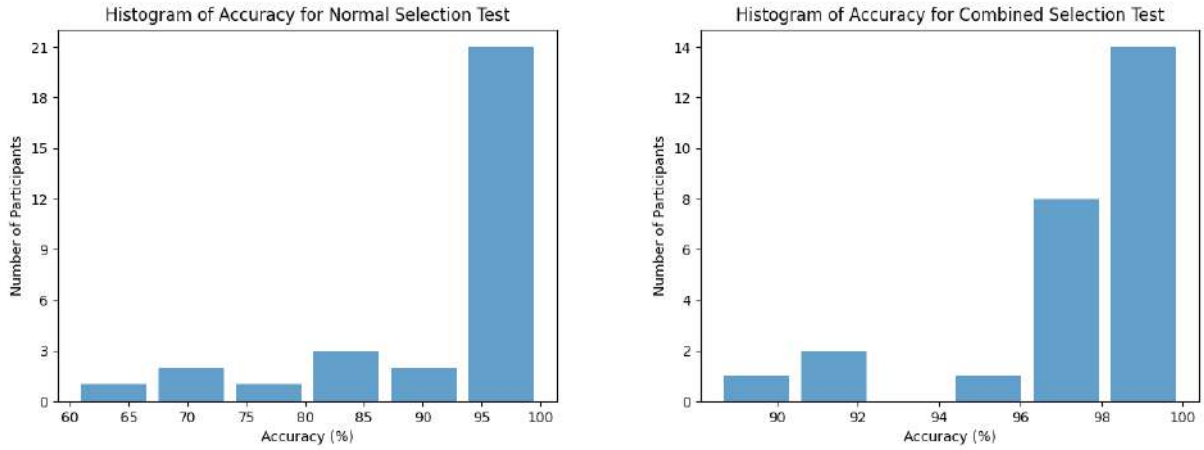
Appendix G: Plots for Accuracy Results



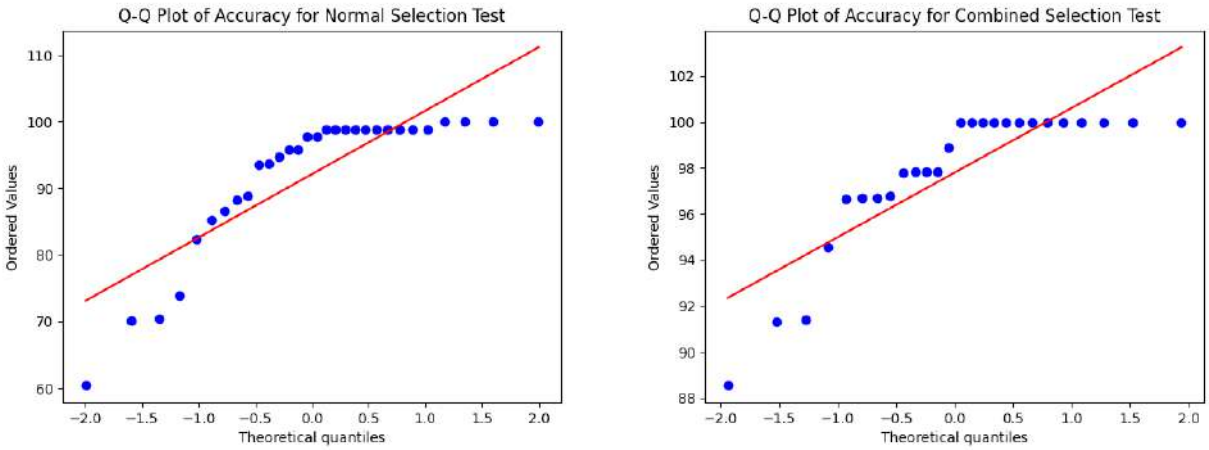
Accuracies scores of VR and Desktop



Q-Q plot of VR and Desktop for accuracy. The quantiles do not follow the theoretical distribution

