

Immersive Visual Analysis of Time-Dependent Multivariate Data Using Virtual Reality

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Visual Computing

eingereicht von

Elitza Vasileva, BSc

Matrikelnummer 01426939

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Mitwirkung: Dipl.-Ing. Harald Steinlechner

Wien, 2. Februar 2022

Elitza Vasileva

Eduard Gröller

Immersive Visual Analysis of Time-Dependent Multivariate Data Using Virtual Reality

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Visual Computing

by

Elitza Vasileva, BSc

Registration Number 01426939

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Assistance: Dipl.-Ing. Harald Steinlechner

Vienna, 2nd February, 2022

Elitza Vasileva

Eduard Gröller

Erklärung zur Verfassung der Arbeit

Elitza Vasileva, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 2. Februar 2022

Elitza Vasileva

Acknowledgements

First of all, I want to thank my parents, Elmira Vasileva and Dobrin Vasilev, for giving me this opportunity to study abroad in the first place. They supported me since the very first day at the university, and thanks to them, I could concentrate on my academic career and chase my dreams. And I am also very thankful to them for scanning necessary documents on very short notice.

I want to especially thank Harald Steinlechner for the exciting project I was able to work on and for his great support throughout my work on this thesis. He guided me the whole time and gave me constructive feedback for improving my implementation and the written part. I also want to especially thank Eduard Gröller for his valuable advice and improvement suggestions on my thesis and for his quick responses and support in stressful times. I am also very thankful to Thomas Ortner, who always gave useful input and helped bring the thesis in the right direction.

Moreover, I want to thank Thomas Maindl and Christoph Schäfer for providing the data and their valuable feedback and remarks for improving the system. Also, I want to thank them for their time discussing the requirements of the tasks and testing our system.

Furthermore, I would like to express my thanks to all my colleagues and friends from university. Without them, these studies wouldn't have been such an amazing journey. We always supported and motivated each other in difficult times. Mainly, I want to thank Dominik Scholz and Bernhard Pointner for their valuable inputs for several aspects of the implementation part of the thesis.

Last but not least, I want to thank the staff of VRVis in general and the Geospatial Visualization, Semantic Modelling and Acquisition group in particular. Most importantly, I thank Andreas Walch for his support with D3 and Attila Szabo and Georg Haaser for their help with F# and Aardvark. Also, I want to thank all 10 user study participants from VRVis for their time, feedback, and honest opinion.

This work was enabled by the Competence Centre VRVis. VRVis is funded by BMK, BMDW, Styria, SFG, Tyrol, and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (879730), which is managed by FFG.

Kurzfassung

Die Verhinderung von Erdkatastrophen wie Asteroidenkollisionen hat für viele Organisationen und Wissenschaftler hohe Priorität. Die Untersuchung möglicher Ablenkungsoptionen für Asteroiden und das Verständnis der Einflussfaktoren sollten Fachexperte dabei helfen, Verteidigungsstrategien und Reaktionspläne zu erstellen. Zu Forschungszwecke verwenden Wissenschaftler in der Regel Simulationen, die einen realen Prozess über einen bestimmten Zeitraum abbilden und somit ein effektives Mittel sind, um einen Asteroideneinschlag und seine Folgen zu demonstrieren. In dieser Arbeit wird ein interaktives Visualisierungstool in Virtual Reality (VR) verwendet, um Simulationsergebnisse nachvollziehbar zu machen. Immersive Systeme wie VR werden in verschiedenen Anwendungsbereichen und zu unterschiedlichen Zwecken eingesetzt, wie zum Beispiel zur Unterhaltung, für medizinisches oder militärisches Training, Rehabilitation, Psychotherapie, Visualisierung und visuellen Analysen. Mit Hilfe von VR-Software und der notwendigen Hardware von immersiven Systemen können realistische Bilder und 3D-Szenen aus der realen Welt dargestellt werden, um den Eindruck des völligen Eintauchens und der Anwesenheit zu erzeugen.

Ausgehend von diesen Überlegungen entwickeln wir in dieser Arbeit ein interaktives Visualisierungstool in VR, um Fachexperte bei der Untersuchung der Eigenschaften und Merkmale von Asteroideneinschlägen zu Verteidigungszwecken zu unterstützen. Dazu verwenden wir die zeitabhängige multivariate *Einschlagssimulationsdaten*. Die Umsetzungsanforderungen werden gemeinsam mit Fachexperten in Form von Aufgaben formuliert und stellen die wesentlichen Eigenschaften dar, die das System enthalten sollte. Als Ergebnis enthält das System eine 3D-Punktwolkenvisualisierung zur Veranschaulichung des Einschlags und der Datenstruktur sowie verschiedene Explorationswerkzeuge zur Analyse und Untersuchung der Punktwolkeneigenschaften. Das zentrale Werkzeug im System wird Probe genannt und ermöglicht es, die Eigenschaften verschiedener Bereiche zu messen, sie miteinander zu vergleichen und Zustandsänderungen im Laufe der Simulationszeit zu beobachten. Neben der effektiven Untersuchung ist die Interaktivität ein weiterer wichtiger Faktor, der zu einem flüssigen und natürlichen Erlebnis beiträgt. Daher bieten wir verschiedene Techniken zur Navigation und Objektmanipulation an, um eine intuitive und reibungslose Interaktion mit dem System zu ermöglichen. Da die Auswahl der Explorationswerkzeuge für die Fachexperte und die Lösung ihrer Aufgaben von entscheidender Bedeutung ist, evaluieren wir unser System zunächst mit den Fachexperten, um festzustellen, ob das System die erforderlichen Funktionen bereitstellt

und ihre Anforderungen erfüllt. Ein weiterer wichtiger Aspekt unseres Systems ist die Interaktivität und Benutzerfreundlichkeit, die wir durch eine Benutzerstudie evaluieren. Wie wir in unseren Evaluierungsexperimenten zeigen, erleichtert unser VR-System den Wissenschaftlern den Explorationsprozess. Es unterstützt sie dabei, neue und bisher unentdeckte Eigenschaften, Muster und Tendenzen in den Daten zu finden.

Abstract

The prevention of Earth disasters like asteroid collisions is a high priority for many organizations and scientists. Studying possible asteroid deflection options and understanding the influencing factors should help domain experts create defense strategies and response plans. For exploration purposes, scientists typically use simulations that represent a real-world process over time and are thus an effective way for demonstrating an asteroid impact collision and its after-effects. In this work we use an interactive Virtual Reality (VR) visualization tool to make simulation results comprehensible. Immersive systems like VR are used in different application domains and for multiple purposes, such as entertainment, medical or military training, rehabilitation and mental therapy, visualization, and visual analytics. With the help of VR software and the necessary hardware of immersive systems, realistic images and 3D scenes from the real world can be rendered to create the feeling of full immersion and presence.

Based on these considerations, in this work, we develop an interactive visualization tool in VR to support domain experts in studying the properties and features of asteroid impact events for defense purposes. We use time-dependent multivariate *Impact simulation data*. The implementation requirements are formulated together with domain experts in the form of tasks and represent the main features that the system should include. As a result, the system incorporates a 3D point cloud visualization to illustrate the impact and the data structure and various exploration tools to analyze and examine the point cloud properties. The central tool in the system is called a probe, allowing to measure the characteristics of different regions, compare them, and observe state changes during simulation time. While effective exploration is the primary goal of our system, interactivity is another important factor contributing to achieving a smooth and natural experience. Therefore, we provide various grasping and navigation techniques to support an intuitive and effortless system interaction. As the selection of exploration tools is essential for the domain experts and for solving their tasks, we first evaluate our system with them to answer whether the system is providing the necessary features and is fulfilling their requirements. Another important aspect is the interactivity and usability of our system, which we evaluate through a user study. As we show in our evaluation experiments, our VR system eases the exploration process for scientists. It supports them in finding new and previously undiscovered properties, patterns, and trends in the data.

Contents

Kurzfassung	ix
Abstract	xi
Contents	xiii
1 Introduction	1
1.1 Motivation and Problem Statement	2
1.2 Contribution	4
1.3 Structure of the Thesis	6
2 Related Work	7
2.1 Virtual Reality and Fields of Application	7
2.2 Empirical Evidence for the Benefits of Virtual Reality	13
2.3 Virtual Reality for Visualization and Immersive Exploration	15
2.4 Virtual Reality Toolkits and Frameworks for Creating Visualizations	20
2.5 Perception, VR Sickness, and Performance in Virtual Reality	23
2.6 Interactions and Metaphors in Virtual Reality	27
3 Analysis of the Time-Dependent Multivariate Impact Simulation Data	35
3.1 The Impact Simulation Data	35
3.2 Domain Expert Tasks for the Impact Simulation Data Analysis	36
4 Impact Simulation Data with 2D User Interfaces	39
4.1 Methodology and Implementation	39
4.2 Results and Limitations of the 2D Approach	46
5 Impact Simulation Data in Virtual Reality	49
5.1 Overview	49
5.2 System Components and VR Solutions to Domain Experts' Tasks	49
5.3 Challenges	50
5.4 Visualization Techniques to Support Impact Simulation Data Exploration in VR	52
5.5 Interaction with the Impact Simulation Data in VR	73
	xiii

6	Perceptual Considerations and Optimizations in VR	89
6.1	Impostor Rendering of Spheres	89
6.2	Transparency with Alpha Blending	92
6.3	Occlusion Handling for Billboards	95
6.4	Performance Overview and Motion Sickness	98
7	Implementation	103
8	Evaluation and Discussion	105
8.1	Target Group and User Study Design	105
8.2	Evaluation with Domain Experts	106
8.3	User Study	109
8.4	Discussion	129
9	Conclusion and Future Work	131
9.1	Conclusion	131
9.2	Future Work	133
	List of Figures	135
	List of Tables	139
	List of Algorithms	141
	Acronyms	143
	Bibliography	145

Introduction

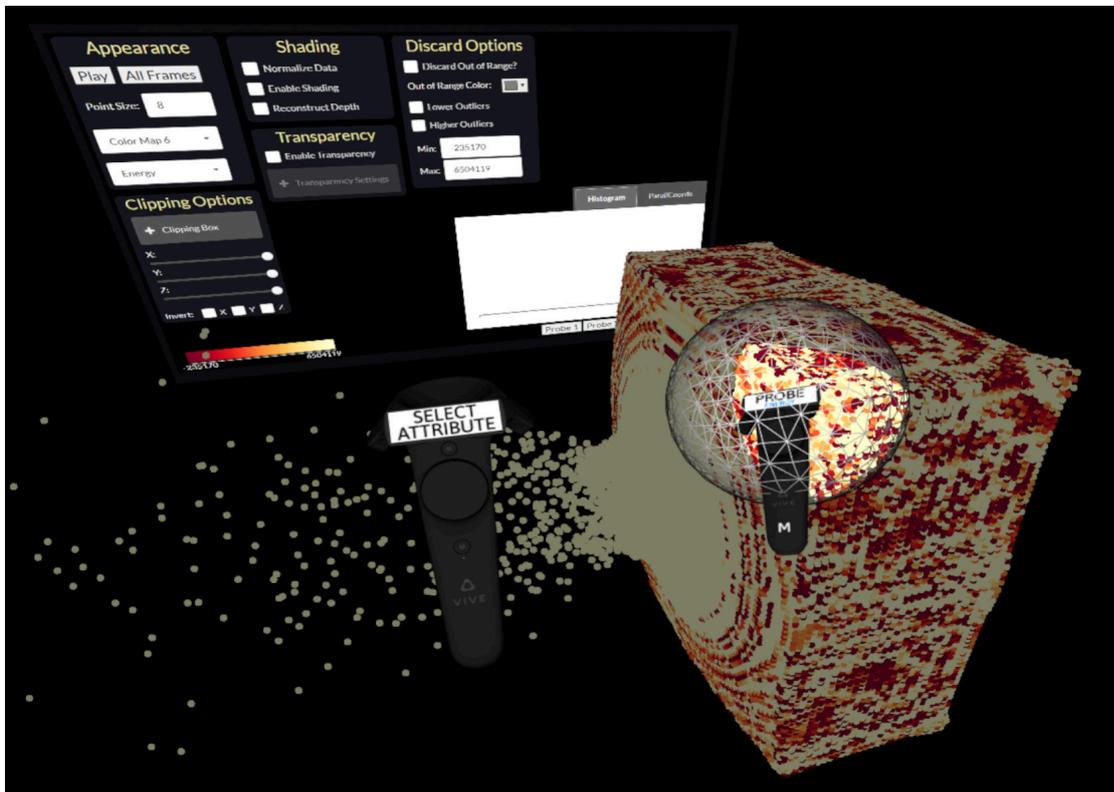


Figure 1.1: Interactive exploration of the *Impact simulation data* in VR

This work presents a novel system for visualization and interactive exploration of time-dependent multivariate data in Virtual Reality (VR). The work is conducted as a design study on the *Impact simulation data* provided by Maindl et al. [MS19]. The data

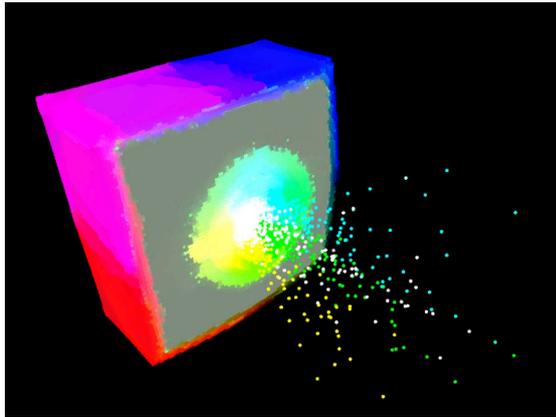


Figure 1.2: Single frame from the *Impact simulation data*

simulation represents the surface shattering and alternation of an asteroid moon after a collision with an impactor spacecraft. The purpose is to analyze how such an impactor could deflect an asteroid to avoid a possible collision with Earth and investigate the factors influencing the deflection. The simulation consists of 400 frames and more than a million 3D points, changing their spatial location over time. Each point has its own attributes that define its properties. The attributes include position, speed, mass, density, energy, pressure, porosity, and other domain-specific internal simulation values. A screenshot of a single frame from the animation of the *Impact simulation data* is shown in Figure 1.2. The goal is to interactively explore the data and analyze and compare different regions in Virtual Reality by providing various tools and informative visualizations. The resulting VR visualization and exploration system of the *Impact simulation data* can be seen in Figure 1.1.

For the astronomy background and applications, we used data provided by the University of Tübingen and the University of Vienna. The design was constantly accompanied by domain knowledge and feedback from partner astronomers - Thomas Maindl and Christoph Schäfer.

1.1 Motivation and Problem Statement

Why and how are visualizations and interactions important?

Data visualization has become an integral part of many scientific disciplines and, even in social lives [OOKO15]. It plays a crucial role in helping interpret and make sense of complex data since it is the main bridge between the raw content of the data and the human intuition, allowing to gain knowledge and insight. With the help of powerful tools and visualizations such as graphs, plots, and charts, researchers can gain visual context and structure of the data. These visualizations are more natural to understand and make identifying trends, patterns, and outliers in large datasets easier. Modern technologies,

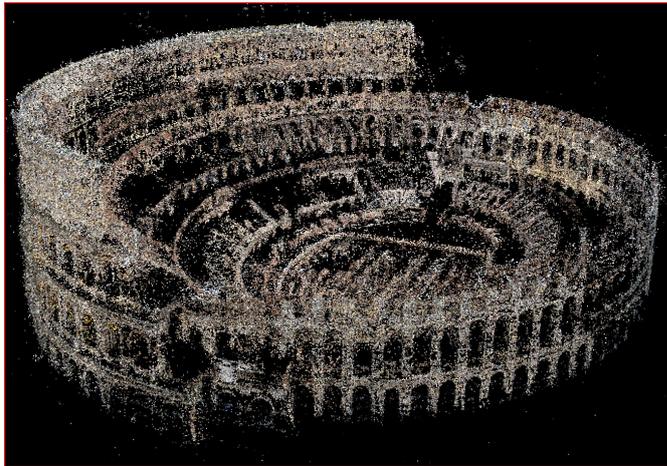


Figure 1.3: 3D point cloud model of the Colosseum in Rome [SCS⁺10]

such as computers, expand the possibilities for visualizations by also providing interaction techniques. Through interactions, scientists can manipulate the visualization, change the viewing perspective, filter out parts, highlight regions, and explore and depict data in a clear and meaningful way. However, each data type has its own properties, like dimensionality, domain, structure, and many more. For this reason, depending on the geometry and topology of the data being modeled and the intended purpose, various types of visualizations can be utilized. For more complex or novel data, finding the most appropriate visualization can sometimes be challenging, which is why the research is so extensive in the field and continues growing by searching and exploring innovative and more expressive ways for visualizing data.

What is specific about the *Impact simulation data*?

The *Impact simulation data* in this work is, in its nature, time-dependent multivariate data consisting of multiple 3D points. The representation of a 3D object or model via a large number of 3D points is called a point cloud (Figure 1.3). Point clouds are most commonly obtained by scanning the real world through various scanning methods, such as laser scanning or photogrammetry. While laser scanners typically produce colored points and radiance values, simulations intrinsically add the time component and other multivariate data, including various attributes and dozens of domain-specific internal simulation values. Therefore, the visualization and exploration of data like the *Impact simulation data* is even more complex since the domain experts need more measurement values and attributes to explore and understand. Also, the time-dependent aspect of the simulation data makes the exploration more challenging and requires the introduction of specific tools supporting temporal exploration.

Why using Virtual Reality to visualize the *Impact simulation data*?

The analysis of time-dependent multivariate data requires spatial and temporal queries and various plots and graphs for visual exploration to gain insight. While visualization

and basic interaction techniques in 2D space work well for various tasks, navigation, spatial queries, and analysis of multivariate data are particularly challenging when using 2D user interfaces [DRST14]. As Cordeil et al. describe in their paper [CCD⁺17], using a 2D screen to project 3D data suffers from various inherent issues such as perspective distortion, occlusions, and interaction limitations using a 2D input device. In this context, we expect that an immersive system for exploring the *Impact simulation data* would offer more freedom and more exploration and interaction possibilities.

Previous research shows an evidence for the benefits of using VR immersive systems to explore three- or higher-dimensional data [LPF⁺09][CCD⁺17][OJC⁺17]. Immersion provides numerous advantages compared to traditional 2D visualization tools [ACC99]. When dealing with multivariate data obtained from a 3D medium, its analysis and exploration is a natural application in VR, since it grants freedom to intuitively interact with the environment by observing the data from arbitrary angles and positions as it happens in the real world. A VR environment allows the direct selection of 3D regions of interest, creating 3D annotations, and performing simple measurements [EDC⁺19]. These exploration capabilities that immersive systems offer are particularly relevant for the *Impact simulation data*, since one of the main goals is to investigate different regions of interest and measure values of the given attributes. The possibility to be "within" the data and directly interact with the model speeds up the process of its understanding and interpretation [LPF⁺09]. Therefore, we will explore the visualization and interaction possibilities for time-dependent multivariate point cloud data by immersion into Virtual Reality and will evaluate the effectiveness of this technology for such a type of data. The work will be conducted as a case study on the *Impact simulation data* in VR with an HTC Vive hardware.

1.2 Contribution

This thesis is a continuation of a former baseline implementation of a 2D system [VS20] for visualization and analysis of the *Impact simulation data*, which is provided by T. Maindl and C. Schäfer [MS19]. The implementation requirements were defined together with the domain experts in the form of specific tasks that should be solved with the help of the application. Within the baseline project's scope, a system was implemented consisting of a 3D point cloud of the asteroid points with a 2D UI on a traditional desktop to interact with the environment. The 2D UI elements provide various interaction possibilities with the 3D view. The primary exploration tool is the *probe*, which, in general, represents a 3D object that can be resized and placed anywhere in the point cloud, allowing domain experts to investigate the region's features. The *probe* in the baseline implementation is a cuboid with a fixed position and size and cannot be transformed and is used to explore only the region at the impact crater. The exploration happens with the help of two interactive visualizations embedded in the application - a histogram and a parallel coordinates plot. These visualization plots represent the data distribution and relationships. They control and facilitate the direct manipulation of the 3D view by brushing and filtering. The probes are settled at a fixed position because, according to

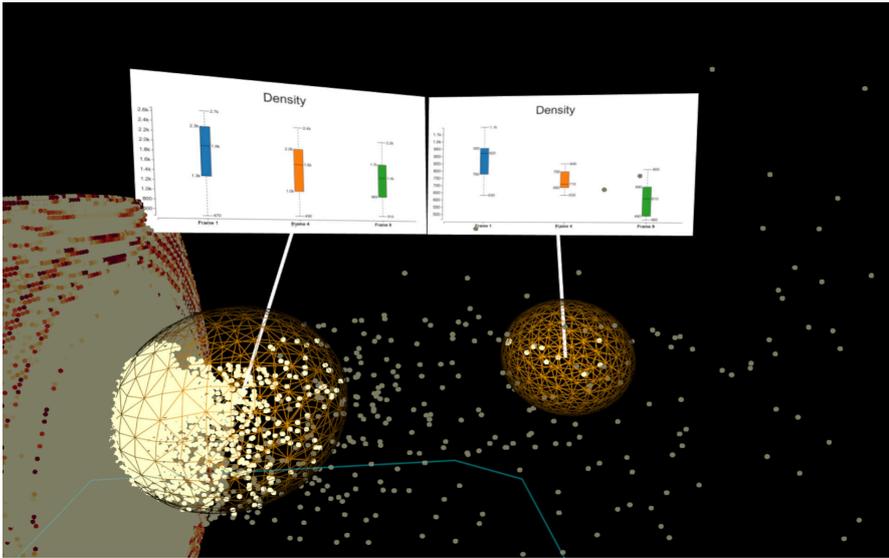


Figure 1.4: Region comparison with two spherical probes for the attribute density

the domain experts, the visualization center is the most interesting and informative part for analyzing. Furthermore, it is cumbersome to manually select the desired cuboids in 3D space precisely using a 2D screen and a 2D interaction device due to occlusions and perspective distortions.

In this thesis, we aim to provide a novel system in Virtual Reality that can fulfill the exploration requirements and tasks of the domain experts and allow the visualization and interactive analysis of the multivariate time-dependent *Impact simulation data*. The requirements and tasks were defined after several meetings and discussions with the domain experts. Since the *probe* is the central exploration element, with the help of an immersive system with stereoscopic vision and controllers with six Degrees of Freedom (6DOF), the user can easily create probes manually by using the controllers and in this way, also explore other regions than the impact crater. Additionally, we aim to provide tools with expressive statistical plots, such as box plots, to compare states of the simulation at different regions and time-steps (Figure 1.4). To find the most meaningful and informative visualizations and interaction techniques for the given data and task, various aspects are researched and considered. Based on our findings, we propose a flexible VR system to solve all essential tasks and ease the exploration and understanding of the *Impact simulation data*. In our work, we also formulate various tasks together with the domain experts to evaluate the resulting system's effectiveness in a user study. To summarize, in this thesis, we have the following contributions:

- We propose a **novel VR system** for visualizing and interactively exploring the *Impact simulation data* to support domain experts in studying the features of impact events.

- We **expand the 2D UI** controls from the baseline application and integrate them into VR to provide additional settings for data visualization and analysis.
- For facilitating exploratory tasks, we provide **spherical probes** with modifiable size to place them on regions of interest and retrieve a graphical representation of attribute distributions displayed by histograms. Moreover, our system supports multiple regions comparison to observe attribute changes and properties at several locations simultaneously. Lastly, we also support a single probe's time comparison, allowing domain experts to observe attribute changes at a specific location for multiple frames.
- We propose various **rendering modes** (impostor rendering, transparency) tailored explicitly for VR to improve perception and visually enhance the properties of the point cloud.
- Combining various menu types and **interaction techniques**, we allow the system manipulation in numerous ways. We provide close-range grasping techniques, long-range pointing techniques for object manipulation, radial menus at the VR controllers, and static 2D menus placed in the 3D space.
- We demonstrate the effectiveness of our system by conducting an **evaluation with a domain expert**, who tests various aspects of the resulting system and solves various tasks he would need when doing research.
- Lastly, we conduct a **user study** to determine the system's usability, expressiveness, and interactivity. The users perform various pre-defined exploration and analysis tasks. In the end, they answer several questionnaires to rate different aspects of the system.

1.3 Structure of the Thesis

Chapter 2 gives an overview of related work, including various fields and applications where VR is integrated, different systems for data visualizations, interactions, and metaphors in VR, and previous research regarding possible issues concerning immersive environments. Chapter 3 informs about the requirements and domain experts' task for the analysis of the *Impact simulation data*. A description of the baseline project implementation used in this work for visualizing the *Impact simulation data* on traditional 2D displays and interaction devices is given in Chapter 4. Chapter 5 gives a detailed description of the resulting VR system for visualizing and exploring the *Impact simulation data* using multiple visualizations, interaction techniques, and animations. Different perceptual issues and performance challenges and their solutions are discussed in Chapter 6. Chapter 7 covers implementation and structural details of the system. A domain expert evaluation and a user study investigating the effectiveness, usability, and intuitiveness of the system is presented in Chapter 8. Chapter 9 concludes the thesis and provides information about possible future work and further developments.