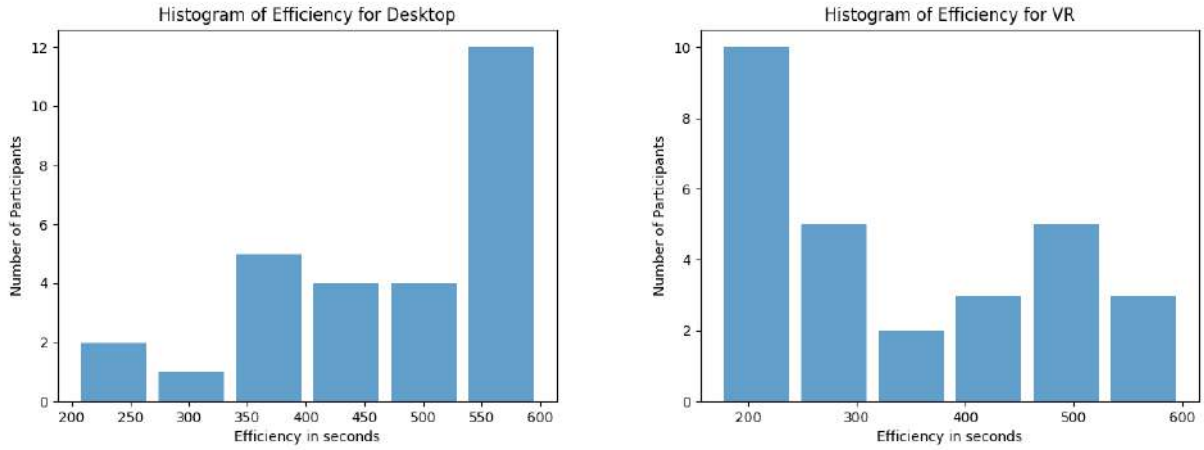


Accuracies scores of normal selection and combined selection tests

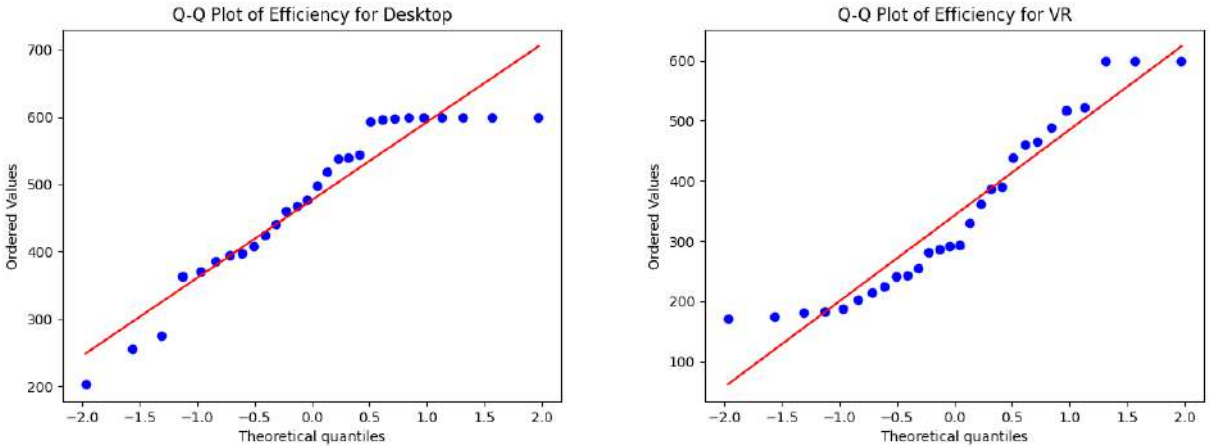


Q-Q plot of the two tests for accuracy. The quantiles do not follow the theoretical distribution

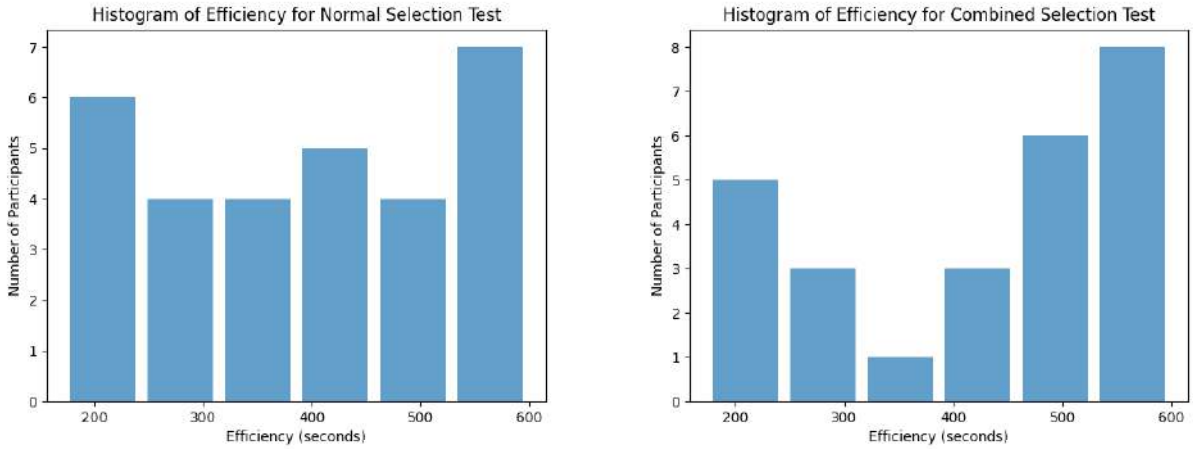
Appendix H: Plots for Efficiency Results



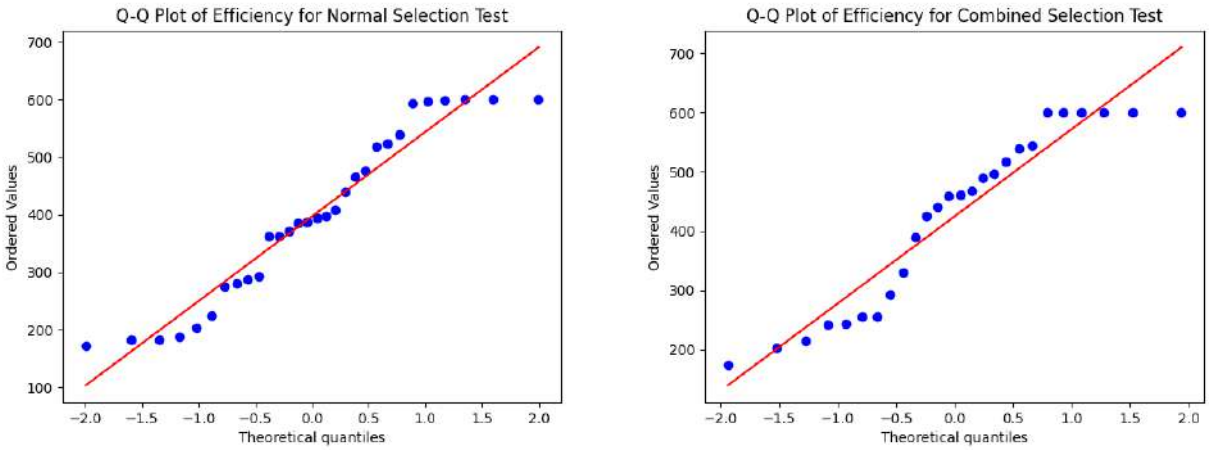
Efficiencies scores of VR and Desktop



Q-Q plot of VR and Desktop for efficiency. The quantiles do not follow the theoretical distribution