Related Work

This chapter overviews virtual reality technology, including its history, different usages, benefits, as well as some issues and limitations of immersive systems. The chapter is separated into six different sections, each covering a different aspect of VR. The first section gives a brief overview of the history of VR and its various fields of application to introduce the reader with this kind of technology and its diverse usages. After that follows a section covering empirical pieces of evidence for the benefits of using Virtual Reality for different domains and tasks. Section three describes immersive data analysis and visualization with the help of VR environments. Information about different visualization libraries and toolkits used in VR is given in the fourth section. The fifth section represents the performance measures of such a system, including motion-sickness and perception issues. The final section summarizes possible interactions, navigation, and system control techniques used in immersive environments.

2.1 Virtual Reality and Fields of Application

2.1.1 Brief Overview of VR History

Virtual Reality is a technology that exists for already more than five decades and provides entirely artificial, computer-generated environments. Ivan Sutherland presented the very first concept of such kind of technology in 1965. His idea was to “make that (virtual) world in the window look real, sound real, feel real, and respond realistically to the viewer’s actions” [Sut65]. Since then, many different immersive systems and devices have been created, such as GROPE, VIVED, CAVE [MG99] and some modern ones like HTC Vive, Oculus Rift [APSK20]. Although there are many definitions of how VR actually can be described, they all portray an experience that is interactive and immersive (with the feeling of presence) in a simulated (autonomous) world [Zel92]. Similarly to that, Burdea and Coiffet [BC03] present the main objective of Virtual Reality; they formulate it as the 3I – immersion, imagination, and interaction, shown in Figure 2.1.

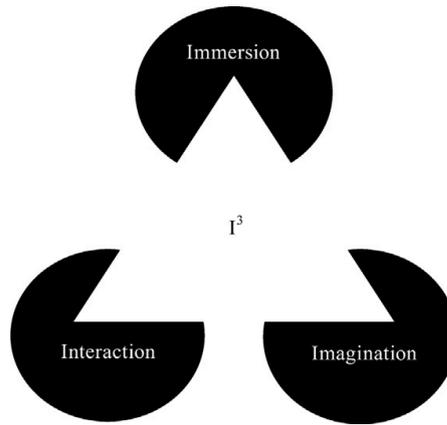


Figure 2.1: 3I of Virtual Reality [BC03]

As Mazuryk and Gervautz [MG99] describe in their research, there are various kinds of VR systems (Desktop VR, Fish Tank VR, Immersive Systems), each of them providing a different level of immersion. Immersive systems are the ultimate version that allows the user to totally immerse in a computer-generated world. The structure of VR systems usually includes an input device, output device, and software. The input devices allow the user to interact and communicate with the system. The output devices represent the virtual environment and mostly contribute to the generation of an immersive feeling. The software also plays a significant role in the system since its primary purpose is to manage the I/O devices by analysing the incoming data and generating proper feedback. Figure 2.2 illustrates a typical structure of the components of a VR system.

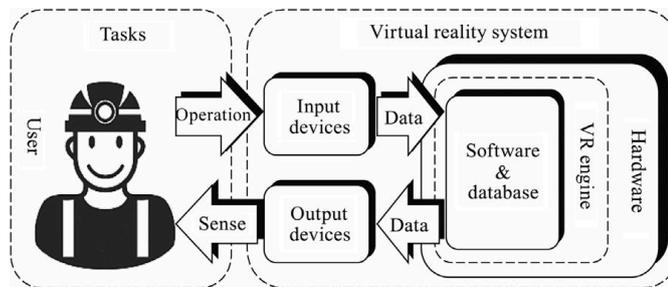


Figure 2.2: Components of a VR system [Zha17]

2.1.2 Application domains of VR

Since VR grants a total immersion experience, provides an environment with 6DOF, and allows the precise detection of movements within the rendered scenes through continuous motion tracking, it finds usages in many different fields:

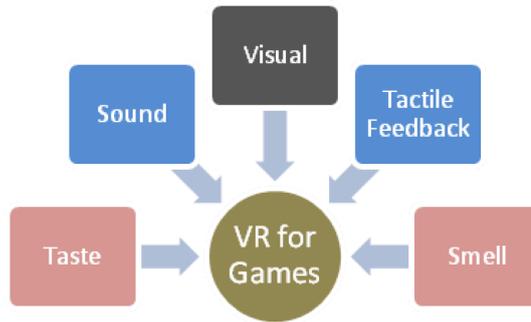


Figure 2.3: Types of Senses in VR [Dan19]

Entertainment

Video games are one of the leading entertainment media at present [Dan19]. One of the most common applications of Virtual Reality systems can, thus, be found in the gaming industry. VR allows the users to be present in another world and reality, interact with objects, and explore the environment as if they are in the real world. To achieve the notion of realistic situations, VR systems focus on the five types of senses (Figure 2.3) a user can experience without the risk of causing actual harm or damage. Some popular VR games can be seen in Figure 2.4.

Industry

Except for entertainment, Virtual Reality has also been actively used in the industry and for manufacturing. It helps engineers and designers to explore and interact with the data in a natural way and thus supports decision making and enables innovations [BV17]. One example is its application in the automotive industry. Ford Motor Company, for instance, has a virtual manufacturing program that not only allows them to shape the design of their cars, but is also used as training for the employees (Figure 2.5a). The goal is to evaluate the overall safety of the assembly process and also determine task feasibility and proficiency to ensure high-quality vehicles for the customers [For15]. Similarly, VR is also used in the aerospace industry. For instance, Embraer’s Virtual Reality Center in Brazil (Figure 2.5b) was established to help engineers visualize the structure and system of an aircraft during the development phase to reduce the time needed for building a new aircraft [FSB16].

Training and Education

Many areas and scientific fields require specific knowledge, set of skills, and training in advance to perform a particular task. Some of these fields like mathematics, physics, or mechanics, however, often include challenging tasks that require a high level of spatial imagination and good orientation in 3D space [HVV21]. In such cases, a Virtual Reality platform can help in reducing complexity and provide a more natural studying environment. Hácha et al. [HVV21] provide such a framework for immersive education

2. RELATED WORK

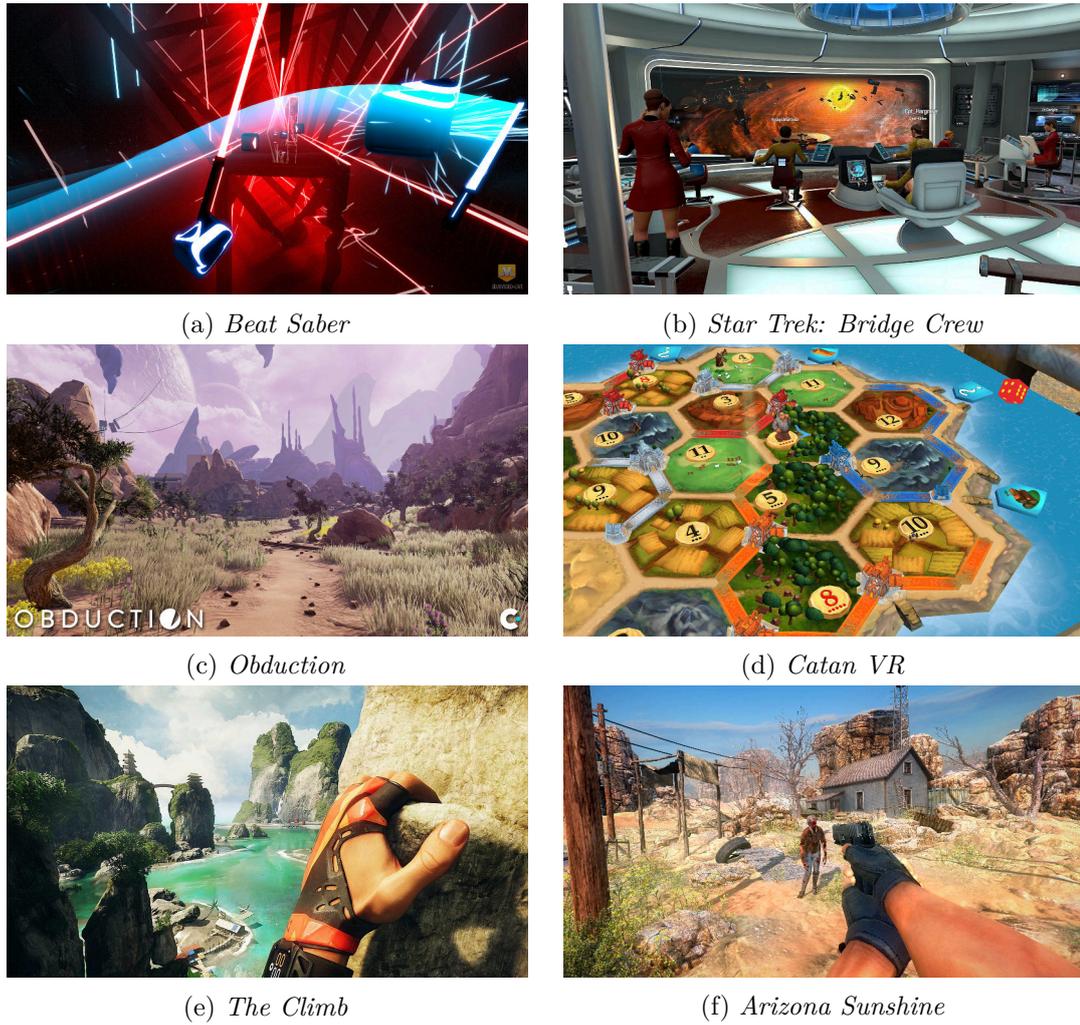


Figure 2.4: Popular VR Games [Ste18]

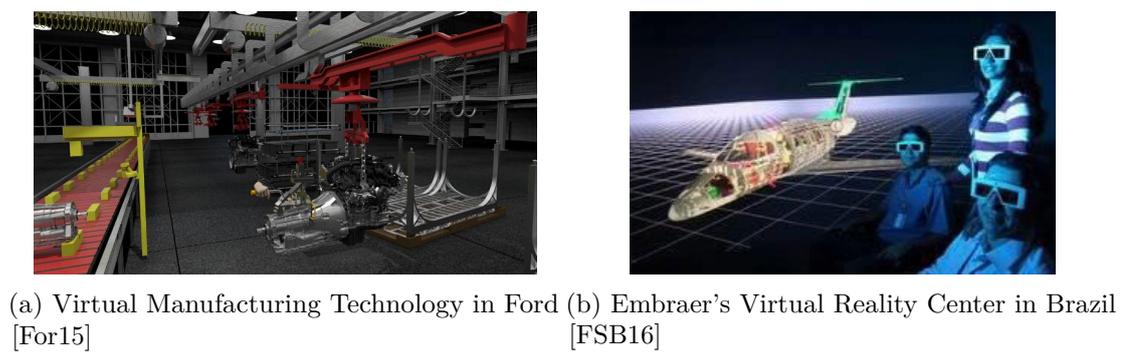
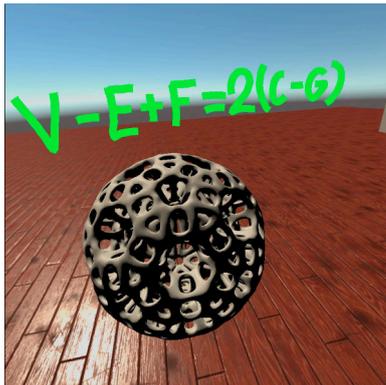


Figure 2.5: Examples for VR in the industry

in computer graphics. The system allows students to interact with 3D objects, annotate items and components, and observe the changes from various angles (Figure 2.6a). Virtual Reality also finds a place in medical training [GSB⁺18]. Using such technology can help professionals and students practice and determine their level of competence for medical treatment. In this field, it is essential to have the proper knowledge and enough practice before performing surgery on a patient. As Samadbeik et al. [GSB⁺18] show in their research, Virtual Reality proved to be helpful in improving the learning process and increasing the accuracy in medical practise (Figure 2.6b).



(a) VR framework for computer graphics education [HVV21] (b) Virtual Reality for surgery training [Vis17]

Figure 2.6: Examples for VR used for education and medical training

Rehabilitation and Mental Health Therapy

Another major utilization of VR can be found in mental health therapy and rehabilitation. It can help individuals deal with various medical conditions and disorders and improve their health. Such conditions include anxiety, pain management, stress and depression, eating disorders, addiction, and substance abuse [JKBWG18]. For instance, burnt patients experience excessive pain during medical procedures, such as burned wound dressing changes. VR is used in such cases as a distraction. Hoffman et al. [HRB⁺06] used a special software called SnowWorld (Figure 2.7a) that portrays an icy cold environment as a contrast to the unpleasant memories of warmth and burning the patient has. To investigate the effectiveness of their system, they used an fMRI to measure brain activity with and without the usage of VR. The results of their study show that pain-related brain activity is reduced when using VR (Figure 2.7b). Additionally, immersive environments are used to help people decrease or overcome different kinds of phobia such as fear of heights [FHF⁺18], fear of flights [RCGMBB13], or fear of spiders [MV19]. The treatment includes a gradual exposure to the problem with a controlled increase of the stimuli by the therapist. This treatment, however, only works if the patient feels present in the virtual world.

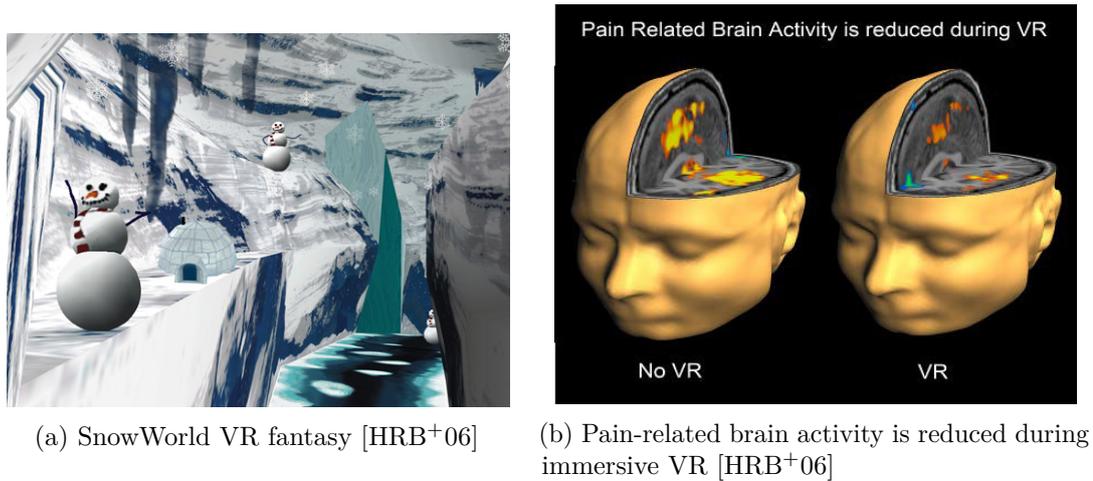


Figure 2.7: Examples for VR used for rehabilitation and mental health therapy

Modelling and Design

VR is also widely used for planning, modelling, and designing, especially in architecture. It not only provides virtual entities that represent real-world views, but also enhances collaboration among designer teams. Cooperation is particularly essential in the early design stages to identify the consequences of their design decisions and better understand the final results [PNFG15]. Construction planning and urban modelling are other areas where VR proves to be useful. By creating immersive construction simulations in VR, architects and designers can anticipate potential problems and improve delivery [DODB20] (Figure 2.8).

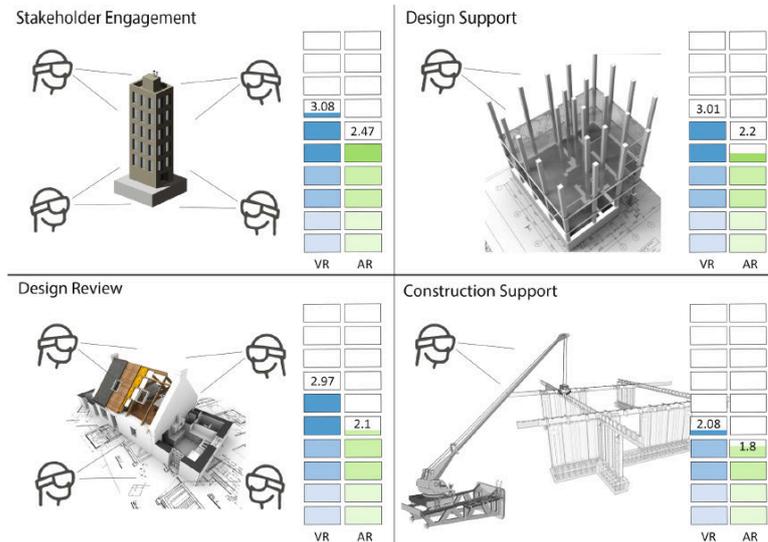


Figure 2.8: VR for architecture, modelling, design, and construction planning [DODB20]

Visualization

Another field where VR gains more and more popularity are different scientific visualizations that can support research and exploration. As early work demonstrates, using an immersive environment helps researchers to more efficiently investigate various scientific fields. Examples for such fields are shape perception [DJK⁺06], structural analysis of underground cave systems [RKSB12], brain tumors [ZDK⁺01], MRI [CCAL12], paleontology [LBS14], exploration of 3D trajectories [HHRD⁺18], coronary artery grafts [LPF⁺09], and many more. The following section will cover the usage of VR for scientific visualization in more detail.

2.2 Empirical Evidence for the Benefits of Virtual Reality

The different fields and applications of VR technology described in the previous section are one evidence for the benefits of an Immersive Environment (IE). As Bowman and McMahan [BM07] describe in their research, those different applications can be taken as success stories. According to the authors, the success lies in the fact that all of them require realistic experiences, precisely what VR systems provide. Realistic experiences include a high level of sensory fidelity (visual, auditory, and other sensory cues) to resemble the real world. For instance, in phobia therapy, the goal is to trigger the brain's fear structures activated by the patient's perceived sensory stimuli. VR technology can produce sufficiently realistic sensory stimuli to trick the patient into believing what he/she sees is real. In medical or military training, VR is also highly beneficial since the goal is to teach the trainee to perform the right actions when exposed to the same situation in real life. However, VR systems do not always need to resemble real-life scenes or situations entirely. They can go far beyond that, which is the case in VR game applications. Still, IEs are also advantageous in this field because such games aim to entertain and engage the user by offering novel situations that are impossible in the real world. VR technology offers additional sensory cues that are not available in most gaming systems. To summarize, many applications require specific sensory stimuli to provide a realistic experience and feel present in the simulated environment, which is possible with VR.

Another potential benefit that Bowman and McMahan mention is spatial understanding. Humans construct the 3D environment from 2D projections perceived by the eyes. The brain is highly optimized for reconstructing 3D scenes with the help of various depth cues such as stereopsis, motion parallax, perspective, and occlusions. VR technologies offer such depth cues in contrast to other technologies. With stereo images and head tracking, they provide motion parallax and stereopsis. This leads to higher immersion and, consequently, greater spatial understanding, prompting greater effectiveness in various applications. Furthermore, immersion might lead to a decrease in information clutter. An immersive environment offers more space than a traditional 2D display. With increased Field of View (FOV), Field of Regard (FOR) and display resolution, it is possible to decrease visual clutter and increase the comprehensibility of the environment.

Arns et al. [ACC99] also study the benefits of statistical visualization in an immersive environment. For this purpose, they conduct an experiment with an immersive environment (C2) based on the XGobi desktop system compared to XGobi itself. C2 is a projection-based immersive system that consists of three projected walls and a projected floor and is beneficial for an observation by multiple users. XGobi is a tool for visualizing and interacting with high-dimensional data by creating plots on a 2D display and works with conventional desktop devices such as keyboards and mice. The idea of the experiment was to identify the structure of the data, where three main structures were considered - clusters, intrinsic dimensionality, and radial sparseness. Additionally, the authors wanted to indicate which system is more natural and intuitive for interacting with the data. The experiment included 15 participants, who were given four different tasks regarding structure identification and interactions. The results show that the users could identify clusters better by using the immersive C2 system and perform better on the sphere test. According to Arns et al. VR offers a better understanding of the 3D topology due to the additional third dimension. On the other hand, users needed less time to interact with the traditional workstation running XGobi than with the C2 IE. However, the authors believe that previous experience with VR systems may have impacted the results since all of the users have used a 2D desktop system before, but only a few of them were familiar with VR systems. In conclusion, Arns et al. suggest that IEs have a high potential for scientific visualization and exploration, but they need to be further investigated.

Fonnet and Prie [FP21] do a more recent survey on immersive technologies for data analysis. In their survey, they review papers dating back from the year 1991 to 2018. They provide information about different aspects of immersive technologies such as sensory mapping, various interaction techniques, collaboration. The authors also review the possible benefits of using immersion for data analytics. Therefore, they discussed methods and evaluations of previously published research to test the feasibility and usefulness of their suggested system. Fonnet and Prie compare immersive and non-immersive systems by covering three types of data - temporal, spatio-temporal, and tree data. They also use two quantitative measures for the comparison - accuracy and completion time. The results show that a higher system fidelity (higher FOR, stereoscopy, or head-tracking) is beneficial in terms of accuracy for more complex data such as a graph or spatial data. The results are more uncertain regarding completion times since not all of the investigated papers cover that aspect. Also, there is a lack of information in system fidelity studies regarding data types such as temporal or spatio-temporal.

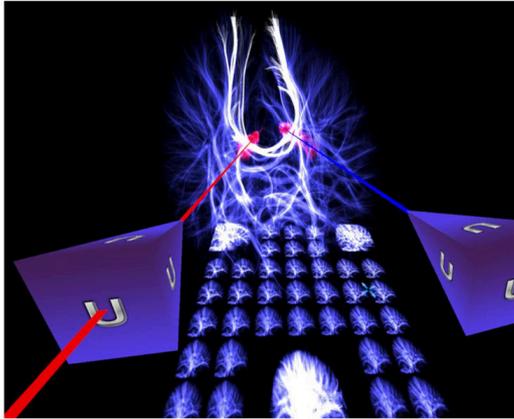
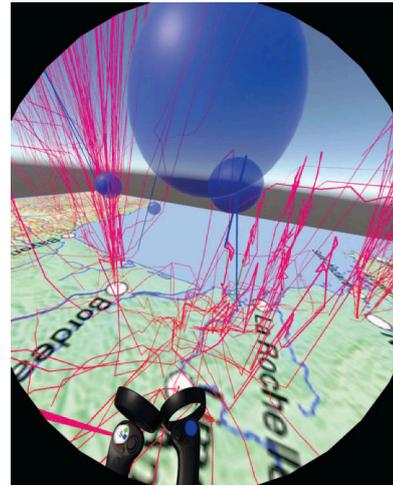
Other papers studying the possible benefits of VR systems are by Raja et al. [RBLN04], who investigate the benefits of immersion for abstract information visualization or Schuchard and Bowman [SB07], who examine the advantages of such systems for a spatial understanding of complex underground cave systems. Furthermore, Marriott et al. [MSD⁺18] provide a highly detailed review on immersive analytics and its many different aspects, including the potential benefits of immersive visualization.

2.3 Virtual Reality for Visualization and Immersive Exploration

As stated before, immersive visualization finds usage in many scientific fields and for various tasks. Hurter et al. [HHRD⁺18], for instance, develop FiberClay, a tool for visualizing and interacting with 3D trajectories (Figure 2.9a). Similar to that, Usher et al. [UKF⁺17] implement a visualization tool for neuron tracing, which is crucial for understanding the structure of the neural bundles in the brain and how information is processed to generate a different behaviour. In general, visualizing and tracing 3D trajectories could be highly useful for exploring the human brain's fibres bundles and, for instance, optimizing safety in aircraft routes. The general idea of FiberClay is to progressively refine a raw sculpting material using multiple interactions to produce the desired goal visualization, which is easier to comprehend. The data is, in general, represented as polylines in three-dimensional space, and FiberClay supports multiple trajectory representations - with lines, points, or animated particles. It also provides a color blending option to map data dimension to color, allowing the user to discern value changes. The system supports the standard navigation options - translation, rotation, and scaling using the controllers. Additionally, the system introduces 3D brushes used to add or remove trajectories or select a cluster. Hurter et al. [HHRD⁺18] conduct a qualitative evaluation with participants who used their system to investigate its benefits and effectiveness. The results showed that by using FiberClay new and unique three-dimensional patterns were identified that were previously not detected.