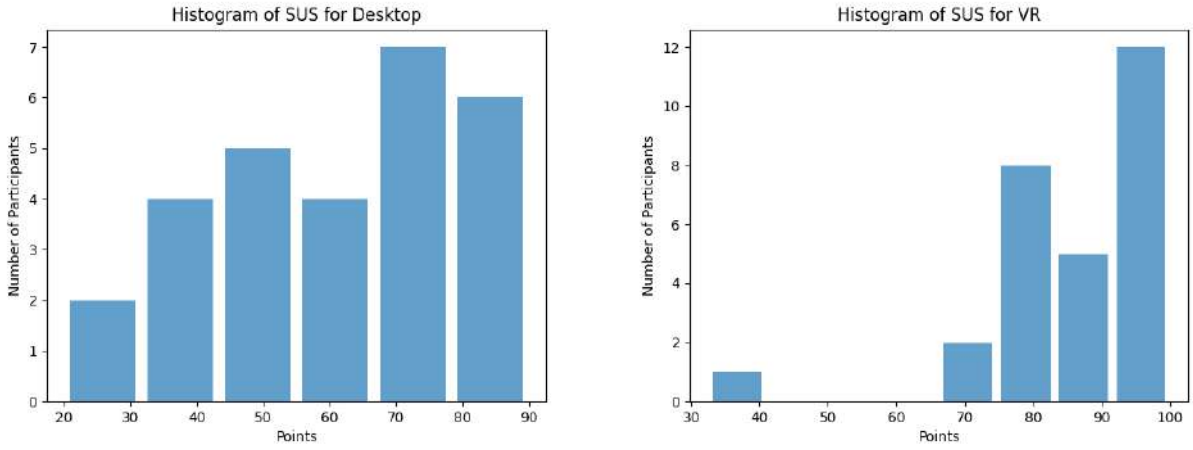


Efficiency scores of normal selection and combined selection tests

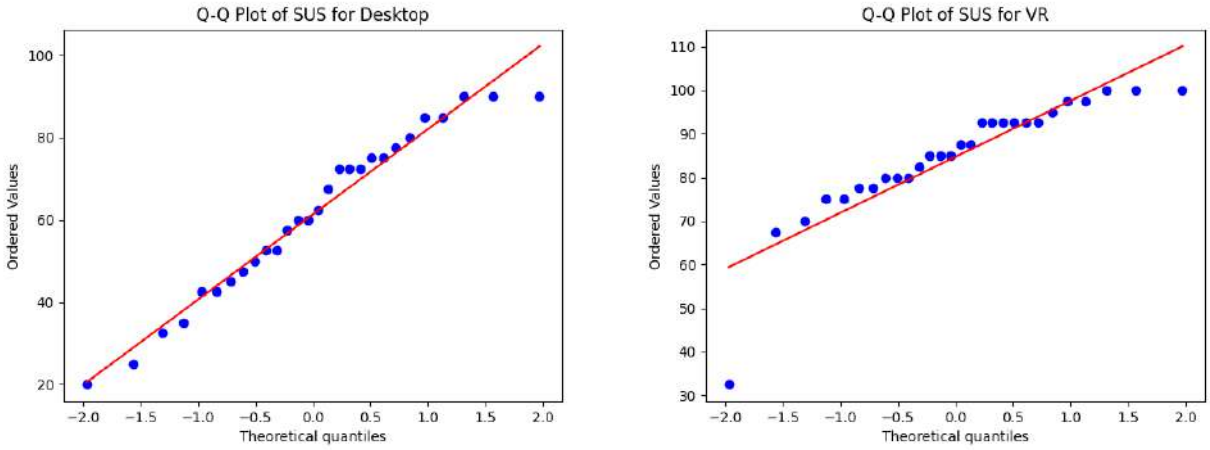


Q-Q plot of the two tests for efficiency. The quantiles do not follow the theoretical distribution

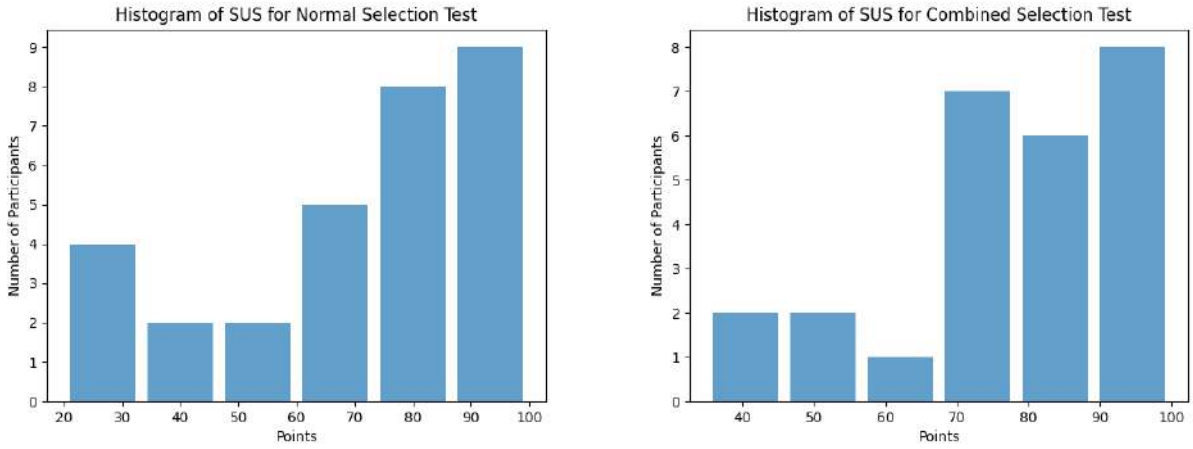
Appendix I: Plots for SUS Results



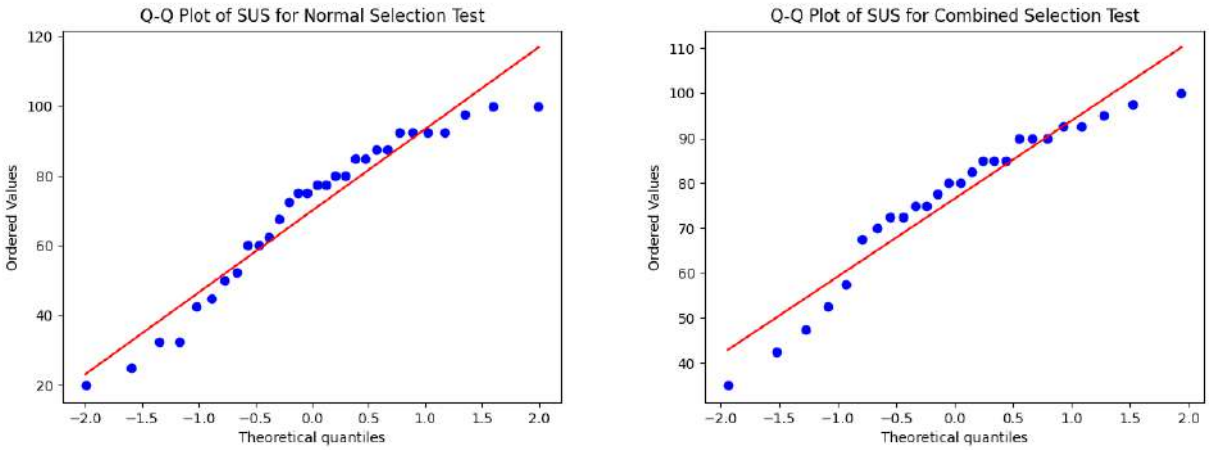
SUS scores of VR and Desktop



Q-Q plot of VR and Desktop for SUS. The quantiles on desktop follow the theoretical distribution