A more recent system in VR for exploring and filtering large trajectory-based datasets is called ReViVD by Homps et al. [HBV20]. The system provides the users with simple 3D shapes (spheres, cuboids, cylinders), which can be used to select and filter out groups of trajectories (Figure 2.9b). These shapes are called selectors, and the user can scale, rotate, and translate them. When a selector is active, all visible ribbons/trajectories touching it will take the color of the selector. ReViVD can be used for different application domains, such as GPS position tracking, traffic simulation, or turbulent particle flows. An evaluation of the system showed that it is particularly useful for refining selections progressively. Additionally, ReViVD proved to be helpful in gaining an overview of the dataset and easily extract trajectories or regions of interest.

Donalek et al. [DDC⁺14] investigate the usage of an immersive VR system as a general-purpose tool for exploring various types of abstract, multi-dimensional data. Their work examines the advantages of such a system compared to traditional visualization tools using desktop and 2D input devices. As a start, they experimented with an OpenSim-based virtual world using different test datasets and various approaches for encoding the multi-dimensional data, such as colors, sizes, transparencies. Build upon that, Donalek et al. create a Unity 3D based prototype data visualizer, called iViz (Figure 2.10a). The prototype can visualize up to 10^6 data points and allows the user to easily choose and randomly shuffle how the data attributes are mapped to each graphical axis. This way, the most optimal mapping could be found according to the scientific application, offering

(a) FiberClay [HHRD⁺18]

(b) ReViVD [HBV20]

Figure 2.9: Examples for visualizations of 3D trajectories

more adaptivity and powerful visual data exploration. Furthermore, iViz supports a broadcasting function so that multiple users can explore the data together in a shared view, which provides a more flexible and collaborative data examination. To analyze the advantages of such an immersive system, Donalek et al. conducted an experiment with geologists using terrain data from Mars. The scientists were split into two groups. The first group was shown a 2D panorama image stitched together from multiple snapshots, and the second group was given a 3D stereoscopic image. Both groups were given a map-drawing task. The results showed that the group using the immersive environment performed better in measuring distances and orientating in the field since VR allowed them to get a better insight into the terrain by having the ability to move in the space.

Another immersive system for exploring multivariate data, called ImAxes, is created by Cordeil et al. [CCD⁺17]. The system uses a simple grammar defining the rules for constructing visualizations in VR environments, which has the axis as a basic grammar element. In general, it allows users the creation of various data visualizations by directly combining and manipulating data axes in 3D space. Different plots can be created depending on the number of axes, their proximity, and relative orientation to each other. These include standard InfoVis visualizations such as histograms, scatterplots, parallel coordinates plots. Examples of such visualizations created with ImAxes are shown in Figure 2.10b. However, ImAxes also proves to be useful in creating new, but helpful visualizations such as 3D circular connected parallel coordinates plots. Cordeil et al. show that their system gives the user the freedom to explore data by creating familiar and novel visualization techniques quickly and efficiently. They demonstrate the effectiveness of ImAxes as a multivariate data visualization tool in VR by creating and evaluating a scenario using a wine dataset.

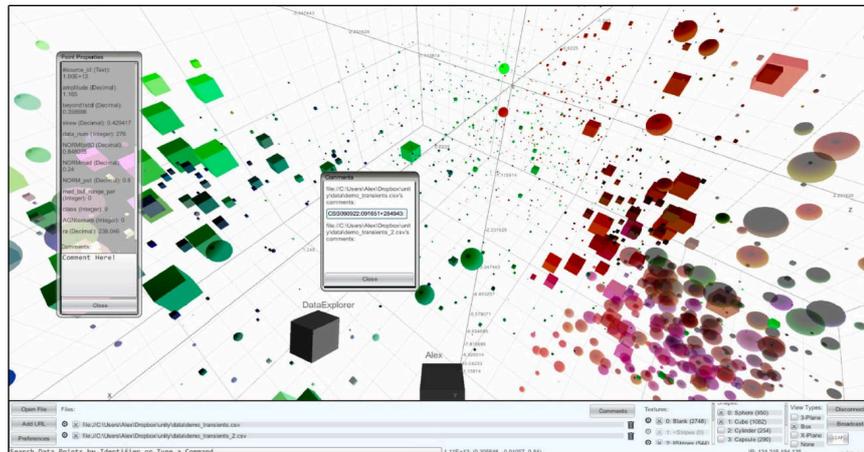
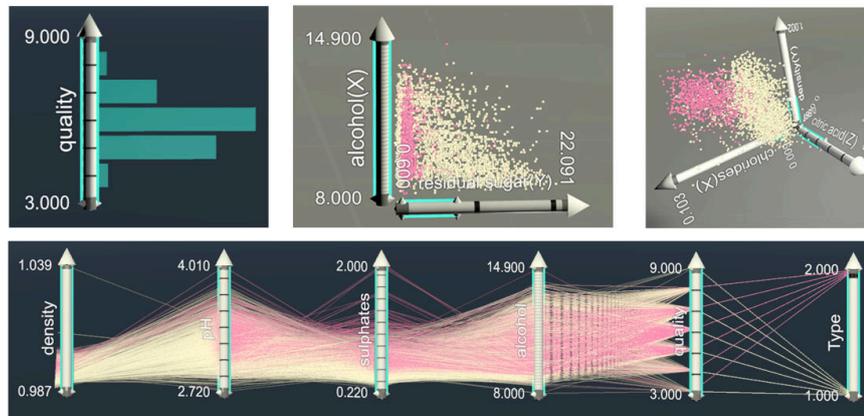
(a) iViz [DDC⁺14](b) ImAxes [CCD⁺17]

Figure 2.10: Multivariate data visualization systems

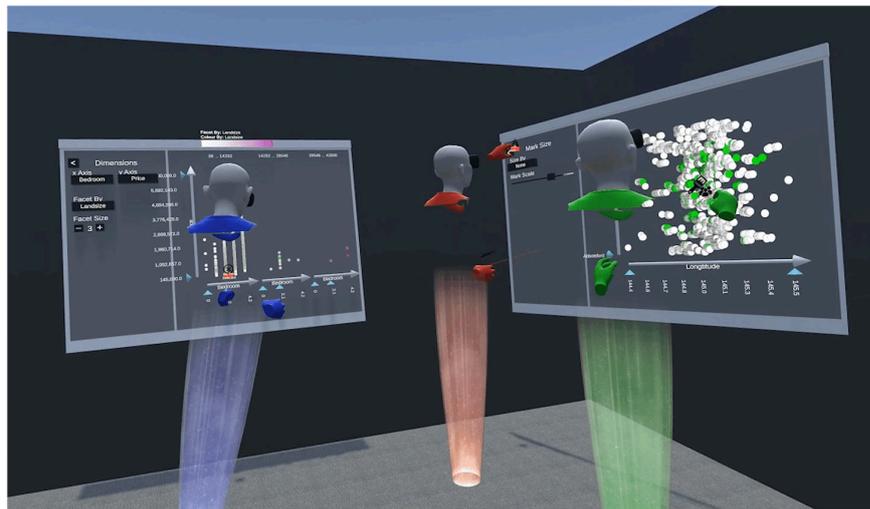
Collaboration is another major factor and is often required when dealing with data visualization and exploration. Immersive systems proved to be helpful and offer new opportunities to support collaborative data analysis and research. Cavallo et al. [CDH⁺19] investigate the advantages of using a hybrid analytic system for collaborative data exploration with the help of VR and AR. Their system, Immersive Insights, combines several high-resolution displays, table projections, and augmented reality visualizations of the given data to allow a simultaneous data exploration by multiple users, as shown in Figure 2.11a. Their goal was to study the effects of different levels of immersion on the exploratory data analysis process compared to standard non-immersive data analysis solutions. To evaluate the system, they conducted a user study in two parts by comparing different modalities of the Immersive Insights in the first part and comparing their hybrid reality system to a desktop-based program in the second part. The results showed that, in general, immersive technologies helped reduce the analysis time and facilitate the

2. RELATED WORK

generation of additional insights. However, they cannot entirely replace non-immersive technologies, but rather complement and improve them.



(a) Immersive Insights [CDH⁺19]



(b) FIESTA [LHC⁺21]

Figure 2.11: Collaborative data visualization systems

FIESTA (Free-roaming Immersive Environment to Support Team-based Analysis) is another collaborative data visualization system by Lee et al. [LHC⁺21], which compared to Immersive Insights is entirely in VR. Each user is provided with a head-mounted display to view the shared visualization workspace to support the collaboration. The users can solve visual analytics tasks by creating a large number of 2D and 3D visualizations and freely placing them on various authoring interfaces like touch panel displays and virtual surfaces. An image of the system can be seen in Figure 2.11b. For the interaction techniques, Lee et al. adapt standard 3D UI metaphors to reduce learning time. For objects in proximity, they use close-range grasping techniques, and for the more distant ones, long-range pointing techniques with a laser pointer. The system additionally

provides pointer, brushing, and annotation tools and the option to clone visualizations. To support collaboration, the users can see each other as virtual avatars at their real-world positions. Furthermore, there is no explicit ownership in FIESTA since everything is visible to all users at any time, and each of the users can interact with any object in the environment. Lee et al. conduct a user study for a given dataset by assigning the users directed tasks in the first session. Then in the next session, they were given a free exploration task to analyze the same dataset and find interesting insights about the data. Additionally, the participants were split into two groups - group A, where participants were limited to create only 2D visualizations and Group B, where the users could also create 3D visualizations.

Hentschel et al. [HWK09] concentrate on the visualization and interactive exploration of large time-dependent simulation data in an immersive environment. As basic techniques for the analysis, they incorporate multiple linked views and brushing. For the data visualization, they use three-dimensional linked scatterplots so that updates of a single plot will be applied to all other plots. The time-dependent aspect of the data is handled by animating the plots. For filtering data out and highlighting a set of selected points, the authors use a brushing operation with a rectangular selection box (Figure 2.12). Visualizing and animating such time-dependent simulation data is a demanding task in terms of computing power and time. For this reason, Hentschel et al. use an additional parallel machine to compute the intensive operations and provide near real-time updates.

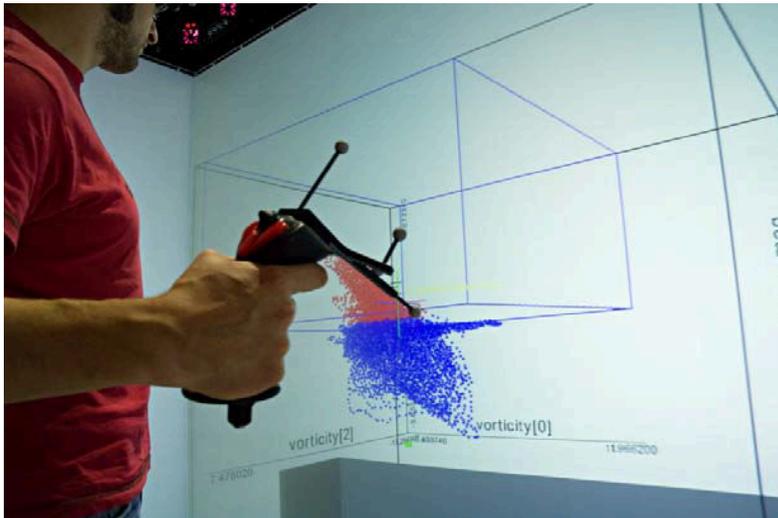


Figure 2.12: Filtering time-dependent simulation data with a selection box in VR [HWK09]

There are also various other fields where visualization in an immersive environment is advantageous. For instance, Laha et al. [LBS14] investigate the effects of VR systems for visualizing and analyzing isosurfaces of volume datasets. For this purpose, they visualize isosurfaces using a CAVE-like display. They evaluated the system by comparing the

effects of three components concerning system fidelity (field of regard, stereoscopy, and head tracking) and compared the results using traditional 3D-texture based rendering. The results showed that a stereoscopic display improves performance for search and spatial judgment tasks, while displays with a higher field of regard are more effective for tasks with 3D-texture based rendering. Zhang et al. [ZDK⁺01] also incorporate an immersive environment for the volume visualization of DT-MRI data. They display geometric representations of the volumetric second-order diffusion tensor brain data within the CAVE virtual room. According to their results, using an immersive system compared to a standard desktop display helps them better interpret large and complex data, achieve a better sense of the structures, and simplify communication between a group of users. Additionally, VR is also often used for point cloud visualization, exploration and rendering [AYE20][DMS⁺18][MJPR16], as well as 3D point cloud labeling and annotation [SSCG17][WQOS19].

2.4 Virtual Reality Toolkits and Frameworks for Creating Visualizations

There are various frameworks and toolkits (VR-Viz [Sai18], DXR [SLC⁺18], IATK [CCB⁺19]) that support the exploration in an immersive environment and faster development. They provide multiple possibilities for the creation of 3D visualizations, plots, and graphs in 3D space.

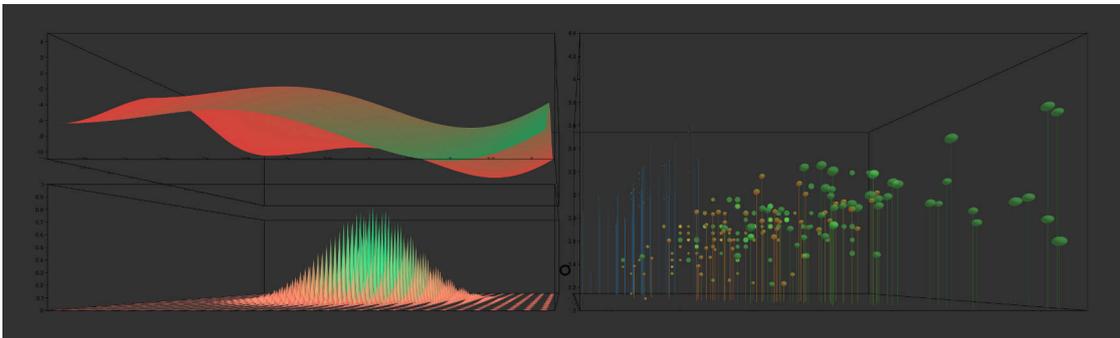


Figure 2.13: Visualizations created with the VR-Viz framework [Sai18]

VR-Viz

VR-Viz is a visualization system for Virtual Reality by Mustafa Saifee [Sai18] that provides high-level react components used to generate 3D visualizations in webVR. The system incorporates the A-Frame framework for building a VR scene, the React library for the DOM manipulation, and the D3 library to create visualizations in VR. VR-Viz is strongly based on Vega-Lite [SMWH17], and for the formal definitions of the visualizations, the author used the JSON syntax. The system provides 24 different graphs and charts, varying from standard plots (e.g. bar charts, scatterplots) to more

specific ones like function plots or visualization graphs for geospatial data. VR-Viz concentrates mainly on the representation component, which is why it provides only basic interactions. It incorporates two interaction techniques - explore and select/focus. The explore interaction is used to get insight into the data and is provided by the VR system itself. The user can rotate the HMD camera and walk-in/out of the data to observe the environment from various locations and angles. For the select/focus interaction, VR-Viz uses a mouse over effect to select or focus on data of interest. Various plots generated with VR-Viz can be seen in Figure 2.13.

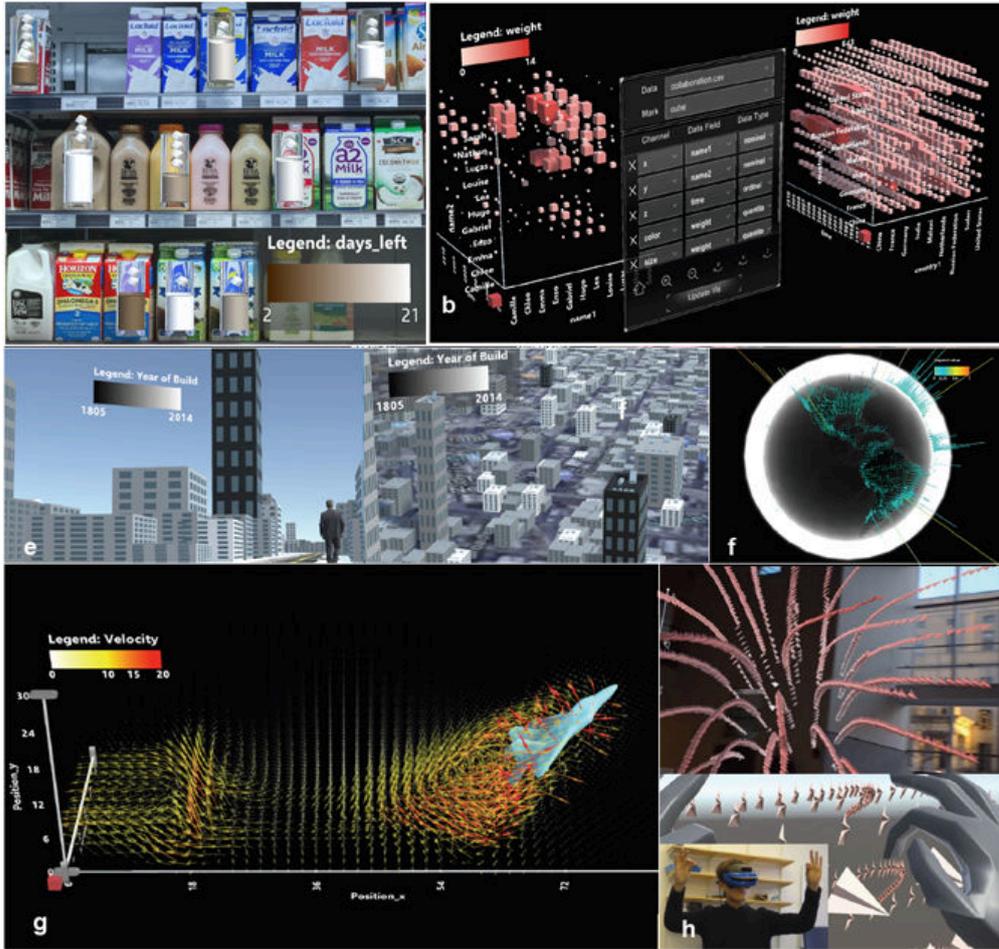


Figure 2.14: Different visualizations created with the DXR toolkit [SLC⁺18]

DXR

Another toolkit for building data visualizations in VR is DXR by Sicat et al. [SLC⁺18], which is based on the Unity development platform. The system provides a concise declarative visualization grammar inspired by Vega-Lite to specify and create different visualization designs. The idea of using a grammar similar to Vega-Lite is to make the

2. RELATED WORK

representations more intuitive and easier to learn and modify. To construct a GameObject in a Unity scene, DXR uses a high-level interface consisting of prefabs and scripts. This way, visualizations (represented as GameObject prefabs - vis-prefabs) can be added to the scene and manipulated via scripting. The visualization specification file is in the JSON format. Similar to the notation of Vega-Lite's "unit", this file describes a single visualization in DXR by encoding and defining different properties (data, mark-type, visual channels like position, color size, and others). Sicat et al. discovered that the JSON syntax and the grammar-based specification could be overwhelming for non-programmers. Consequently, they provide an in-situ graphical user interface allowing the users to change different properties of the visualization directly in VR. DXR supports the creation of very diverse visualizations for multiple use-cases, varying between embedded representations, immersive workspaces, standard 2D and 3D visualizations, 3D flow fields and more. Examples of visualizations created with DXR are shown in Figure 2.14.

IATK

IATK is another toolkit for immersive analytics, which is developed by Cordeil et al. [CCB⁺19]. It provides an abstracted data model and a simple and composable grammar that allows the creation of multiple different visualization components. IATK supports 1D, 2D, and 3D orthogonal axes and orthogonal grids. It allows the creation of geometric objects via the Unity GUI, such as points, lines, and bars with varying sizes and colors depending on the dimension of the data. The toolkit also supports visual linking so that the same data points could be connected across different visualizations. Additionally, for efficient visual data exploration, IATK introduces multiple interaction possibilities. It supports filtering, brushing and linking, details on demand, and various animations such as animated transitions. IATK provides a high-level interface in Unity and an expressive API for the rapid construction of novel visualizations in immersive environments.

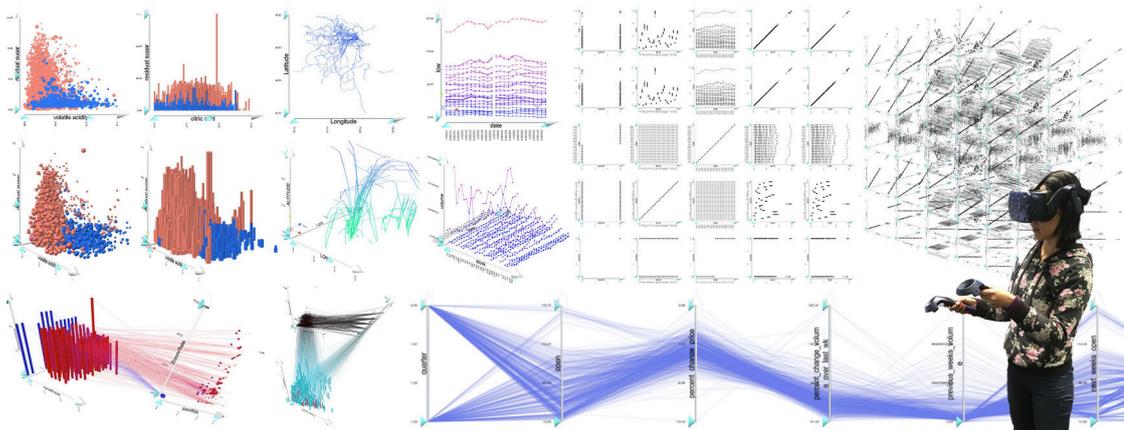


Figure 2.15: IATK (Immersive analytics toolkit) [CCB⁺19]

VRIA

A more recent framework for creating immersive analytics experiences in VR is called VRIA by Butcher et al. [BJR21]. VRIA is a web-based framework and, similarly to VR-Viz, is built upon A-Frame, React and D3. Like DXR and VR-Viz, VRIA is also based on Vega-Lite and supports tabular data formatted as JSON or CSV. The visualization configuration file also consists of a dataset, a graphical mark, and a set of visual encoding channels for each data point to modify different properties. The visualization types that VRIA currently supports are Cartesian plots such as bar graphs and scatterplots. The system is intended for both non-WebVR and WebVR developers since the former are more likely to have used JSON before and thus are more prone to edit configuration files directly. The latter, on the other hand, can take advantage of VRIA's API to create visualizations. Using the `vis config` the user can enable or disable different interaction types allowing him to do selection, filtering, details on demand, and brushing and linking. Moreover, using various constraint combinations, multi-dimensional queries could also be created. VRIA also supports the generation of collaborative environments where users can simultaneously create and interact with visualizations. Figure 2.16 shows example visualizations created with VRIA. Butcher et al. compare all four most recent immersive analytics toolkits in their paper in terms of the features they offer. An image of that comparison table can be seen in Figure 2.17.

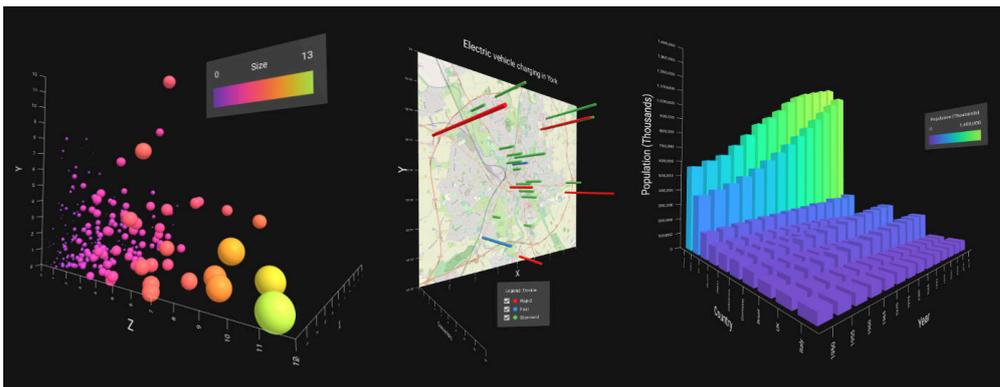


Figure 2.16: Visualizations created with VRIA [BJR21]

2.5 Perception, VR Sickness, and Performance in Virtual Reality

The key elements to consider when developing VR applications are performance, perception, and possible motion sickness. Despite the advancements in VR technologies, it is common that users still experience sickness symptoms when being in a virtual environment [SSB⁺20]. VR sickness is known by several names, including cybersickness, motion sickness, and simulator sickness. Although they refer to a similar condition, they are not entirely synonymous. Motion sickness is associated with adverse symptoms and

Feature	IATK	DXR	VR-Viz	<VRIA>
Platform	Unity	Unity	Web	Web
Displays	VR/AR	VR/AR	VR	VR/AR
Dimensionality	1/2/3D	1/2/3D	3D	1/2/3D
Interactions	✓	✓	✓	✓
Linked Views	✓	✓		✓
Custom Marks	✓	✓		✓
Code Playground	◇	◇		✓
GUI	✓	✓		✓
Declarative Config ⁽¹⁾		✓	✓	✓
API	✓			✓

◇ other software is responsible for providing that feature

Figure 2.17: Comparison between the Immersive Analytics Toolkits [BJR21]

readily observable signs caused by exposure to real (physical or visual) and/or apparent motion [Gro14]. Cybersickness, on the other hand, differs from motion sickness in the sense that the user is often not moving physically, but feels like she is due to moving visual imagery [LaV00]. Simulation sickness is a sickness not caused by the actual situation being simulated, but rather from shortcomings of the simulation [PCC92]. Since there is no term that covers all the described types of sicknesses, in this work we will apply the term VR sickness. VR sickness includes various symptoms such as nausea, headaches, and dizziness.