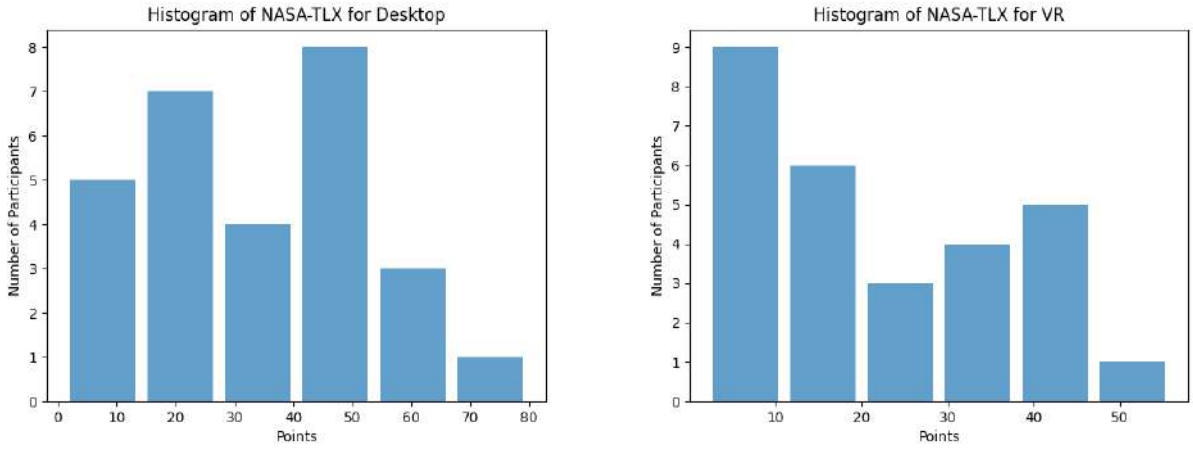


SUS scores of normal selection and combined selection tests

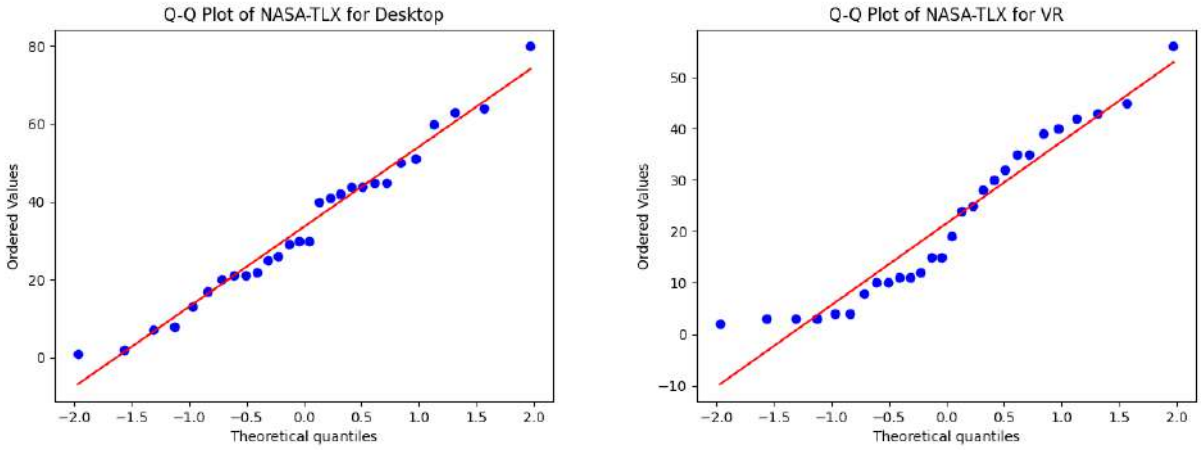


Q-Q plot of the two tests for SUS. The quantiles do not follow the theoretical distribution

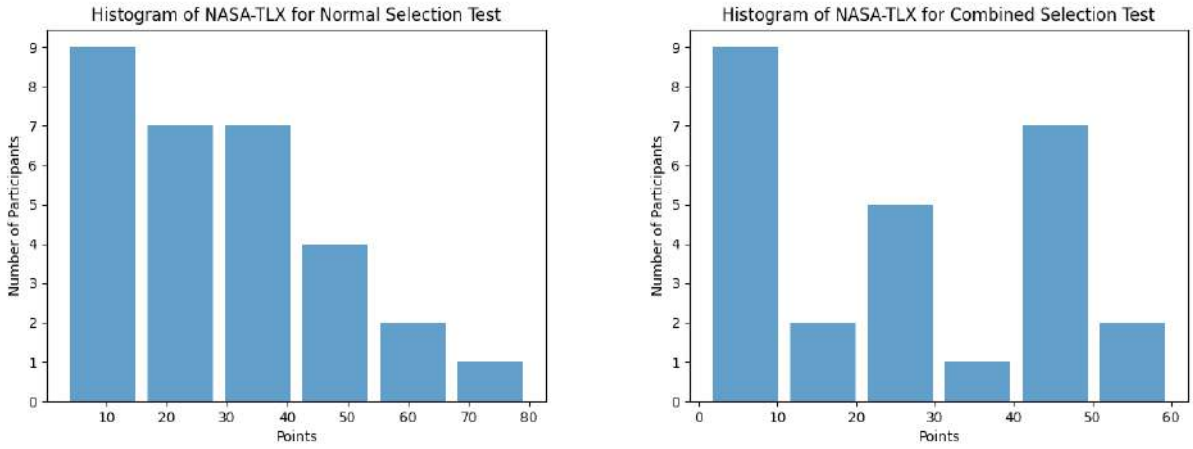
Appendix J: Plots for NASA-TLX Results



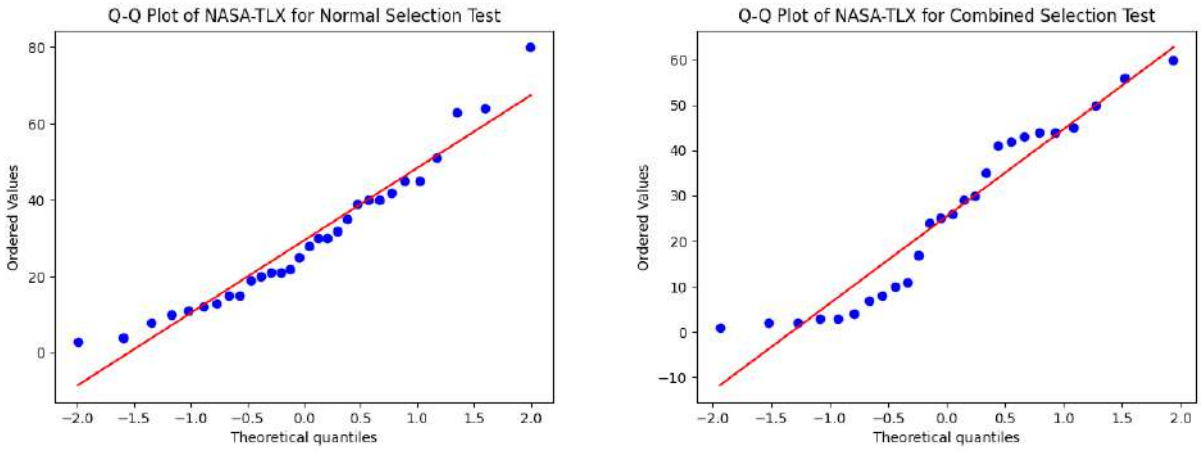
NASA-TLX scores of VR and Desktop. Fewer points means less workload, hence, a better score