One of the significant factors influencing the emergence of VR sickness are the technical parameters of the system. The resolution, frame rate, jitter, and latency have a significant impact on the feeling of presence and reducing sickness [LTRCB19]. A high enough frame rate is essential in order to "trick" the user into believing that what he/she is experiencing is real and thus reduces the possibility of inducing VR sickness [Iri14]. Additionally, high-resolution rendering should yield more presence than a pixelated rendering. Jitter means having timing variations of a set of signal edges deviating from their ideal values. In terms of a VR system, having high jitter is related to a higher probability of getting cybersickness, which is why it should be reduced to as little as possible [SNL18]. Having an overall low latency of the system is also especially important for achieving and maintaining a good user experience [GGH⁺18]. These four parameters do not have the same impact on presence in VR. As Thibault et al. [LTRCB19] show in their research, jitter and frame rate play a more crucial role for presence than latency, and the resolution depends mainly on the explored entity.

Despite the technical parameters influencing the emergence of VR sickness, many other factors have been researched and impact the occurrence of VR sickness symptoms. The most common way used in cybersickness studies to investigate the possible factors is the Simulator Sickness Questionnaire (SSQ). The SSQ was developed by Kennedy et al. [KLBL93] in 1993 and consists of several questions regarding the severity of symptoms on the scale of 0-3 (0 = not at all, 1 = slightly, 2 = moderately, 3 = very). The scores are computed for three categories of symptoms - nausea (e.g. general discomfort, sweating),

oculomotor (e.g. fatigue, headache, eyestrain), and disorientation (e.g. difficulty to focus, dizziness, vertigo) [RO16]. Each category has its own weighing, and a computed sum score below 10-15 is considered normal. According to Kim et al. [KPCC18] and their results of the SSQ, disorientation has the highest score, followed by the oculomotor, and the lowest score is by nausea (Figure 2.18). This means that users most frequently experience symptoms in the disorientation category.

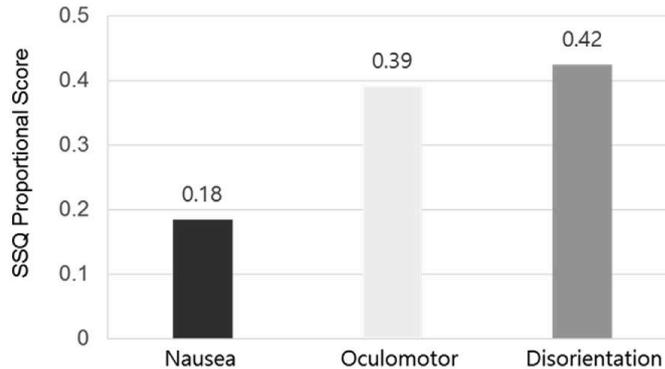


Figure 2.18: SSQ profiles for a VR device [KPCC18]

Guna et al. [GGH⁺18] investigate the influence of video content type and users' background preferences on VR sickness perception, as well as their physiological response. For this purpose, they use the standard SSQ and also measure several physiological parameters such as heart rate, breathing rate, skin temperature and skin conductance response. Their study uses two types of video content - neutral and action content, which is displayed in five different systems and machines - a 2D TV set, three generations of Oculus Rift VR HMDs and the mobile version Samsung Gear VR. Guna et al. also take the personal preferences (e.g. adrenaline and extreme sport preferences) of the participants into consideration for the influence on VR sickness. Previous research showed that, for instance, a PC gamer is less prone to get VR sickness symptoms and is more immersed compared to a non-gamer [RMG⁺16]. The authors conduct a user study with 26 participants with various backgrounds and different experience levels with VR devices. The results of the SSQ showed that, indeed, action video content had a higher effect in inducing VR sickness to the participants than a neutral video. The same results were confirmed by the physiological measures. They show that the parameters did not differ significantly from the baseline level during the neutral video clip, indicating a higher relaxation level. Additionally, the experiments confirm that participants with higher preferences for adrenaline sports experience significantly less VR sickness and discomfort.

A more extensive and novel study about the factors associated with VR sickness is conducted by Dimitrios et al. [SSB⁺20]. In their research, they examine previous literature on HMDs reporting SSQ scores about factors that impact VR sickness. For this purpose, they review 55 papers in total, which were selected after a screening process. The factors that the authors examine include content, visual stimulation, locomotion, and time. The content is split into four different categories (360° videos, gaming content,

minimalist content, and scenic content), where in each category, the interactions and environmental features differ. Visual stimulation refers to the amount of visual movement within the content and is classified as low (content with slow visual changes) or high (content with fast visual changes). Locomotion is related to how the user navigates through the virtual world, classified into three categories - stationary, controller-based moving or walking. Previous research shows that physical walking decreases the chance of getting VR sickness compared to navigation happening through the controller [CGBL98]. The last factor, the time, is split into three categories ($< 10min$, $\geq 10min$, $\geq 20min$). Intervals of 10 min were taken because, in earlier research, it has been found that VR sickness increases after 10 minutes [MM11]. Considering the given factors, the findings of the study by Dimitrios et al. show that in most of the papers, gaming content recorded the highest SSQ mean. The locomotion type also highly influences the SSQ score by displaying the highest total SSQ mean for the controller-based movement. Furthermore, the review shows that visual stimulation, as well as exposure time, also have an impact on VR sickness.

In order to reduce the probability of inducing VR sickness to users, the factors mentioned above should be taken into consideration. Additionally, Patrão et al. [PPM15] present some assumptions for reducing VR sickness according to their research. Firstly, they suggest keeping the users immersed the whole time during the experience, meaning to always respond to the user input and avoid the feeling that the system is not responding. Secondly, adding a frame reference may help to prevent the users from feeling not well. Last but not least, the authors suggest to "treat VR as a theatre, not as a film", meaning that some camera tricks (such as zoom in and out) should be reduced to a minimum.

Another important factor to consider when developing a VR application is the depth perception. Armbrüster et al. [AWK⁺08] present a study that examines the depth perception in virtual environments and what factors might influence it. They conducted a user study with 23 participants, who had to verbally estimate the distance of objects in an IE. The authors experimented with three types of virtual environments. The first one is called "no space", meaning that the environment was infinite, blue, with no additional depth information. The second type was "open", where the system provides a green floor and a blue sky, and the third one illustrates a closed gray room, where linear perspective was used as a secondary depth cue. The objects to be estimated were placed at ten distances between 40 cm and 500 cm in two conditions - either only one object was displayed or all ten were shown simultaneously. Distances of objects placed under 150 cm are regarded as peripersonal distances and between 150 cm and 500 cm as extrapersonal distances. Additionally, the presence of a metric aid was varied in the scene by placing a tape in front of the participants on the ground. Furthermore, the influence of sex, hyperopia, myopia, and binocular ability were analyzed as possible factors having an impact on depth perception. Finally, the participants had to fill in a questionnaire about physical complaints, experience in VR, and self-evaluations of their estimations. The results indicate that neither of the three virtual environments improved depth estimations. Furthermore, the presence of a tape measure did not have a significant influence on the

estimations in any of the environments. The results also show that depth estimation performance is better in peripersonal compared to extrapersonal space. The subjective ratings indicate that only few of the participants experience discomfort symptoms and they were either neutral or were highly fascinated by the virtual experience. However, different to the results, the subjective ratings show that participants feel to be more accurate in the closed space environment.

2.6 Interactions and Metaphors in Virtual Reality

Interaction techniques are essential for accomplishing various tasks in the virtual space and manipulating objects, and interacting with the environment. In their book *3D User Interfaces*, LaViola et al. [LKM⁺17] describe the different 3D interaction techniques by classifying them depending on the user interaction task. The following sections will give an overview of the interaction techniques depending on the task.

2.6.1 Object Selection and Manipulation

The first type of interaction tasks can be categorized as selection and manipulation. This category includes tasks as grabbing, moving, orienting, and scaling objects. 3D manipulation tasks can be further classified according to their isomorphism. The isomorphic view represents a "strict, geometrical, one-to-one correspondence between hand motions in the physical and virtual worlds" [LKM⁺17], which is more natural and thus better for the users. Nonisomorphic approaches, on the other hand, allow users to manipulate objects differently than in the real world by introducing "magic" virtual tools such as laser beams, voodoo dolls, and others while still maintaining performance and usability. The 3D manipulation techniques are also classified by metaphor, forming a fundamental mental model of a technique. LaViola et al. classify the metaphors for 3D manipulation in six different categories:

- **Grasping Metaphors** One of the most natural and intuitive techniques for grasping and manipulating objects is with one's hand. Two distinct approaches support objects grasping - hand-based grasping and finger-based grasping.
 - **Hand-based grasping** - The hand-based grasping is the most common and simple approach since the hand is observed as a single-point effector without tracking each of the fingers. Hand-based grasping can be further divided into two techniques - simple virtual hand and Go-Go technique. The Go-Go technique extends the simple virtual hand, allowing the user to change the length of the virtual arm and thus access and manipulate remote objects that are otherwise out of the grasping range.
 - **Finger-based grasping** - The finger-based grasping techniques are more realistic, but also more complex since the positions of the fingers are also considered for grasping. LaViola et al. describe three types of finger-based techniques: rigid-body fingers, soft-body fingers, and god fingers.

There are also some enhancements for the grasping metaphors, such as 3D bubble cursor or PRISM, allowing to employ more complex transfer functions and interaction logic to increase the usability and efficiency of the grasping techniques.

- **Pointing Metaphors** The idea behind pointing metaphors is to enable the user to select and manipulate distant objects by simply pointing at them. If the direction of pointing intersects a virtual object, the user can confirm the selection. There are two main types of pointing techniques - vector-based and volume-based.
 - **Vector-based pointing techniques** - These techniques are rather easy to implement and commonly used for pointing in 3D systems. They require only a vector to compute the object that the user intends to select and manipulate. Examples of vector-based pointing techniques are ray-casting, fishing reel, and image-plane pointing.
 - **Volume-based pointing techniques** - This techniques require both a vector and a volume to determine what the user intends to select and manipulate. Typically, the volume stays in some relation to the vector. When having a cone, for instance, the vector defines the axis. Examples of volume-based pointing techniques are flashlight, aperture selection, and sphere-casting.

There are also enhancements for the pointing metaphors such as bendcast or depth ray. The main focus of these enhancements is to make it easier to distinguish between multiple objects and to achieve a higher level of precision.

- **Surface Metaphors** Surface metaphors refer to using touch gestures to interact with virtual objects directly. It is most commonly used for 2D surfaces, but multi-touch surfaces can also be used for 3D interactions.
 - **Surface-Based 2D Interaction Techniques** - The most common interaction techniques with 2D context are dragging and rotating. By dragging, the user can select and translate an object directly by touching it with one or more fingers and then perform a sliding operation.
 - **Surface-Based 3D Interaction Techniques** - Having a third dimension makes the interaction more complex and not so intuitive since the users also have to control the depth of the virtual objects relative to the surface. Some techniques for surface-based 3D interaction are pinching, void shadows, and balloon selection. The pinching, for instance, is used to scale (shrink or enlarge) a virtual object by pinching the fingers so that the distance between two contact points determines the new size.
- **Indirect Metaphors** As the name implies, indirect metaphors allow the users to manipulate virtual objects without directly touching or interact with them.
 - **Indirect Control-Space Techniques** - These techniques allow the user to interact with the virtual environment by using an additional physical space

- that is distinct from the real location of the virtual environment. Examples are a virtual pad, a virtual interaction surface, or a levels-of-precision cursor.
- **Indirect Proxy Techniques** - With these techniques users can interact with proxies directly. The proxies represent copies of objects within the virtual space. The interactions and manipulations applied to the proxy are then mapped to the original objects. Examples are a voodoo doll or a WIM (world-in-miniature), providing an exact copy of the virtual world at a smaller scale.
 - **Indirect Widget Techniques** - These techniques include the usage of widgets that the user can directly manipulate, which will then indirectly affect other virtual objects. Examples of such techniques are the virtual sphere and the arcball.
- **Bimanual Metaphors** The bimanual metaphors are a type of interaction techniques where both hands/controllers are required for interactions.
 - **Symmetric Bimanual Techniques** - In the symmetric techniques, both hands are performing the same actions. Examples of such techniques are spindle and iSith [WBB06]. With this technique, both controllers are used to form a virtual spindle that extends from one controller position to the other. The spindle centre is the primary point of interaction, used to translate an object and simultaneously rotate the yaw and roll of an object.
 - **Asymmetric Bimanual Techniques** - In asymmetric interactions, both hands are performing different actions to accomplish an interaction, where the movement of both hands can happen at the same time or at different times. Examples for this category are Spindle + Wheel and flexible pointer. The traditional spindle technique is extended by a virtual wheel used to rotate also the pitch of the selected virtual objects.
 - **Hybrid Metaphors** Sometimes there is no single best 3D manipulation technique to cover all needed and possible interaction scenarios for a given task/system. Therefore, combining several interaction techniques can provide more flexibility and bring together the best properties.
 - **Aggregated Hybrid Techniques** - The user is provided with a manual or automatic tool for selecting the desired manipulation techniques from a limited set of possible options. For example, using a 3D menu can display a virtual toolbox to choose the desired manipulation technique.
 - **Integrated Hybrid Techniques** - In such a technique, the system transparently switches between different interaction techniques depending on the context of the current task. A famous integrated hybrid technique is called HOMER (hand-centred object manipulation extending ray-casting). With this technique, the user can select an object through ray-casting, and then the user's virtual hand automatically moves to the object and is attached to it.

Afterwards, the manipulation mode is activated, and the user can alter the position and rotation of the virtual object.

2.6.2 Navigation

Navigation is another crucial aspect of virtual reality systems. Navigation can be split into two components - travel and wayfinding. Travel represents the motor aspect, meaning the task of moving from one location to the other. On the other hand, wayfinding can be observed as a cognitive process, where planning and decision-making play a role in finding a final desired destination. The reasoning behind travel can be classified into three tasks - exploration (no specific goal), search (travelling to a specific goal or location) and manoeuvring (involving small, precise movements to perform a specific task). The 3D travel can also be classified according to different taxonomies, each covering a different aspect of travelling:

- **Active vs. Passive** - In an active travel technique, the movement of the viewpoint is directly controlled by the user. In a passive technique, on the other hand, the system controls the viewpoint's movement.
- **Physical vs. Virtual** - In a physical travel technique, a user's physical body translates or rotates in order to adjust the viewpoint. In a virtual technique, on the other hand, even though the virtual viewpoint is being changed, the physical body of the user stays primarily stationary.
- **Using task decomposition** - The task of travelling can be decomposed into three subtasks. These tasks are direction or target selection, velocity/acceleration selection, and conditions of input.
- **Classification by metaphor**
 - **Walking Metaphors** - Physical walking is the most natural technique for travelling in a 3D world. Due to occlusions and restrictions, it is, however, not always possible. Therefore, there are various types of walking metaphors, split into three main categories:
 - * **Full Gait Techniques** - This category represents the real physical walking involving all biomechanics of a full gait cycle. Examples of such techniques are real walking, redirected walking, and scaled walking. In the redirected walking, for instance, the user moves along predefined paths in the real world, which differ from the paths perceived in the virtual world to overcome the limitations of real walking.
 - * **Partial Gait Techniques** - In these techniques, the user still moves in some way, but the movement is more restricted and includes only specific aspects of the human gait cycle. Walking in place and a human joystick are examples of partial gait techniques.

- * **Gait Negation Techniques** - This walking metaphor takes advantage of special locomotion devices to provide a somewhat realistic walking while the body is not actually translated to another location. Different types of treadmills, low-friction surfaces, or step-based devices are some examples of gait negation techniques.
- **Steering Metaphors** - steering metaphors refer to the continuous control of the moving direction. it could either be an absolute or relative direction of motion.
 - * **Spatial Steering Techniques** - In these techniques, the movement of travel is controlled by manipulating the orientation of a tracking device. Depending on the tracking device, these techniques can be classified as gaze-directed steering (depends on the looking direction), hand-directed steering (depends on the user's hand orientation), torso-directed steering (depends on the direction of the user's torso), and lean-directed steering (depends on the leaning direction of the user's body).
 - * **Physical Steering Props** - In these techniques, specialized physical devices are used. Examples of such props are cockpits or cycles.
- **Selection-Based Travel Metaphors** - This category of metaphors describes techniques for travelling by either selecting the desired target to reach or a specific path to travel along.
 - * **Target-Based Travel Techniques** - In these techniques, the main goal is the final target to be reached and not the actual path to get to it. For this purpose, a 2D map or a 3D world-in-miniature can be used to specify the desired target location.
 - * **Route-Planning Travel Techniques** - The goal of such a technique is to go along a specific path, defined by the user. There are two possible ways to achieve that - either draw a path or mark points along a path.
- **Manipulation-Based Travel Metaphors** - In these techniques, the users typically apply hand-based object manipulation metaphors such as Go-Go or HOMER in order to manipulate the viewpoint or the entire world. Such techniques are beneficial if both travel and manipulation techniques are required in the system.
 - * **Viewpoint Manipulation Techniques** - There are various ways to achieve travelling through manipulating the viewpoint. For this purpose, the user can perform camera manipulation, avatar manipulation, or fixed-object manipulation.
 - * **World Manipulation Techniques** - The second type of manipulation-based travelling is achieved by modifying the entire world relative to the current viewpoint. There are two ways to accomplish this, categorized by the number of points used for the manipulation - single-point world manipulation or dual-point world manipulation.

2.6.3 System Control

Except for the selection and manipulation of objects and navigation, interaction techniques are essential for controlling the system. The idea behind is to use various commands to change the system state and request the system to perform a particular function or change the type of interaction. According to LaViola et al. [LKM⁺17] techniques for system control can be classified into six different categories:

- **Physical controllers** - The physical controllers are devices that provide buttons and switches for changing the state or mode in an application on a desktop system.
- **Graphical menus** - Graphical menus in 3D space can be divided into three categories - adopted 2D menus, 1-DOF menus, and 3D widgets. The 2D menus are floating 2D planes in the 3D space with different options and settings to select. 1-DOF menus represent a one-dimensional operation for selecting an item. Very often, these menus are attached to the user's hand, arranged in a circular pattern.
- **Voice commands** - Voice commands is another technique for manipulating the state of the system. It can be performed by either simple speech recognition or through spoken dialogue techniques.
- **Gestural commands** - The system control with gestural commands is achieved by performing different gestures or postures. The gesture commands can be further split into seven categories - mimic gestures, symbolic gestures, sweeping, sign language, speech-connected hand gestures, surface-based gestures, whole-body interaction.
- **Tools** - Tools are real-world devices for 3D interactions that have an equivalent virtual representation. These tools are usually highly intuitive due to their real-world correspondence.
- **Multimodal techniques** - The multimodal techniques represent the combination of different input modalities. These techniques provide some advantages such as decoupling, error reduction and correction, flexibility, and complementary behaviour.

Each of these techniques for system control has its benefits and strengths, but also limitations compared to the others. Therefore, the choice of technique strongly depends on the system, tasks to be performed, and target users.

2.6.4 Other Approaches

Other more innovative interaction techniques go beyond the traditional 3D techniques described in the previous sections. One such technique is presented by Clifford et al. [CTL17] who develop a technique for pseudo-telekinetic object manipulation in VR. The authors got inspired by the Star Wars universe and The Force ability of some characters.

They believe that this kind of object manipulation is rather well suited to VR technology, because it will allow users to interact with virtual objects at a distance. In their system, the users can rotate, translate, push, and pull objects and with the extending option, the user can perform the action in a rate-controlled manner. According to the interactions, the control behaviour is represented as a state machine, entering and exiting different states. First, to manipulate an object, it should be indicated, what happens when a ray intersects with it. Afterwards, if the trigger of the controller is pressed, the selecting state is entered. Then different buttons of the controller can trigger one of the four states for manipulating an object. While in one of these states, if the user tilts/nods his/her head, the extending state is entered, which can control the rate of change for the given object manipulation. An evaluation of the system's usability with participants showed that, in general, the force-extension metaphor is promising. However, some participants had difficulties in keeping an object in focus while performing a nodding gesture with their head.

Analysis of the Time-Dependent Multivariate Impact Simulation Data

3.1 The Impact Simulation Data

The data used in this thesis is provided by T. Maindl and C. Schäfer [MS19]. They study how the orbit of an asteroid potentially colliding with Earth can be altered by the kinematic impact of a spacecraft. The given data represents a patch of the secondary body of the binary near-Earth asteroid (65803) Didymos, called Dimorphos. Dimorphos is the target asteroid of NASA's Double Asteroid Redirection Test (DART) [NAS17], as well as ESA's Hera [ESA19c], which are both part of the Asteroid Impact and Deflection Assessment (AIDA) mission [ESA19a]. DART aims to test whether a kinematic impactor could successfully deflect an asteroid from Earth's orbit for planetary defense purposes. Hera will, later on, analyze the consequences of DART's impact and gather data to understand the asteroid's structure and the effects of the collision. This data will make it possible to develop strategies to handle an eventual future impact on Earth. Figure 3.1 shows the concept of AIDA's mission.

The *Impact simulation data* by T. Maindl and C. Schäfer models the effects of an impact event on an asteroid. The data is multivariate and time-dependent and represents a section of an asteroid consisting of more than one million 3D points, changing their spatial location over time. Each point has attributes such as position, speed, mass, density, energy, pressure, porosity, and other domain-specific internal simulation values. The data consists of 400 frames to store the attribute value changes in time. When visualized, the data points form a cuboid structure with a crater in the center where the points diverge (see Figure 1.2). T. Maindl and C. Schäfer are interested in studying and analyzing

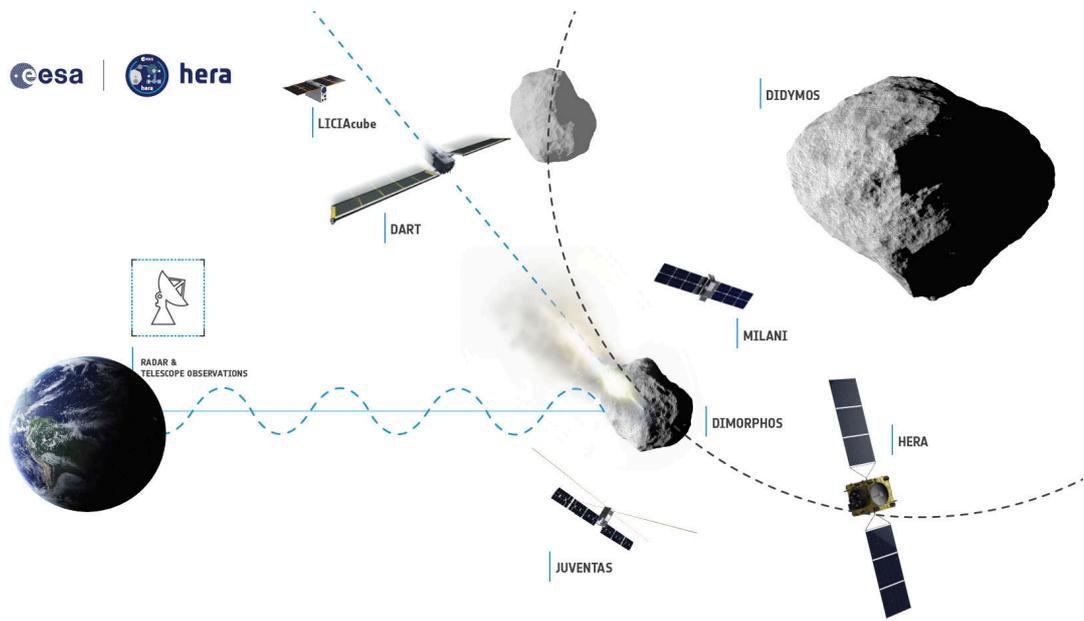


Figure 3.1: NASA DART and ESA Hera mission concepts [ESA19b]

the asteroid characteristics at different areas. By simulating the impact with different attributes and at multiple time-steps, they want to explore and understand the data characteristics and how they can influence the deflection of an asteroid.