Q-Q plot of VR and Desktop for NASA-TLX. The quantiles on desktop follow the theoretical distribution

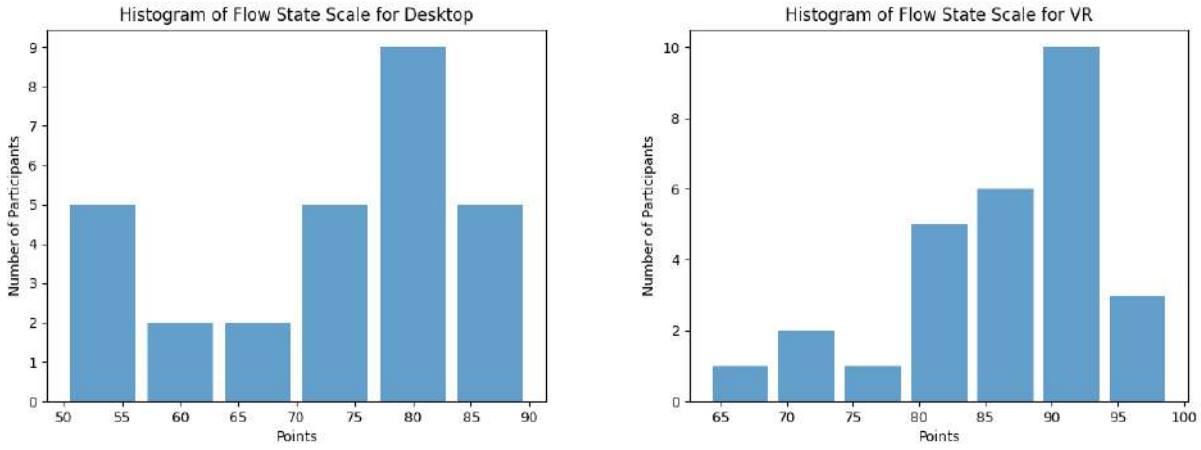


NASA-TLX scores of normal selection and combined selection tests

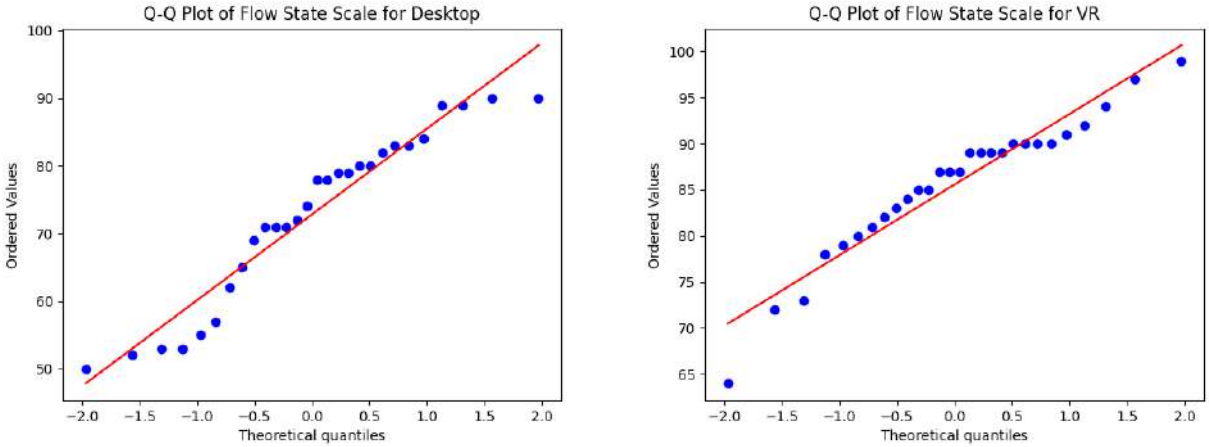


Q-Q plot of the two tests for NASA-TLX. The quantiles on the normal selection test do follow the theoretical distribution

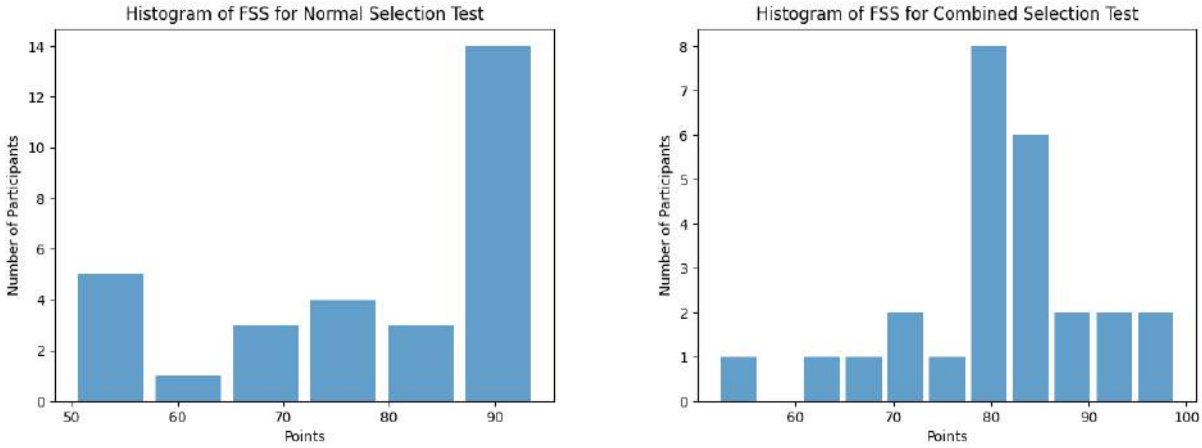
Appendix K: Plots for FSS Results



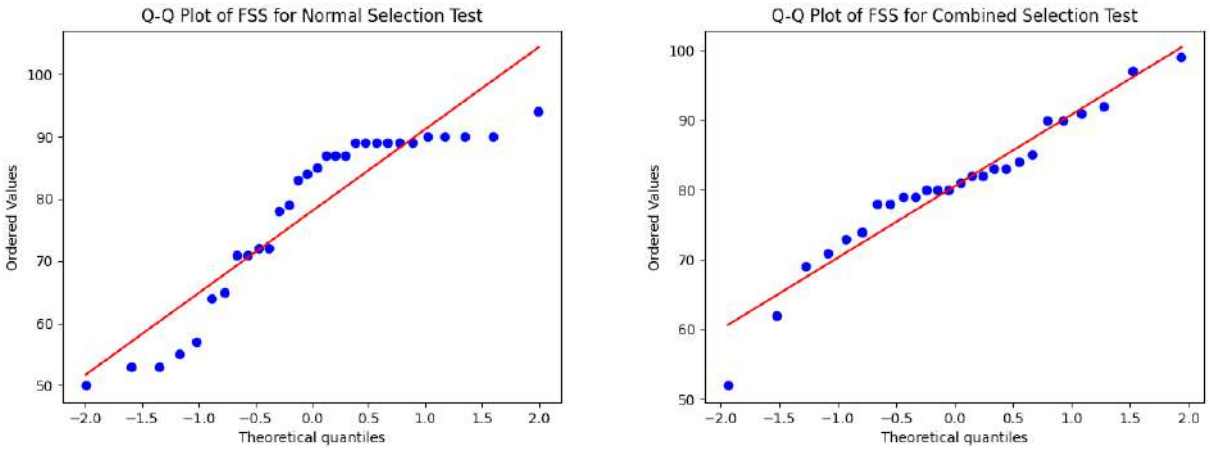
FSS scores of VR and Desktop



Q-Q plot of VR and Desktop for FSS. The quantiles on VR follow the theoretical distribution



FSS scores of normal selection and combined selection tests



Q-Q plot of the two tests for FSS. The quantiles on the combined selection test do follow the theoretical distribution

	VR Mean	VR SD	Desktop Mean	Desktop SD
Global	**85.57**	**7.48**	**72.82**	**12.36**
Challange-Skill Balance	84.29	10.91	74.64	13.82
Action Awareness	74.82	9.49	59.29	16.19
Clear Goals	91.07	9.0	79.64	14.39
Unambiguous Feedback	87.68	10.04	80.89	14.21
Concentration	88.57	9.24	73.75	14.12
Sense of Control	92.5	9.31	79.64	16.2
Loss of Self-Consciousness	80.36	12.88	69.82	13.19
Transformation of Time	75.89	15.06	59.64	13.02
Autotelic Experience	88.93	10.64	71.07	16.28

Q-Q FSS results per dimension for Desktop and VR, the better mean is marked in green