3.2 Domain Expert Tasks for the Impact Simulation Data Analysis

The main purpose of this thesis is to provide a Virtual Reality system for the domain experts that can help them explore and get insight into the *Impact simulation data*. Therefore, the user group of our system are the domain experts who study the process and results of impacts through simulations. We implemented and designed our VR system according to the domain experts' requests and requirements. The essential features of the system were specified during several meetings before and during the development process and were defined in the form of tasks.

One of the requirements is to get insights about the structure of the data on the surface and the inner parts of the simulation for different attributes. The domain experts are particularly interested in finding extreme values, especially at the impact crater. Due to the cuboid shape of the data visualization, the inner parts of the structure are occluded. The domain experts need a tool or option to cut out or filter different parts of the data from the visualization. They are interested in tools that can help them analyze regions of interest in more details. Despite analyzing regions separately, the domain experts also need a tool allowing them to compare different areas and investigate their similarities and

differences to understand the impact better. Furthermore, T. Maindl and C. Schäfer are interested in observing and examining attribute changes at a particular region over all simulation time-steps (frames). The experts also request an option to export statistical information about data properties at individual regions gathered during exploration. The exporting functionality is not considered in the current work and can be part of future implementations. In our thesis, we address the following set of tasks:

- **T1:** Get an overview of the data model at the surface and of inside parts
- **T2:** Find outliers and extreme cases in the simulation data
- **T3:** Filter out data to remove irrelevant information and highlight relevant locations
- **T4:** Explore separate regions or portions of the data to get insight into the attributes and their distribution at the given areas
- **T5:** Compare multiple regions in terms of statistical differences and similarities to detect trends and patterns
- **T6:** Analyze state and attribute changes in time by comparing different time-steps

Based on the derived tasks, we developed our system to support the necessary features for the domain experts. How we design our VR system to address all of the tasks, will be described in detail in Chapter 5. The next chapter gives an overview of a basic implementation of a desktop system used to visualize and explore the *Impact simulation data*. This 2D desktop system is integrated into and used as a basis for the current work.

Impact Simulation Data with 2D User Interfaces

To understand the bases of the current thesis, in this chapter, we provide an overview of a basis project that has been carried out as preliminary work. The goal of the basis project was to enable the visualization and exploration of the *Impact simulation data* on a desktop with 2D devices in order to solve the tasks defined together with the domain experts. The resulting desktop system provides several visualizations and implements various interaction techniques and tools to support the faster and intuitive exploration of *Impact simulation data*. The system is able to tackle multivariate data and, at the same time, consider the time component. Additionally, it provides some limited techniques allowing the user to explore the data at given time-steps and observe the changes over time. This chapter explains the desktop system's capabilities and available tools since they are integrated into our VR system. Also, we review the tasks the system is solving either fully or partially and the existing limitations. Then, Chapter 5 provides an extensive description of the VR immersive system that builds on top of the 2D analysis tool and is the main focus and contribution of this thesis.

4.1 Methodology and Implementation

Given the project's goal and tasks, and based on the data structure and characteristics, different UI controls and visualization techniques were implemented to support *Impact simulation data* exploration. In order to solve task **T1**, we provide a transfer function with different colors mapped to the points and also axis aligned clipping planes to explore inner parts. The data range and out of range options are used mainly for solving tasks **T2**. For task **T3**, we provide histograms and a parallel coordinates plot that can be used to filter out data points by brushing. We used a set of predefined box-shaped probes to solve task **T4**, but these probes are only limited to the central region of the simulation.

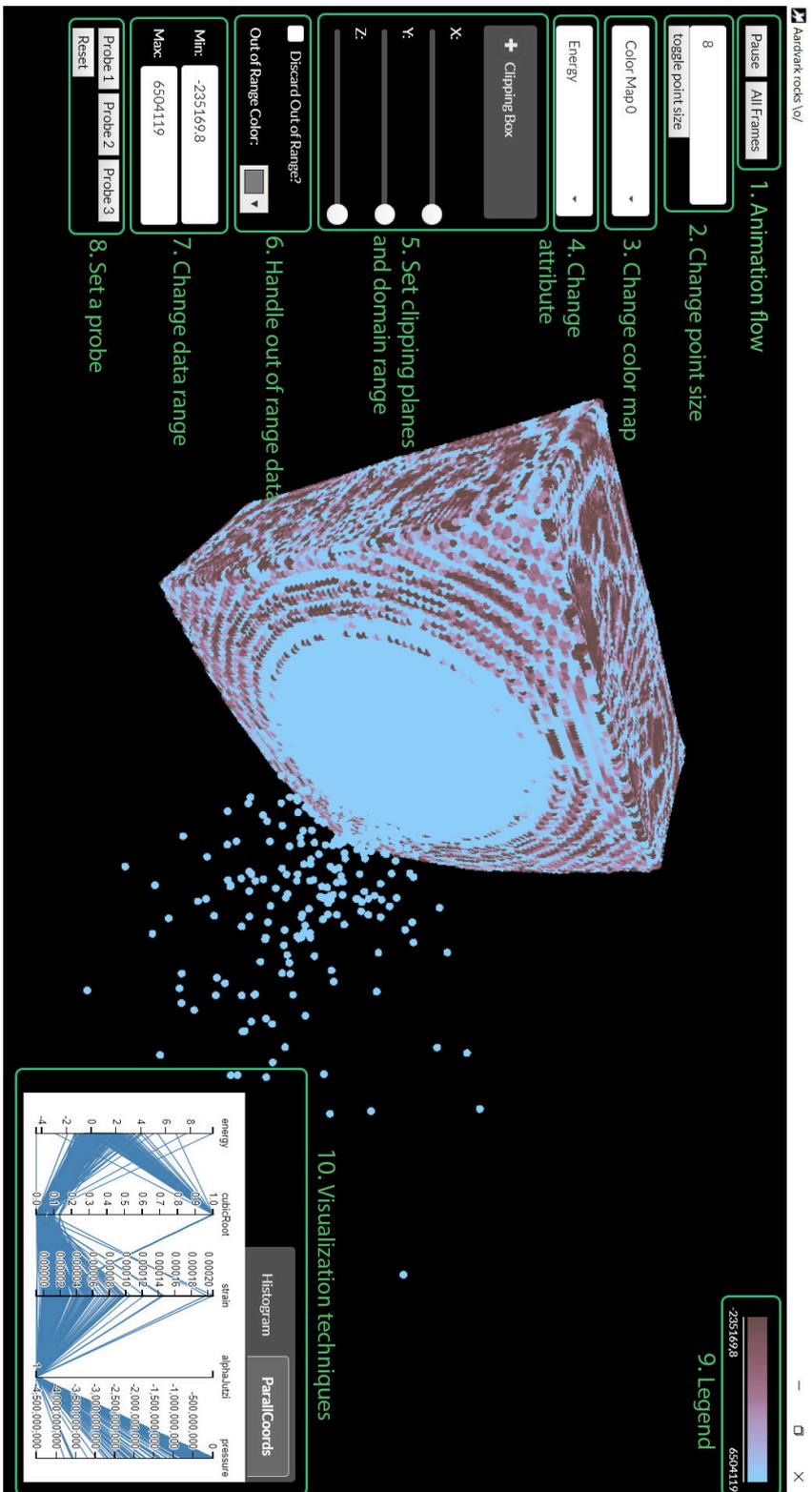


Figure 4.1: Overview of the interface for the 2D desktop analysis of the *Impact simulation data* [VS20]

Task **T5** is not considered in this project and the solution for task **T6** is limited. For **T6** domain experts can use the histogram to observe distribution changes in time at the central region since the plot is updated in real-time for every frame of the simulation. Figure 4.1 gives an overview of the interface of the implemented program. Each green section illustrates a different part of the control options to modify the 3D visualization, which are explained in the following:

1. **Animation flow** - The two buttons ("Play/Pause" and "All Frames/One Frame") are used to control the flow of the animation. The simulation consists of multiple frames that change over time. The "Play/Pause" button is used to pause and play the animation. The "All Frames/One Frame" button does not directly impact the 3D visualization, but influences the histogram. It controls if the histogram should display only the current frame at the time when the button was selected or continuously display the new data as the animation cycles through all frames.
2. **Change point size** - The field allows changing the size of the points in the 3D visualization and the button "toggle point size" directly switches the sizes from 8 to 20 units and vice versa.
3. **Change color map** - The color map represents a transfer function that maps a value to a specific color. The drop-down menu is used to select a different color map.
4. **Change attribute** - Using the drop-down menu, the user can change the current attribute of the points. This will also affect how the color map is applied to the data and enforce a new histogram visualization, which only displays one attribute at a time.
5. **Set clipping planes and domain range** - The clipping planes accordion menu is used to control the permitted range for the 3D points. The ones outside of the clipping planes are discarded. The sliders for the axes in X-, Y-, and Z-direction also restrict the domain range and slice it at different depths. This makes the exploration and distribution of the central points of the 3D visualization easier.
6. **Handle out of range data** - Certain points may not fall within the range of the current attribute's min and max values. The same might also happen when applying filtering using the two visualization techniques. The points that are not within the range could either be discarded entirely from the scene or identified by specifying their color by opening the color palette next to "Out of Range Color". The default color is grey.
7. **Change data range** - The data range depends on the currently selected attribute, since each one of the attributes has a different extent. The given fields are used to restrict the data range by setting new min and max values. Changing the data range will affect which points in the visualization are visible.

8. **Set a probe** - Each of the three probe buttons invokes a 3D box with a different side length, positioned at the center of the visualization. They are used to explore the data in the middle where the collision has the most significant impact and thus greatest value changes of the different attributes. The probes filter out the data and when selecting one of the probes the two visualization techniques (a histogram and a parallel coordinates plot) display information according to the data covered by the probe.
9. **Legend** - The legend is used to show the color map currently in use and illustrates how the min and max values of the active attribute are mapped to a color.
10. **Visualization techniques** - To further support the data exploration, two more visualization techniques are introduced - a histogram and a parallel coordinates plot, explained in the following paragraphs.

Figure 4.2 demonstrates a different setting using the above described controls. In this new setting, the color map is changed. The domain range is restricted in the X-direction and slightly in the Z-direction, so that it is possible to observe the inside parts of the 3D point cloud. The data range is also modified by fixing the min value at 0.5 so that all points outside of the range are colored in black.

The two main visualizations used to explore the data are both implemented using the JavaScript library D3. The user can switch between them by selecting one of the two tabs on the right bottom corner of the program (named "Histogram" and "ParallCoords").

4.1.1 Histogram

The histogram [Hol18] is used as an approximate representation of the distribution of the data according to the values of a given attribute. The histogram can be activated by choosing one of the three probes. Each probe has a different extent and cuts out a section of the data at the center of the simulation, where the collision happens. When clicking on a probe a histogram of the respective selection is displayed, showing the distribution of the data of the current attribute. The user can additionally explore the 3D point cloud by using brushing to filter out regions of interest. The filtered particles that are out of the range are encoded with grey color by default, but this could easily be changed by the UI controls at the left part of the screen. Additionally, the "Discard out of range" option could be enabled, so that all points outside the range of the filter are not rendered at all. Also, the user can click on the "All Frames" button, which will update the histogram for each frame if the animation is playing. In general, the histogram is useful for detecting clusters and patterns and gives an overview of the distribution of the data. Figure 4.3 shows an example usage of the histogram by selecting one of the probes (Figure 4.3a), coloring or discarding the points (Figure 4.3b), and using brushing to filter the data (Figure 4.3c).

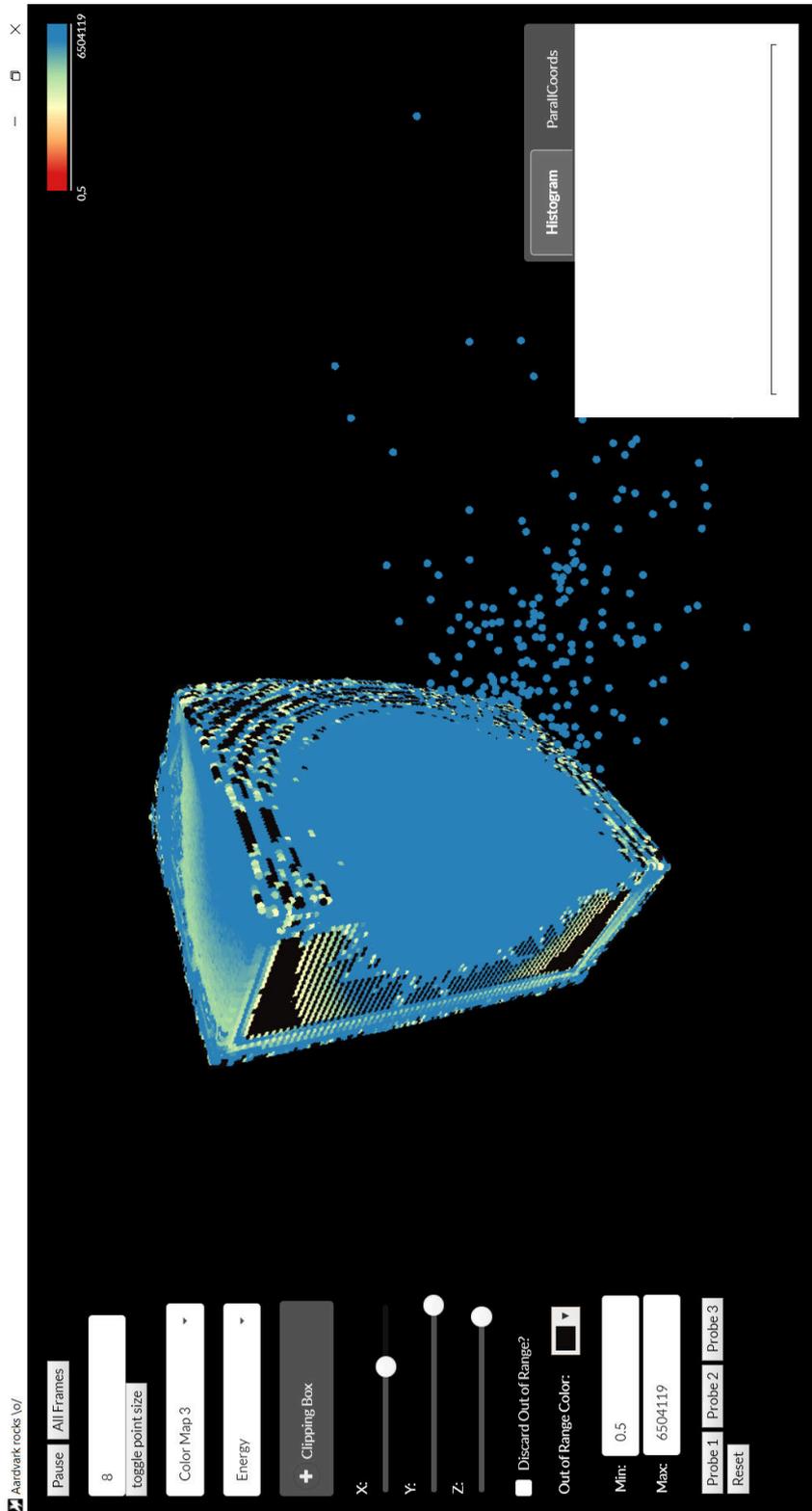
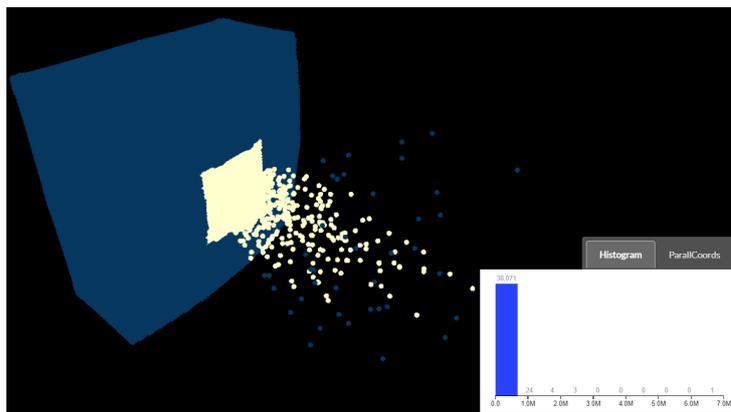
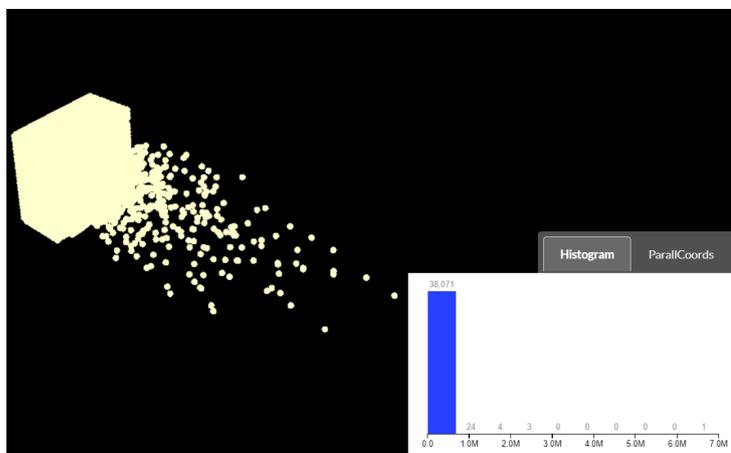


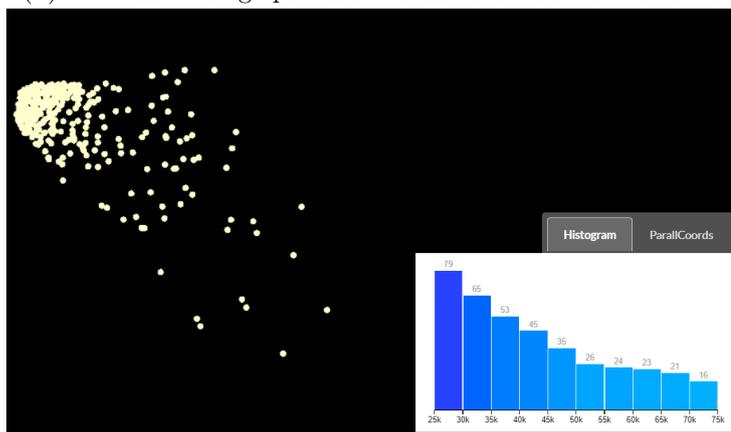
Figure 4.2: The *Impact simulation data* visualization with modified settings using the UI controls



(a) Probe 3 is selected, the histogram is displayed and the out of range color is set to dark blue



(b) The out of range points are discarded instead of colored



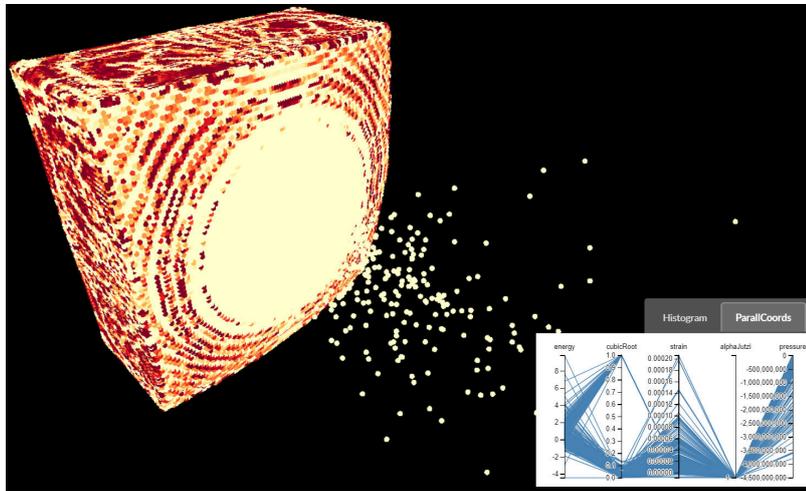
(c) The points are filtered by using histogram brushing

Figure 4.3: Examples of using the histogram to filter data

4.1.2 Parallel Coordinates

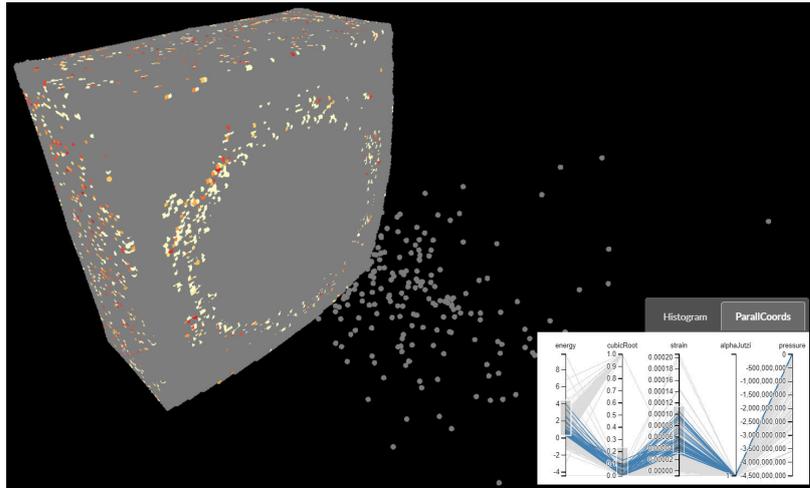
A parallel coordinates [Dav11] plot is used to visualize multiple dimensions at the same time. It is beneficial for the given data to evaluate the relations of the different attributes and their respective values (compared to the histogram, where only a single dimension is observed). Besides, a parallel coordinates plot also has the filtering option to examine only parts of interest in the data. This way, it is possible to simultaneously filter every individual dimension and display only the points that correspond to the specific filtering criterion. Analogously to the histogram, the parallel coordinates plot is activated by one of the probes' buttons and the discarding option has the same effect. However, the "All Frames" button is not available for the parallel coordinates plot, because in the project we limited the analysis with a parallel coordinates plot to single time frames.

Figure 4.4 gives an example of how the parallel coordinates plot can be used to reduce and explore the data. Equivalent to the histogram, the parallel coordinates plot is also activated by selecting one of the probes. However, there is also a default parallel coordinates plot that uses the first 500 entries of the data and is visible even when none of the probes is selected (Figure 4.4a). We limited the number of entries in order to reduce visual clutter on the parallel coordinates plot. We use the first 500 entries, because we empirically discovered that they are a sufficient representative for getting an overview of attribute correlations. According to the values of these entries, brushing is applied on three of the attributes (energy, cubic root and strain) and all the points that do not fall into the range of the filters are colored in gray (Figure 4.4b). Figure 4.4c shows the result after discarding the points that are out of range.

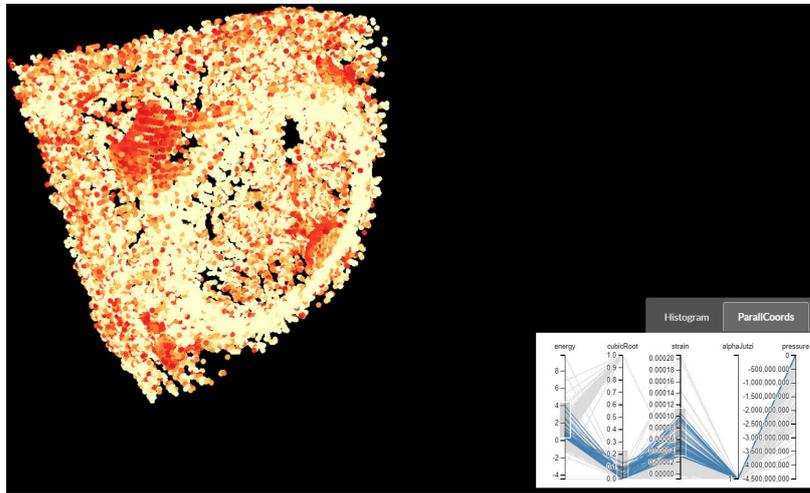


(a) The default parallel coordinates plot is displayed

Figure 4.4: In this example we use the parallel coordinates plot to filter out the points according to three dimensions. When discarding the out-of-range values the internal structure is exposed showing regions with moderate energy, low cubic-root, and low strain.



(b) Filtering is applied on three of the dimensions (energy, cubic root and strain)



(c) The out of range points are discarded

Figure 4.4: In this example we use the parallel coordinates plot to filter out the points according to three dimensions. When discarding the out-of-range values the internal structure is exposed showing regions with moderate energy, low cubic-root, and low strain.

4.2 Results and Limitations of the 2D Approach

The resulting system can visualize the given multivariate data in 3D space and provides various interaction techniques for their exploration. The tool is a combination of a 3D visualization and 2D abstract views. The 2D UI elements provide various interaction possibilities with the 3D view, as well as two visualization techniques - a histogram and

a parallel coordinates plot - representing the data's distribution and correlations. These visualization techniques also control and facilitate the direct manipulation of the 3D view by using brushing and filtering. The quality and visual appearance of the 3D point cloud of the data is sufficient for the domain experts. However, the given approach, cannot solve all domain expert tasks or is only very limited. The existing problems and limitations will be described in the following paragraphs.

Technical Limitations of Histogram and Parallel Coordinates

The histogram displays the data distribution at a particular time, representing the distribution for a given frame. For a user to observe the change and advance of values in time, a new histogram distribution needs to be displayed each time the frame of the animation is updated. Computing the distribution again for each frame leads to a computation overhead. Fortunately, it is not necessary to recompute it every time because the position and size of each probe are fixed. Consequently, the histogram values are recomputed for each probe and frame only when the attribute is changed, leading to faster and smoother running of the program and preventing long computation times. However, this performance improvement comes at the expense of not having flexibility for exploring the data model fully since only one probe with a fixed size and position can be placed at a time. Also, this system is not optimized for placing multiple probes and displaying several histograms for various regions, which restricts the domain experts in exploring and comparing different areas.

The second visualization technique also caused similar problems. Each probe covers a different number of data points - Probe 1 contains about 8 000 points, Probe 2 about 18 000 points, and Probe 3 about 38 000 points. Visualizing the data from the first probe in the parallel coordinates plot already produces a lot of lines across the dimensions, which results in visual clutter. Displaying all data points reduces readability significantly. Therefore, we decreased the dataset size for the parallel coordinates plot by using a random subset of the dataset. We used 10 % of the points covered by each probe in the current implementation. This implementation is not optimal and excludes attribute values that the randomly computed subset might not cover. Still, the parallel coordinates plot was not the central element for that project, and the given results were sufficient for the domain experts. In this thesis, we are not extending and improving the given limitations of the parallel coordinates plot since this is not the main focus of our work. Instead, we provide different tools and visualizations suitable for the VR environment to support domain experts.

Conceptual Limitations and Possible Future Improvements

Although the most essential region to investigate for the domain experts is the impact crater of the visualization, where all three probes are currently fixed, other parts might also be of interest. Therefore, this system is very limited in solving task **T4**. It also does not provide a comparison tool or option between different regions (**T5**). The solution for task **T6** also does not entirely fulfill the domain experts' needs and expectations. They

need a tool to compare simulation changes over time. Consequently, the given system needs to be upgraded and extended to cover all needs of the domain experts.

Most of the described problems and limitations could be overcome by introducing additional visualization and interaction techniques. The extensive amount of research in the field of immersive data visualization and exploration (Chapter 2) has shown that immersive environments could offer a lot of benefits and advantages compared to traditional 2D desktop visualizations. Bearing this in mind, we expect that using virtual reality to visualize and analyze the *Impact simulation data* will offer more options in terms of interactions and make the exploration faster and more intuitive. Integrating the system in an immersive environment will provide domain experts and users with powerful tools for data analysis, such as selecting the 3D probes at the desired locations and displaying the corresponding information directly in the 3D space. Such an approach would give more flexibility and expand the interaction and selection possibilities to fulfill all implementation requirements. The following chapter will give a detailed description of the resulting system for visualizing the *Impact simulation data* in VR.

Impact Simulation Data in Virtual Reality

5.1 Overview

As mentioned in the previous chapter, using 2D devices to visualize and explore the *Impact simulation data* introduces some limitations due to the specific nature of the data. Therefore, the goal of this thesis is to provide a Virtual Reality system incorporating various visualization and interaction tools to support domain experts in exploring and analyzing the *Impact simulation data*. We try to answer whether the exploration is intuitive and effective and helps domain experts in gaining insights faster and easier by using an immersive environment. The resulting system combines well-established visualizations such as histograms and box plots and various interaction and navigation techniques, suitable for an immersive system. Given this, the following three sections will cover different aspects of the system to describe its structure and functionality. Section 5.2 gives a brief description of how we address the domain experts' tasks and what solutions we propose. Section 5.3 gives an overview of the challenges that needed to be considered and overcome. The different visualizations techniques are covered in Section 5.4. Finally, Section 5.5 describes the implemented interaction possibilities and how to trigger them.

5.2 System Components and VR Solutions to Domain Experts' Tasks

We introduce an immersive system combining several components, tools and interactions to support domain experts with solving their tasks and ease the exploration process. There are two main components in our virtual reality space. The first component is the 3D

point cloud visualization of the data and is placed at the center of the environment. The second component is an intuitive 2D UI adapted from the 2D desktop system described in Chapter 4. The 2D UI is placed on a TV screen in the VR space and is used for modifying various properties of the point cloud visualization. These properties include filtering and brushing options, appearance properties such as point size or color, clipping capabilities and others that will be described in detail in the next sections. An image of the two components can be seen in Figure 5.1.

Our system works with standard Virtual Reality setups with room-scale tracking, including a headset, two controllers and two base stations. In this work, we propose the so called *probes* as a key concept the visualizations and interactions are based on. A *probe* is a 3D object which can be resized and placed arbitrarily in the scene by using the two controllers tracked in the 3D space. To keep the operations as simple as possible, we use spherical probes which turned out to be versatile and sufficient for our application. The VR application is an extension of the 2D application and provides additional functionality to solve the tasks. We also incorporate additional options to help solve the tasks. For task **T1**, except for the axis-aligned clipping planes, the user can activate a controller-based clipping plane that moves with the controller. For solving task **T2**, the system supports an additional mode, i.e. transparency mode, which uses alpha blending to highlight points or regions with extreme values. For filtering data (**T3**), the user can additionally highlight or discard the lower and higher outliers with the discard options provided in the 2D UI. The *probe tool* is the central exploration component in our system used to create the probes. With the help of the probes tasks **T4**, **T5** and **T6** can be solved. A probe allows users to explore a particular region of the point cloud model by displaying the attribute distribution with a histogram or showing statistical information in the form of text, including different statistical measures for the points inside the probe (**T4**). The system supports the placement of multiple arbitrary probes and provides an option to compare them with the *analyze regions tool* (**T5**). As a tool for comparison we propose box plots placed on planes in the 3D environment, which can be toggled by the user. Similarly to that, the user can also compare attribute changes in time at a specific region by activating the *analyze time tool* (**T6**). This tool allows users to control the simulation flow and go back and forth through the frames. The user can also select the frames to analyze further. For each selected frame, the system displays a box plot. The following sections will describe these exploration tools and the necessary interactions to trigger them in detail.

5.3 Challenges

Several challenges need to be taken into consideration before developing the system. The first challenge is regarding the **performance**. As mentioned earlier, the *Impact simulation data* consists of 400 frames with more than one million points, each point having multiple attributes. Since the data has multiple frames, it is challenging to make the visualization efficient while still being able to interact with the simulation and get real-time feedback. Further preminent issues to consider are the **visualization**

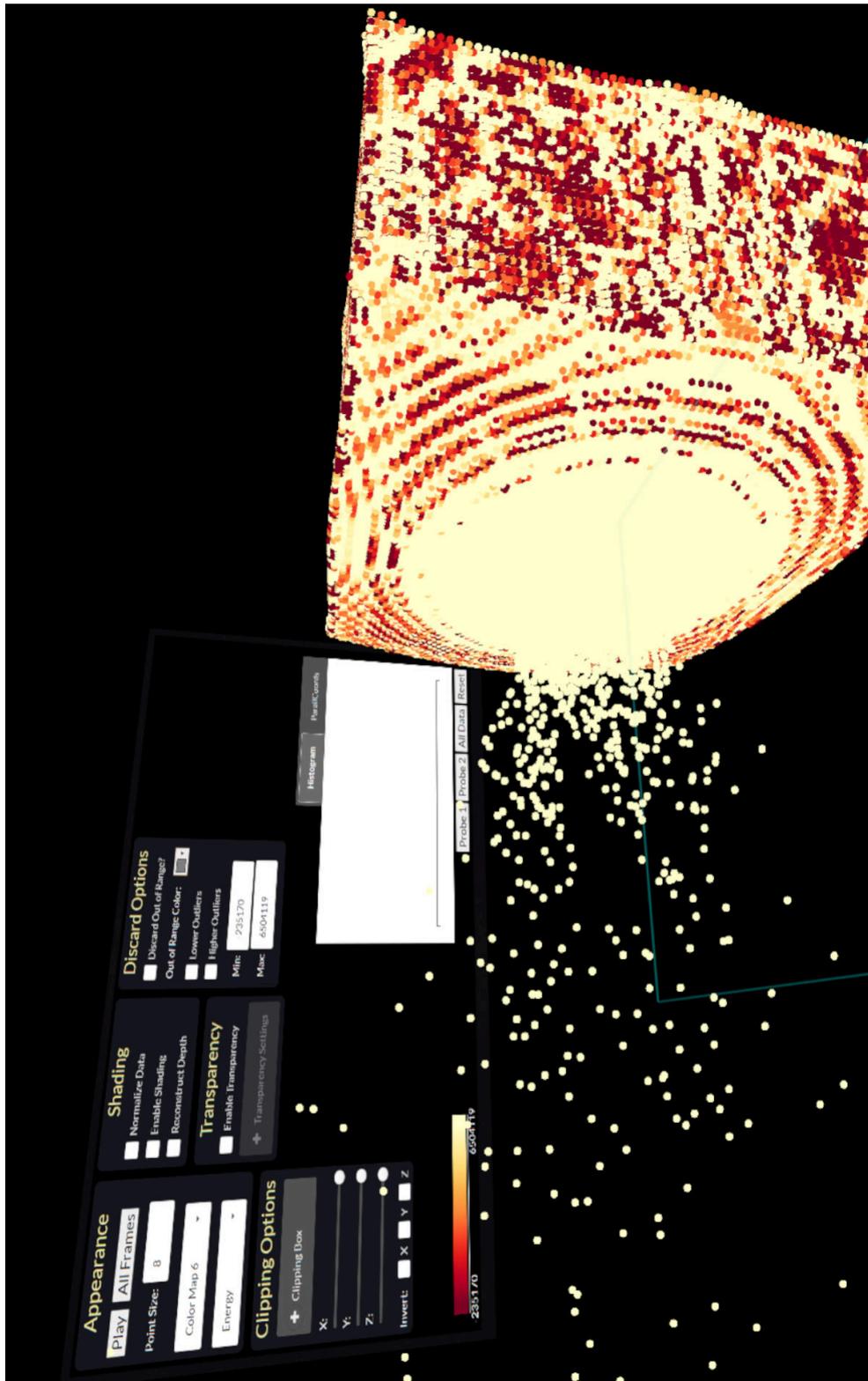


Figure 5.1: The two main components of the VR system. In the right part of the image is the 3D point cloud visualization of the data. On the left is a 2D UI placed on a TV screen used to modify properties of the point cloud.

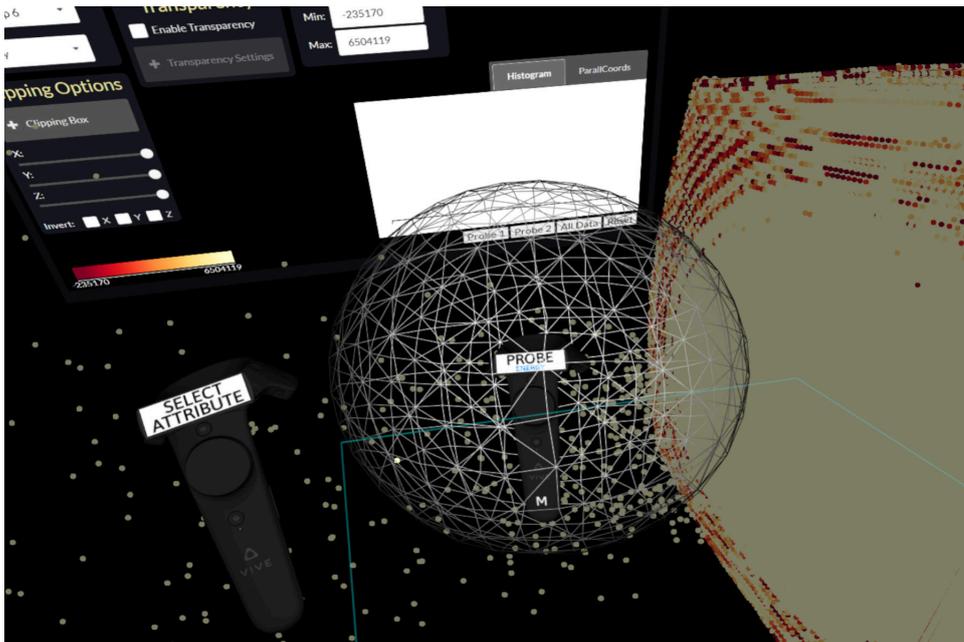
challenges. More specifically, it is essential to find such visualization techniques that can be expressive, give information about the simulation, and also cover the multivariate and time-dependent aspects of the data. Closely connected with that are the **interaction** challenges. On the one hand, users need to naturally interact and move within the environment which should resemble the real 3D world. On the other hand, it is also beneficial to introduce additional interaction techniques that are not possible in the real world, but can provide more functionality and support a faster exploration, such as grabbing and moving objects from a distance. **Depth perception** is another key element to consider when developing a system in the 3D space. We planned to have some of the objects to be transparent, which causes additional computational overhead to achieve realistic depth perception.

A 3D environment in an immersive system offers more space than a 3D environment displayed on a 2D device. The additional third dimension allows one to move and place objects freely in space and observe them from various angles. Still, **occlusions** could occur and make the exploration challenging if there are many objects in the scene that are overlapping. Therefore, we also had to implement various algorithms and interaction capabilities to reduce occlusions. Last but not least, a very common problem with VR systems is **motion sickness** as discussed in Section 2.5. It could cause unpleasant symptoms that can distract users from the VR experience and reduce comfort. According to previous research and various recommendations, we developed our system intending to decrease the probability of motion sickness emergence. In the following chapters, we explain how we deal with the given challenges while still following the main objective of this work.

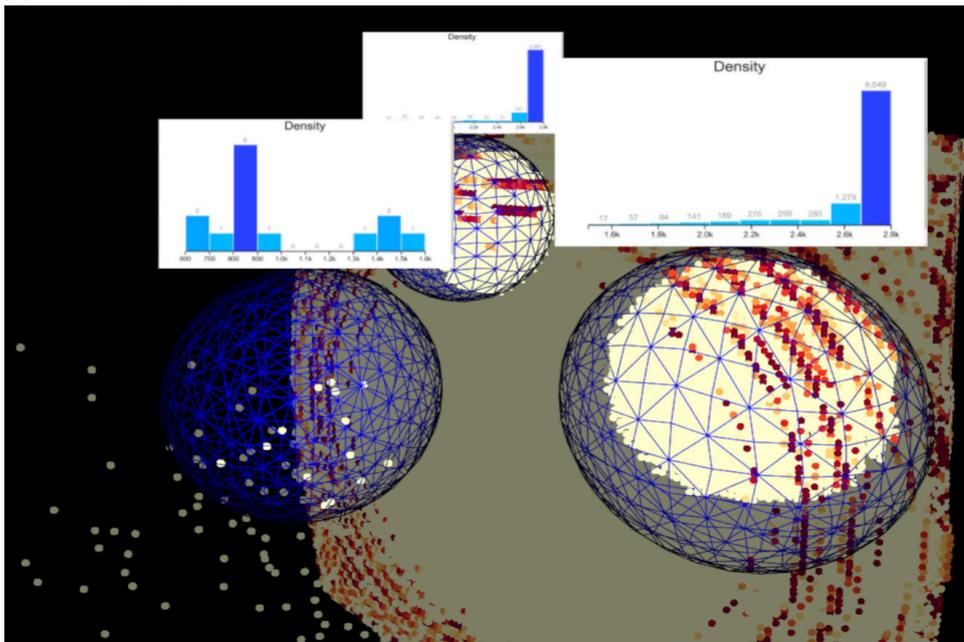
5.4 Visualization Techniques to Support Impact Simulation Data Exploration in VR

5.4.1 Exploration of Individual Regions

One crucial exploration aspect for the domain experts is analyzing different regions that might influence the impact and find areas with extreme attribute values. Therefore, we provide the *probe* as a central exploration tool. The probe is used to trigger several graphs, plots, and various visualization techniques to support the better understanding and insight of the *Impact simulation data*. Our VR system extends the basic 2D application with arbitrary manual probes (3D spheres) with varying sizes that can be created and placed anywhere in the environment by the user. For the VR system, we use spheres instead of box-shaped probes, because they do not have edges and do not require rotation transformations. Figure 5.2a illustrates how a probe is created with the controllers. The grey color indicates that the probe is still not placed in the environment, but is attached to the controller and can be modified. Once a probe is placed, the sphere is colored in blue and a 2D plane with a histogram above the probe is displayed. As Figure 5.2b shows, three probes are placed in the scene, and for each one of them, there is a histogram above the sphere. The histogram displays the distribution of the data points inside the

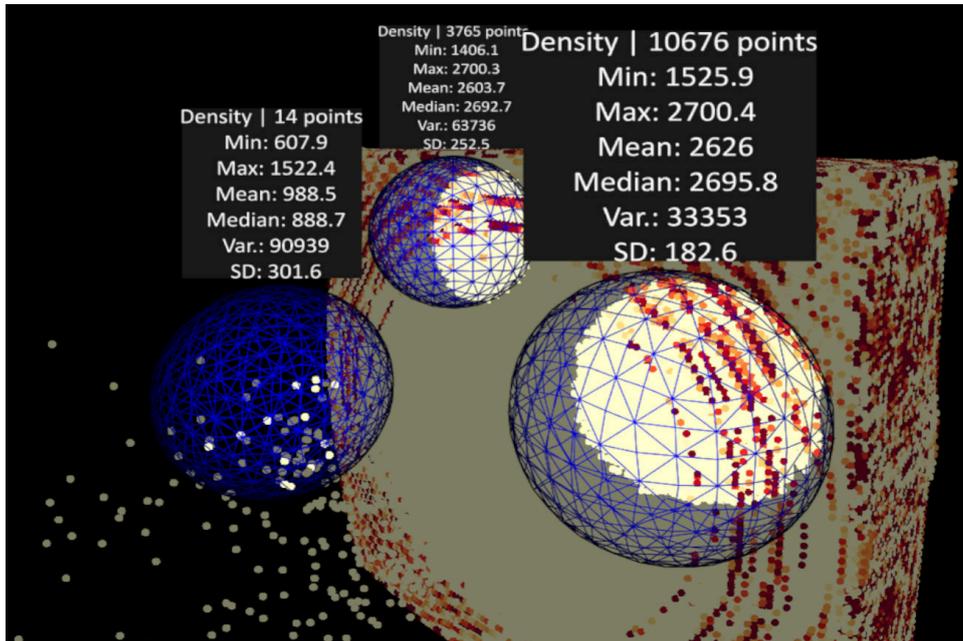


(a) Creating a probe, which is attached to the main controller and can be modified

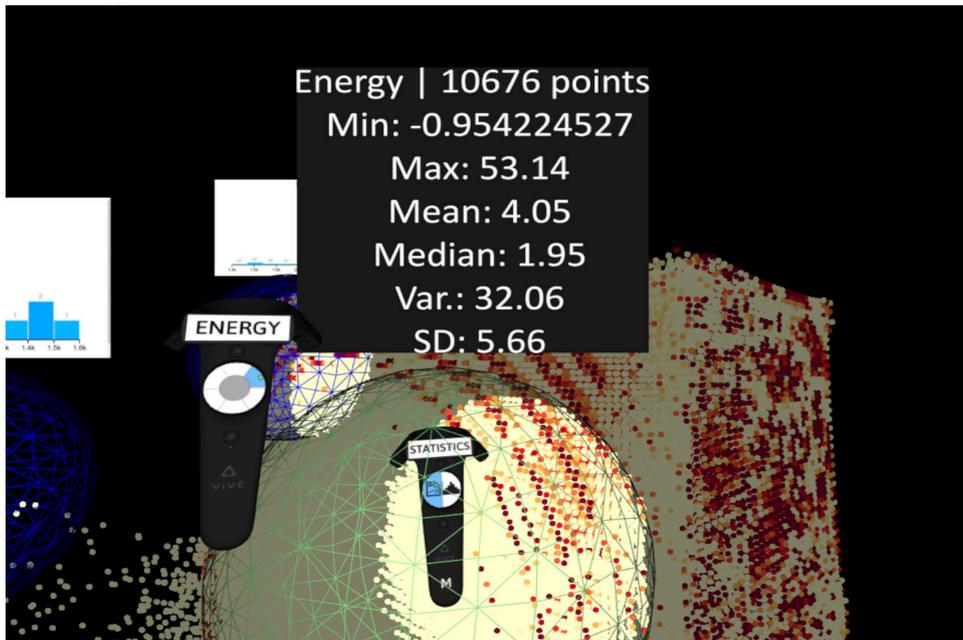


(b) Three probes with density attribute and varying sizes placed at different locations

Figure 5.2: Example of using the *probe tool* for creating probes, displaying histograms and statistics, and changing individual attributes.



(c) The same three probes, all with statistics instead of histograms



(d) Changing the probe attribute to energy for one of the probes

Figure 5.2: Example of using the *probe tool* for creating probes, displaying histograms and statistics, and changing individual attributes.

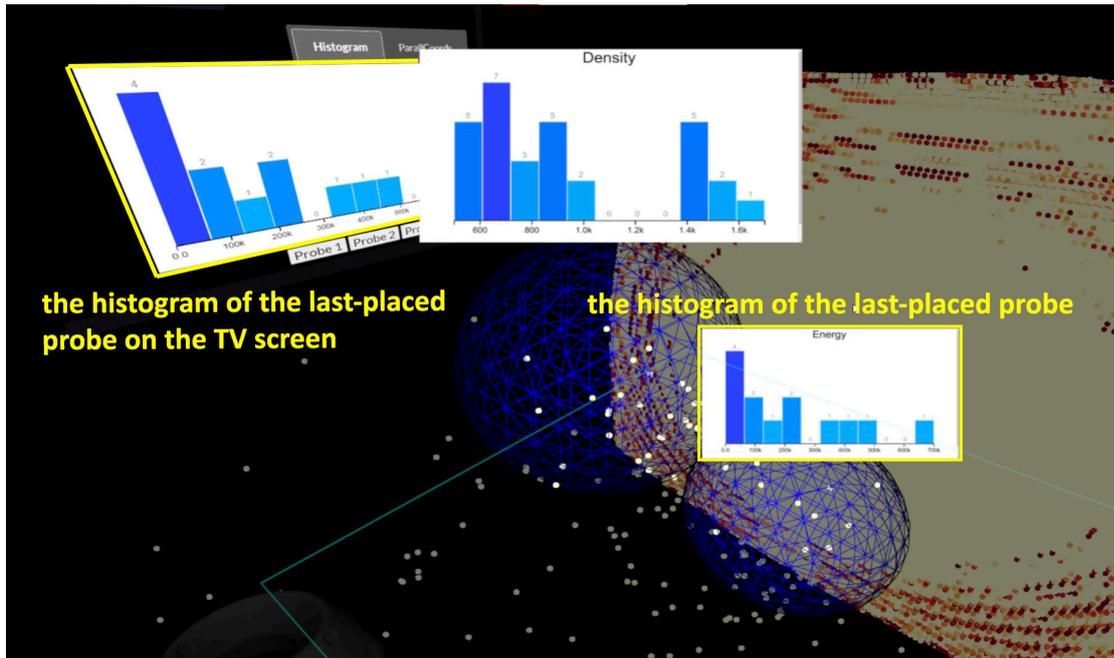


Figure 5.3: The histogram of the last-placed probe in the scene is also visible on the TV screen. The histogram above the probe is simply a texture, while the one on the TV screen is interactive and can be used for filtering data.

probe for a given attribute, which is displayed at the top part of the histogram plane. The histogram is split into several bins with a specific range. The higher the bin, the more points fall into that range according to the selected attribute. If there are no points from the point cloud inside the probe, the histogram plane is empty. The points lying inside a probe are highlighted with higher luminance, meaning the color appears brighter than points outside the range of a probe. The attribute of the histogram can be changed at any time, and the new distribution is computed accordingly and displayed. While the attribute visualized in the histogram can be changed at any time, we do not provide complex interactions such as brushing in VR. Nevertheless, to enable more complex interactions we also provide a 2D UI placed on a TV screen supporting extended capabilities of the original 2D application. The histogram of the last-placed probe is visualized on the TV screen and can be used for filtering the data (Figure 5.3).

Furthermore, the users can also display statistical information for the given probe instead of the histogram. An example of this feature is given in Figure 5.2c, where the histogram is switched to show statistics for all of the probes. The statistical information contains the current **attribute** of the probe, the **number of points** that the probe contains, the **min**, **max**, **average** and **median** values, and the **variance** and **standard deviation** for all the points in the probe. The statistics can help users read out exact values, while the histogram can provide a global overview of the data distribution. The user can place

multiple probes in the scene with different sizes, change the attributes individually, and switch between histogram and statistics. Figure 5.2d shows the new statistical information for one of the probes after changing the attribute from density to energy. The manual placement of probes through the controllers in the 3D immersive environment can be defined as a "Details on Demand" technique according to Schneiderman's Mantra [Shn96]. It allows users to explore particular regions of interest and display more information and details about the data points in the given region only after the user requested them, which covers the requirements for task **T4**.

To show the histogram and statistics information we rely on view-aligned billboards. Each billboard is attached to a probe, placed at the sphere center, and is offset so that it is on the top part of the probe. Additionally, every billboard is always facing the headset meaning that no matter from which angle or position the user observes the scene, the information on the billboard can always be seen. If there are multiple probes in the environment, it is likely that the billboards overlap from the viewing position. In order to handle this occlusion problem, we implemented an algorithm that is described in details in Section 6.3.

5.4.2 Region Comparison and Analysis

Another request from the domain experts was to be able to compare different regions (**T5**). They are particularly interested in discovering how a given attribute varies at the impact crater compared to other parts of the data simulation. The histogram and statistics are suitable for getting insights into the distribution for a particular region. Clearly, remembering statistical information between successive measurements is not feasible. Simply placing them simultaneously in the environment poses the problem of occlusions, especially if there are multiple probes in the scene. Additionally, in our implementation of the histograms we always use a maximum of ten bins. We limited the bins to reduce visual clutter. Also, ten bins are sufficient to get a general overview of the attribute distributions. The bins are adjusted to the current data range of the points lying inside a probe. Consequently, this may result in histograms with different bin ranges, making it hard to compare them. Therefore, for the comparison of multiple regions we introduce the *analyze regions tool*. This tool is used to investigate probes further with box plots. A box plot is a type of graph in descriptive statistics graphically displaying numerical data through quartiles. It represents a dataset based on five values - the minimum, maximum, the sample median, and the first and third quartiles. The box plot is constructed of two parts - the whiskers and the box. The whiskers can represent several alternative values, but, in this work, we use the minimum and maximum values as whiskers. The whiskers are displayed with a short horizontal line at the lowest and highest value. The box is drawn from the first to the third quartile with a horizontal line in the middle to denote the median.

Once the *analyze regions tool* is activated, an empty plane is displayed above the secondary controller and is attached to it so that the user can move within the environment and always have the plane at hand (Figure 5.4a). This plane is used to show multiple box

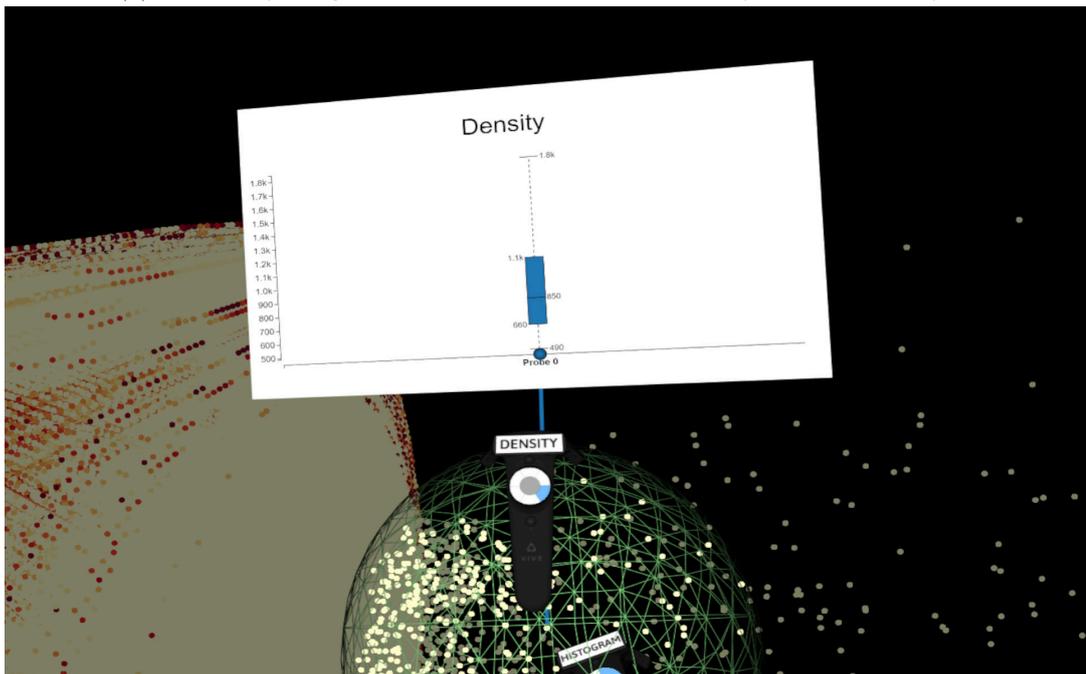
plots, one for each selected probe. With the main controller the user can select for which probes to display a box plot on the plane (Figure 5.4b, Figure 5.4c). The probes that are selected are colored in yellow to indicate that they are currently analyzed with the *analyze regions tool*. The user can place the plane with the box plots in the 3D environment at any location (Figure 5.4e), and this will create another empty plane attached to the secondary controller. The user can then again select new probes (a probe can be selected in multiple planes) to display box plots for them. Another option is to copy the information of an already placed plane and only change its attribute (Figure 5.4f, Figure 5.4g). In this way, an unlimited number of planes for different probes or attributes can be created and compared with each other (Figure 5.4h). The box plots in each plane are displayed horizontally in the same order in which the probes were selected.

The box plots in a plane are always displayed for the same attribute, denoted on top of the plane. Even if the selected probes are examining different attributes for the histogram, the box plots plane has its independent attribute that can be changed at any time with the secondary controller (Figure 5.4d). The horizontal line on the bottom of the plane denotes the probes, and the vertical line on the left shows the data range. The range depends on the minimum and maximum values in the probes. The box plot with the biggest difference of its highest and lowest value is the largest so that the other box plots are vertically shorter. The properties of the box plots can help users interpret datasets and compare regions by recognizing similarities, differences, and trends. First, the user can tell whether the compared probes are different by looking at the boxes. If the boxes do not overlap, then the compared probes are entirely different in terms of attribute values. If they overlap, the user can consider the median lines. If the median line of one box plot lies entirely outside the box of another box plot, it is likely that the probes are different. On the other hand, if both median lines lie within the overlap between two boxes, then the final step is to compare the whiskers. The whiskers represent the range of the two extremes, where large ranges indicate wider distributions, thus, more scattered data. With these properties of the box plots, the domain experts can read out crucial information about different regions and also recognize trends and similarities, which is what was required for task **T5**. A scenario showing the different functionalities of the *analyze region tool* can be seen in Figure 5.4.

As mentioned earlier, each box plot is a statistical representation of the data points inside a single probe and is displayed in the same order in which the probes were selected. However, it might be difficult and impractical to remember which probe corresponds to which box plot. Therefore, we introduce visual links used to connect each box plot to its respective probe. A similar approach has already been used in previous research in 2D as well as in 3D environments. Waldner et al. [WPL⁺10], for instance, use visual links to connect related pieces of information across various application windows and this way highlight relevant contents. Prouzeau et al. [PLE⁺19] use visual links to illustrate the connections between data points in visualizations with their respective context in the 3D space. The authors also optimize the link layout placement with respect to a particular point of view for reducing occlusions and clutter for one or more users in the immersive

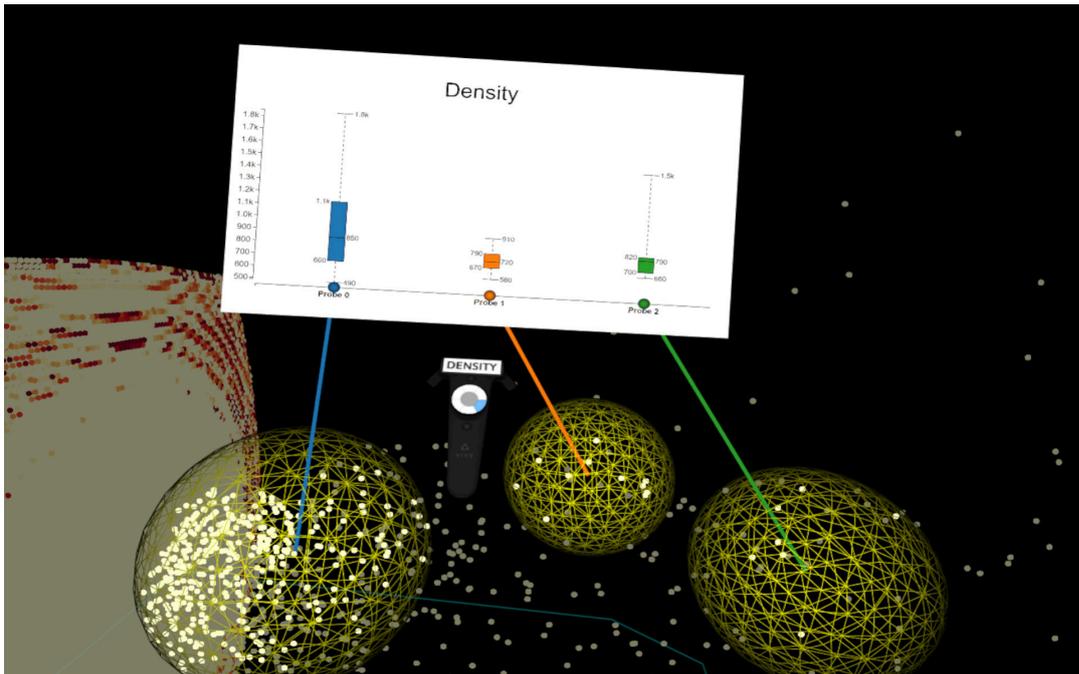


(a) The *analyze regions tool* is activated and an empty plane is displayed

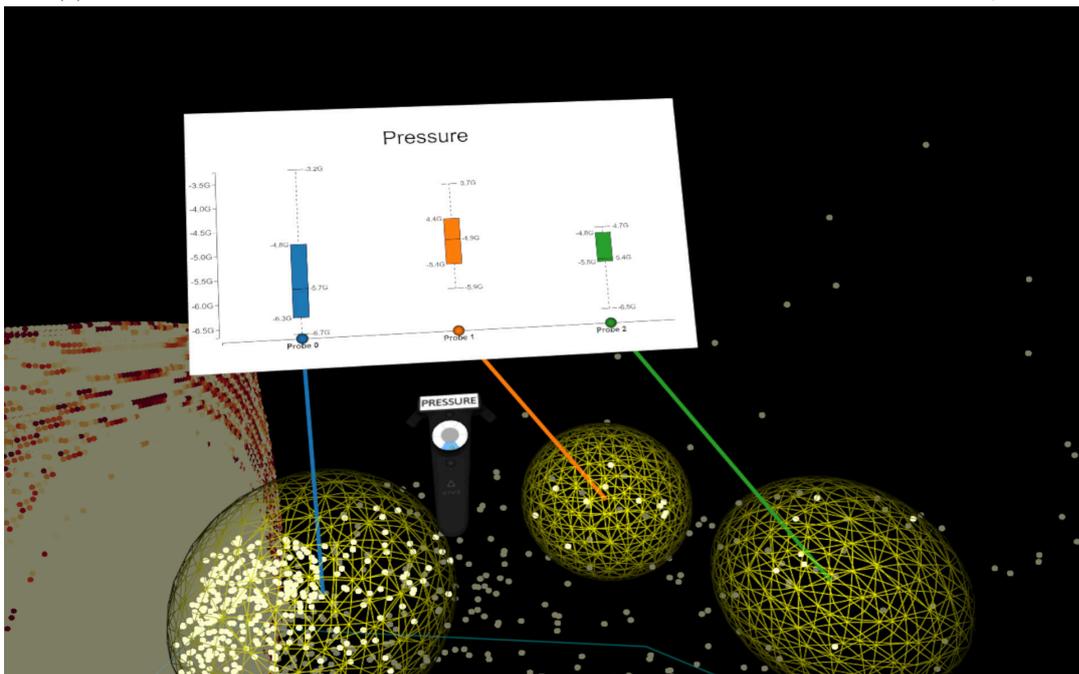


(b) One of the probes is selected for analysis and a box plot is displayed on the plane

Figure 5.4: Example of using the *analyze regions tool* for comparing regions and their characteristics according to different attributes

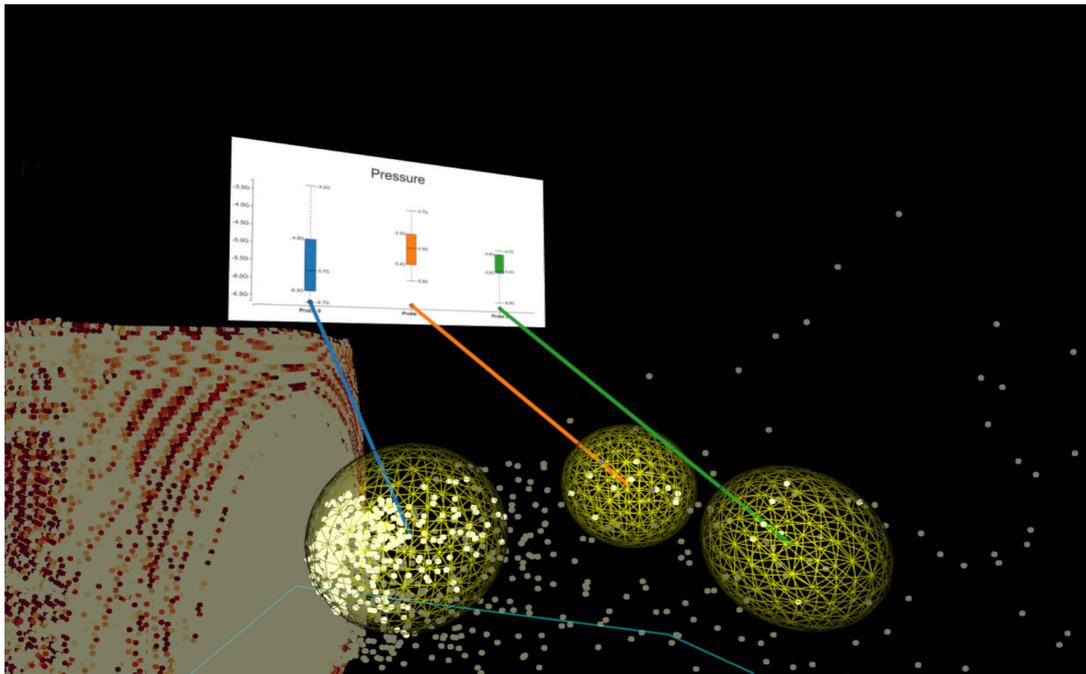


(c) Two more probes are added in the scene and are selected for a box plot analysis

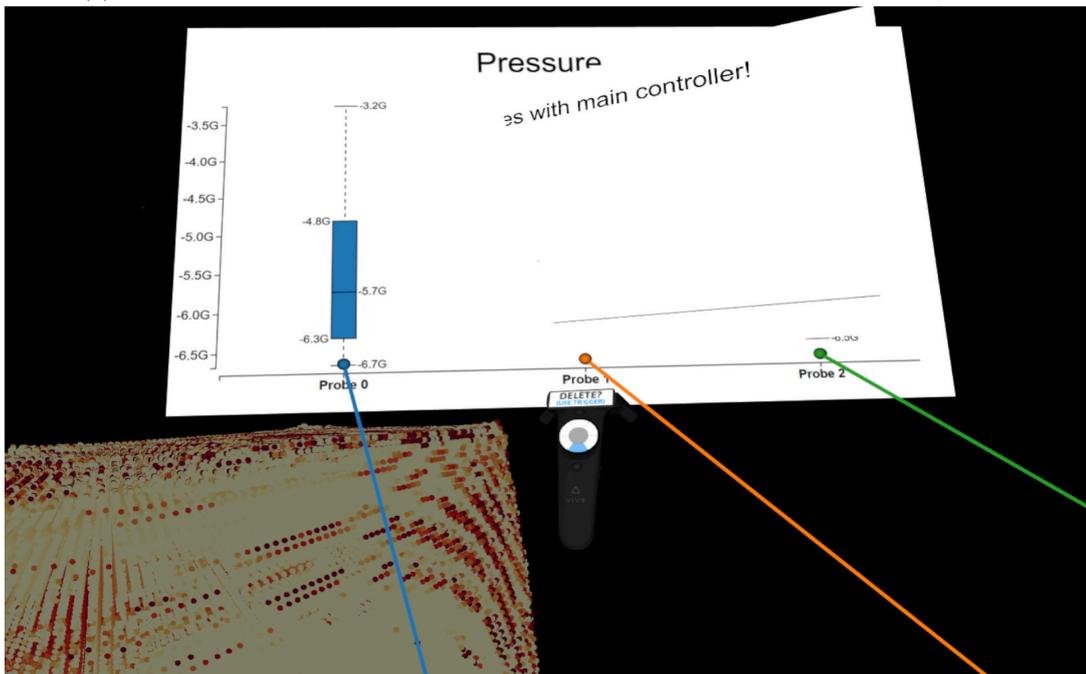


(d) The attribute of the active box plot is changed to pressure

Figure 5.4: Example of using the *analyze regions* tool for comparing regions and their characteristics according to different attributes

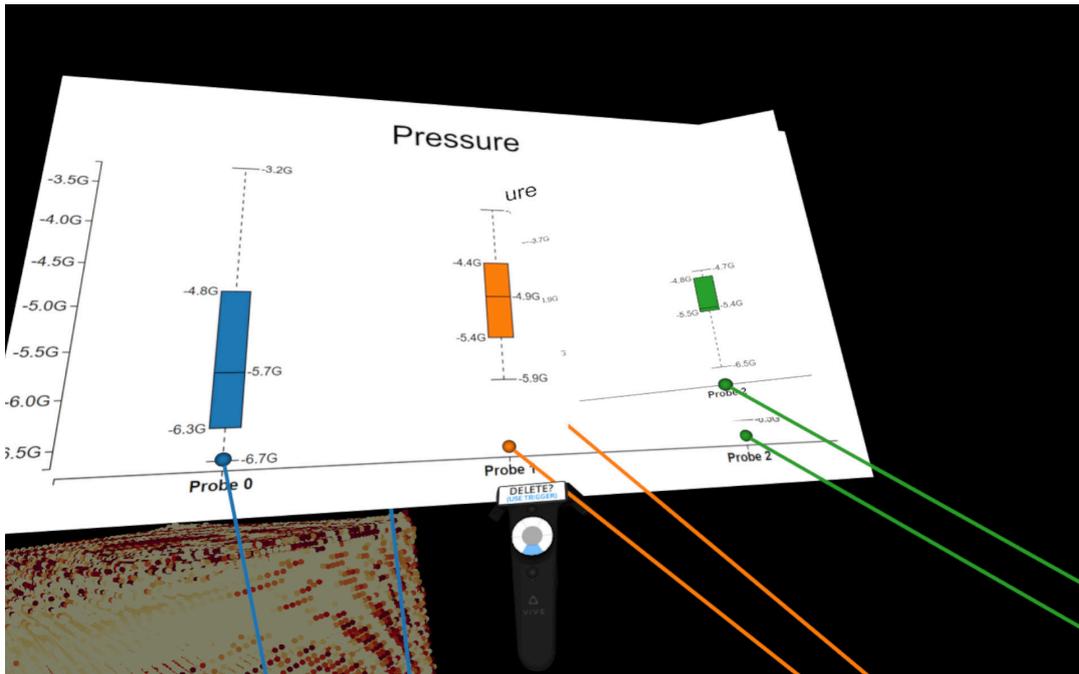


(e) The box plot is placed in the environment and cannot be modified anymore

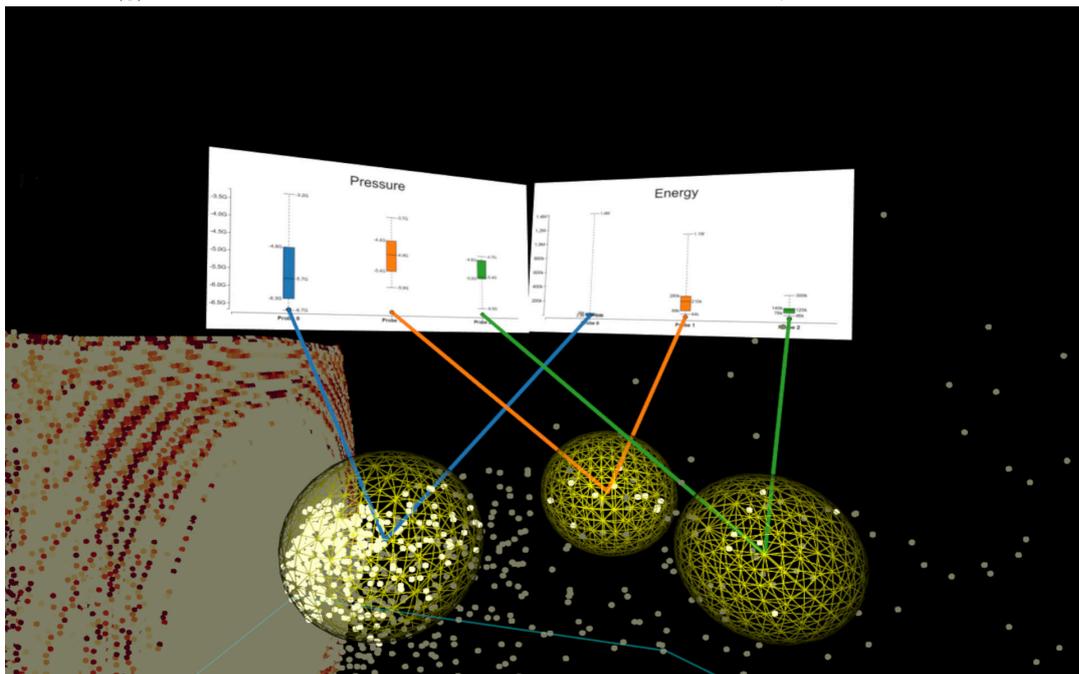


(f) A new empty plane is intersected with the placed box plot to copy its information

Figure 5.4: Example of using the *analyze regions tool* for comparing regions and their characteristics according to different attributes



(g) The copied information is now visible on the currently active box plot



(h) The attribute of the copied box plot is changed to energy and placed too

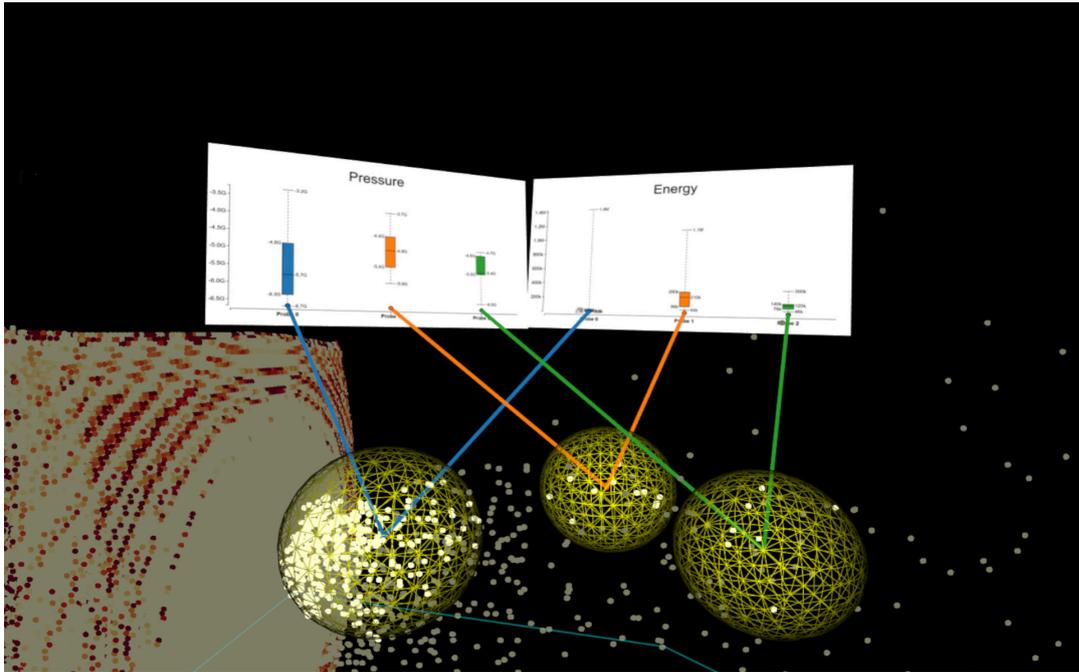
Figure 5.4: Example of using the *analyze regions tool* for comparing regions and their characteristics according to different attributes

environment. Their implementation uses a force-based model with various computations and forces applied to displace the lines and minimize intersections.

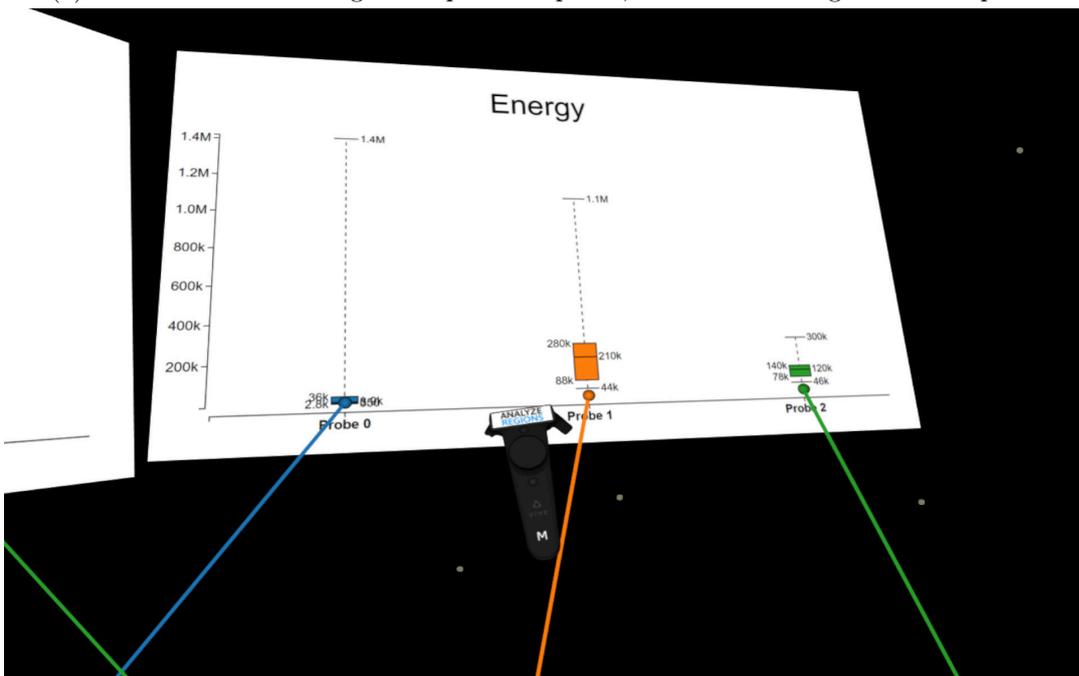
In our approach, we use simple straight lines to connect a box plot with its respective probe. We do not consider possible occlusion issues, as this goes beyond the planned scope of the work and will possibly be part of future improvements. Each of the lines connects the center of the probe with the bottom part of the box plot. Each line has a distinct color that corresponds to the color of the box plot. In each plane, the box plots are colored according to a pre-defined color scheme, meaning that, for instance, the first box plot in all planes will be blue, the second will be green, and so forth. Figure 5.5a shows how the visual links are connecting a probe to its respective box plot and how the lines are colored according to that. However, all selected probes for the *analyze regions tool* are colored in yellow to indicate that they are picked in at least one plane. We decided not to color the probes with the color of the line since a probe may be connected with multiple lines with different colors, which will lead to confusion. The user also can turn visual links off for each plane, which means that no visual links for that plane will be visible (Figure 5.5c, Figure 5.5d). We introduced this option to avoid obstructions if there are many probes and planes and to improve readability. Figure 5.5 illustrates the feature for turning visual links off.

5.4.3 Spatio-Temporal Exploration

Besides comparing different regions, the domain experts are also interested in observing attribute value changes of the simulation in time (**T6**). Therefore, we implemented functionality similar to the *analyze regions tool*, called *analyze time tool*, to allow for the spatio-temporal exploration of the data. The animation of the data simulation consists of multiple frames, each with varying attribute values and point positions. With the *analyze time tool*, the user can select a region/probe and compare attribute changes and differences in time. Differently from the *analyze region tool*, the plane this time displays box plots for different frames of a given region. The currently selected probe is colored in orange, and there is only one visual link, colored in white to indicate the plane and the region of investigation. The main controller provides options to play, pause, stop the animation and switch between frames, similar to video player controls. Once a probe is selected, the active frame at the time of the selection is displayed as a box plot in the plane (Figure 5.6b). In order to add box plots for other frames, the user can employ the video player options to switch between frames and select an additional frame to be displayed (Figure 5.6c, Figure 5.6d). No matter the order of adding frames, they are always displayed in the correct time sequence on the plane. If a frame already exists on the plane, it will not be added twice since the box plot would contain the same information. The *analyze time tool* allows users to observe and investigate changes at different locations and times during and after the impact.

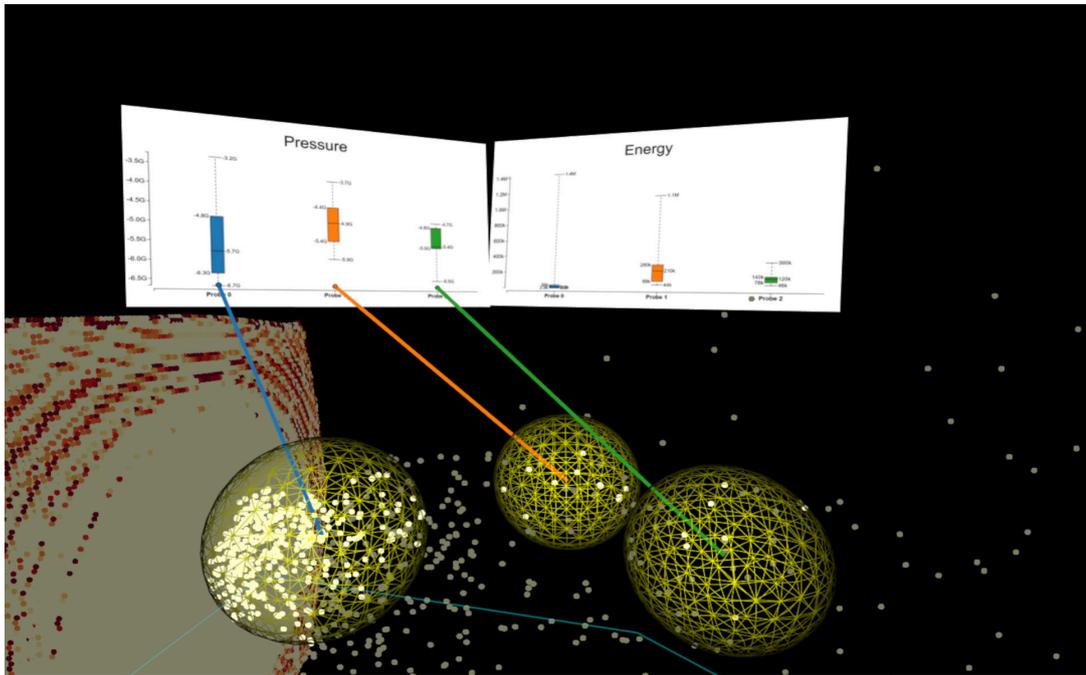


(a) Visual links connecting a box plot to a probe, colored according to the box plots

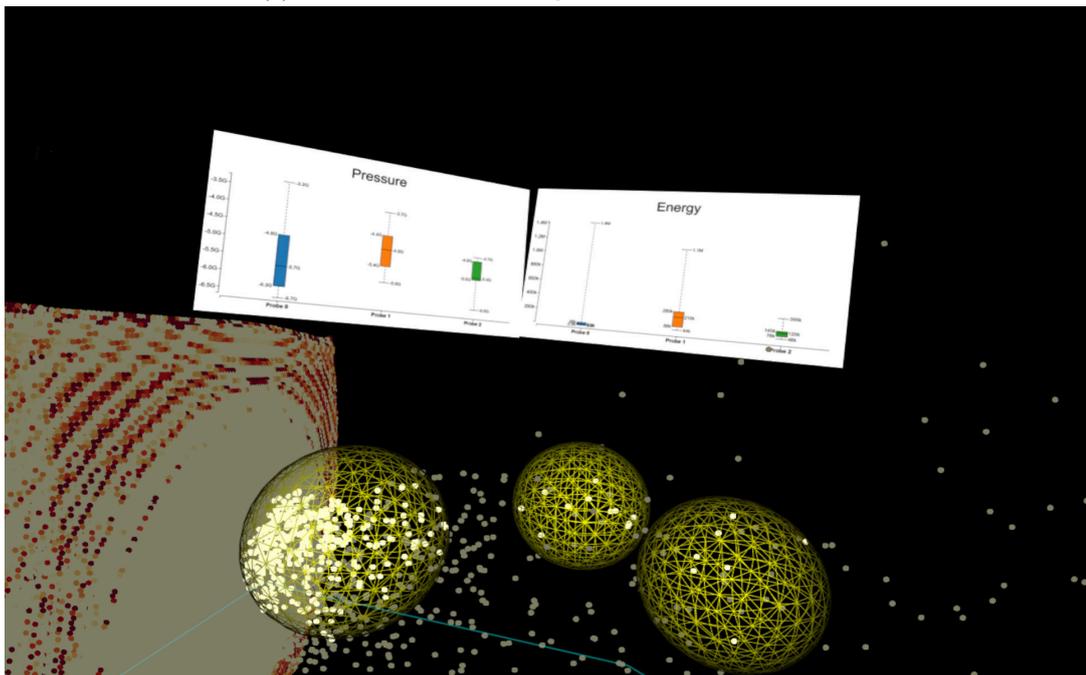


(b) The main controller is used to turn the visual links off for one of the planes

Figure 5.5: Example of turning visual links off



(c) Visual links for the right plane are turned off



(d) Visual links for both planes are turned off

Figure 5.5: Example of turning visual links off

5.4.4 Additional Features Supporting Exploration

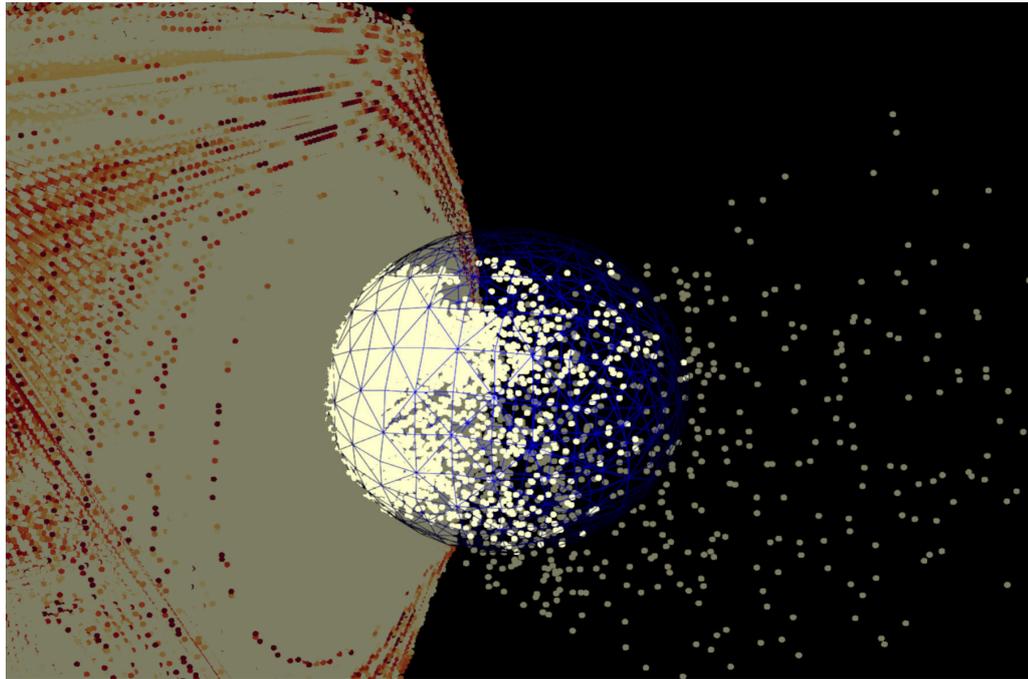
The previously described tools cover the needs for solving tasks **T4-6**. In this section, we discuss the improvements and extensions to the basic 2D desktop implementation for solving tasks **T1-3**. In our VR application, we introduce the *clipping tool*. With this tool, the user can activate a clipping plane attached to the controller and, in this way, cut the data from any position and angle and observe the inner structure (Figure 5.7), which is otherwise obstructed by the outer points.

The 2D desktop implementation includes multiple options that are particularly useful for exploration and the given tasks, so we decided to incorporate them in our VR version of the system. However, in VR we extend the functionality of the 2D implementation with new features. These features are displayed on a 2D UI placed on a 3D TV screen model as a browser texture, which is interactive. The 2D UI with its additional functionality can be seen in Figure 5.8. In order to interact with the 2D UI in VR, the user should activate the *ray tool* and use the controllers to turn on/off the different options. Some of the basic options are explained in Chapter 4. We introduce additional features that provide further interaction, visualization, and rendering options in the current work. The 2D UI is organized into five different parts according to the purpose of each option.

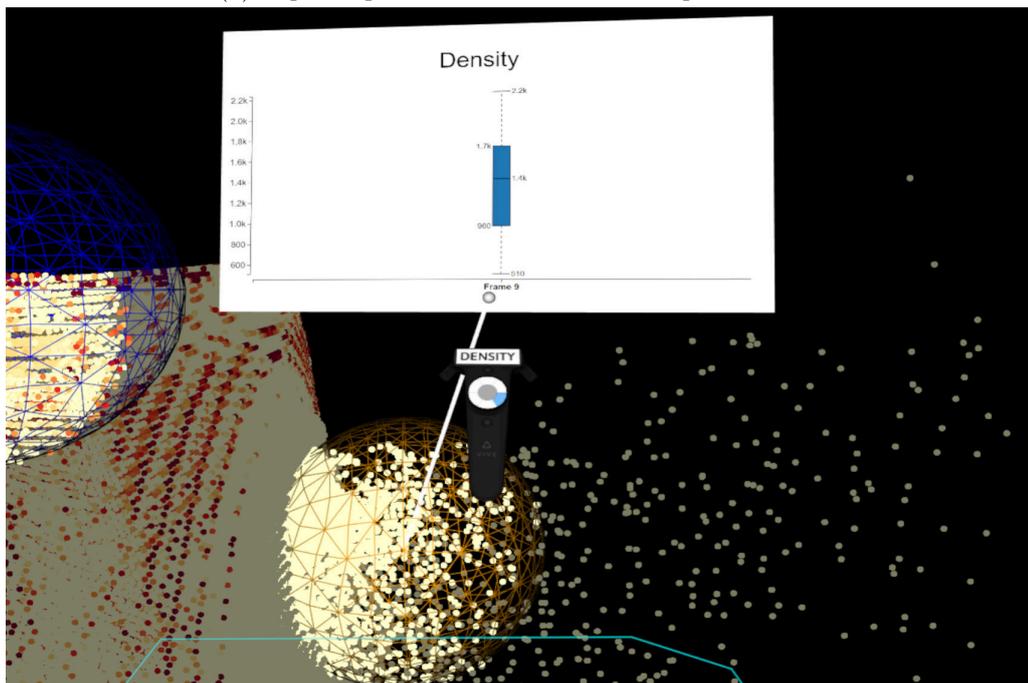
The **Appearance** area provides options for modifying the look of the 3D point cloud by changing the point size, selecting a different color map (Figure 5.9a), or attribute, and starting the simulation. Alternating these options influences the visual properties of the points and is therefore helpful for solving tasks like **T1**.

With the **Clipping Options** the points can be reduced by using axis-aligned clipping planes (Figure 5.9b). This is particularly helpful for solving tasks **T1** and **T3**. The sliders are used to control the position of a clipping plane according to an axis. The checkboxes for each axis allow to invert it, meaning that the clipping will be applied from the opposite direction so that the previously clipped points will be visible and vice versa.

The **Shading** options are used to control how the user perceives the data points. With the "Normalize Data" checkbox, the current attribute values are converted into the range from 0 to 1. This option also affects how the color map is applied, which is particularly useful for some attributes and their mapping. The "Enable Shading" option is used to mimic a realistic 3D sphere rendering of the data points. Without this option, the data points are perceived as 2D flat circles facing the camera, which is unsuitable for perceiving the structure, especially at regions with the same color. Therefore, a realistic 3D sphere rendering improves point perception with the help of precomputed shading and lighting. The method to achieve realistic 3D spheres is called impostor rendering and uses flat billboards. Additionally, the user can enable the "Reconstruct Depth" option to recompute depth values according to the point positions and the viewing position. Ensuring correct depth at each fragment is required in order to avoid overdraw and reduce "popping" effect [DST19], which is resulting from flat billboards that occlude neighbors. The shading options are a valuable addition to help users investigate the structure and placement of the points (**T1**). Figure 5.9c illustrates an example for applying the shading

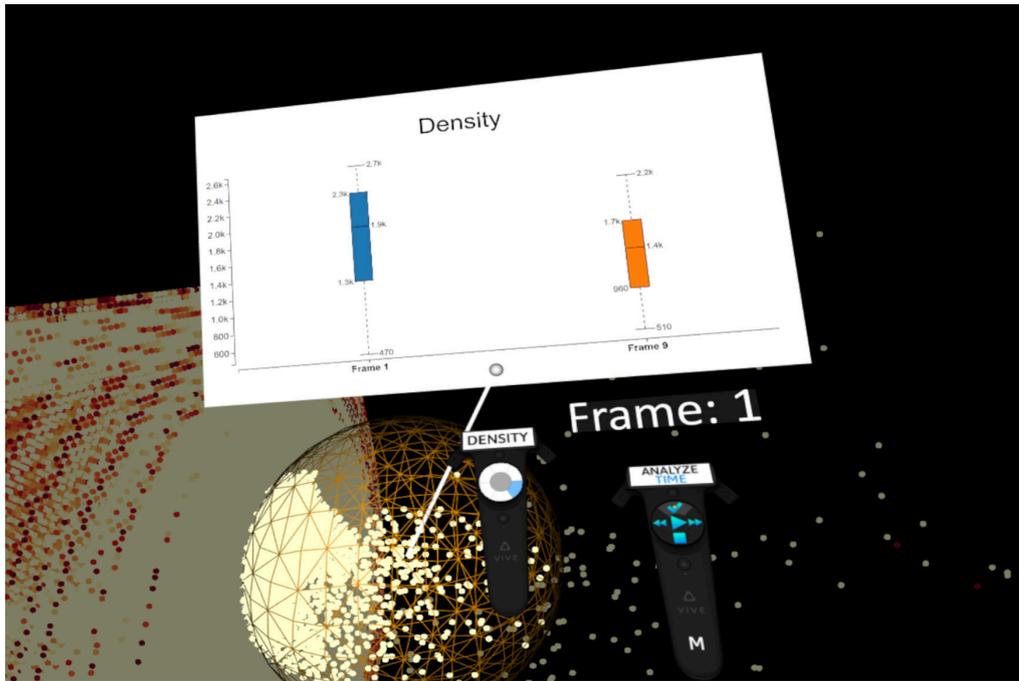


(a) A probe placed at the center of the point cloud

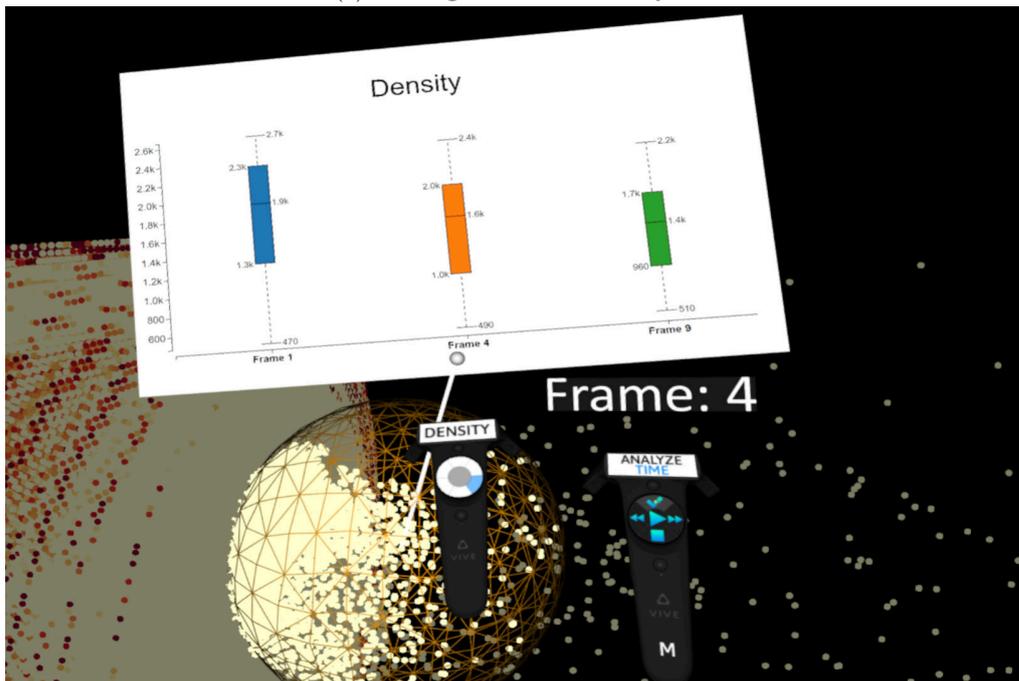


(b) Selecting the probe to analyze the density attribute value changes at the given region. Frame 9 is added for analysis as it is the active frame at the moment of selection

Figure 5.6: Example of using the *analyze time tool*



(c) Adding frame 1 for analysis



(d) Adding frame 4 for analysis

Figure 5.6: Example of using the *analyze time tool*

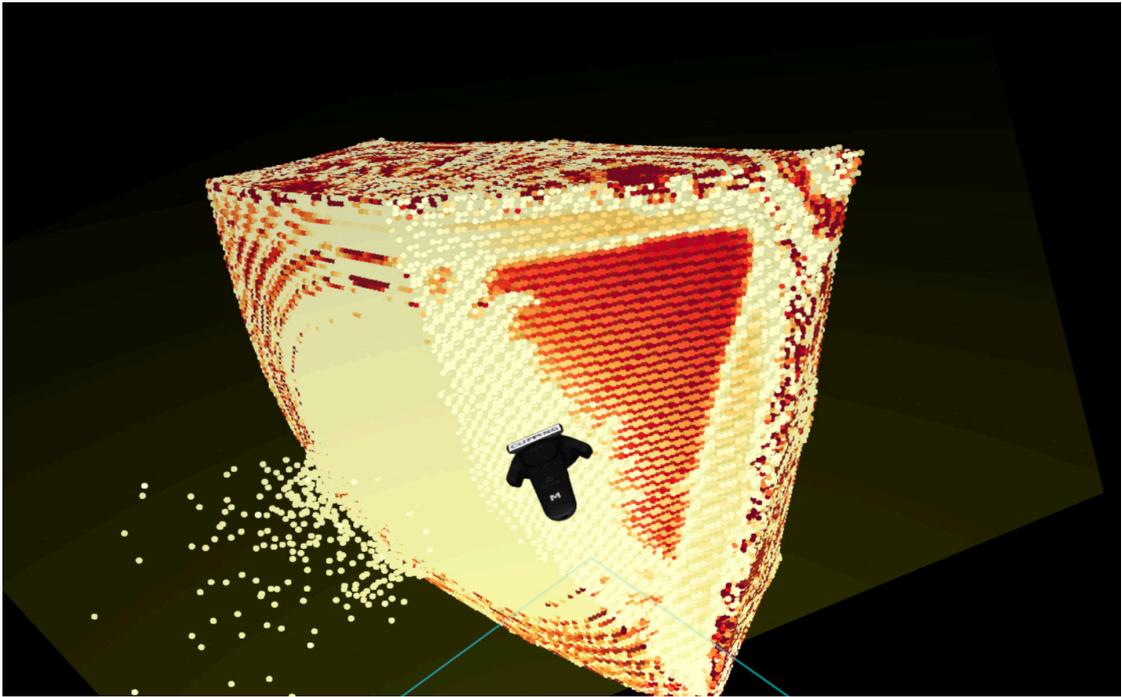


Figure 5.7: The *clipping tool*, used to cut through the 3D point cloud and observe the inner structure

options to modify the appearance of the points.

The "Enable Transparency" option in the **Transparency** panel will trigger alpha blending, which will result in partial or complete transparency of the points. Instead of mapping color to the attribute values of the points, with transparency, the mapped value is alpha, which ranges from 0.0 (fully transparent color) to 1.0 (fully opaque color). Transparency can help users to easily detect inner parts and regions with high or low attribute values. It still preserves the whole structure of the point cloud and is, thus, beneficial for all three tasks **T1-3**. Our transparency panel offers various properties that can be modified to change the mapping or transparency features. These options will be described in detail in Section 6.2.

The **Discard Options** panel provides graphical widgets to discard points that do not lie in a particular range for the given attribute or to color them with a uniform color (Figure 5.9e). Additionally, there are options to detect the lower and higher outliers in the data [HPPS18] and either discard them or highlight them with a color from the color picker (**T2**, **T3**).

Lastly, in the bottom right corner, the white plane can either display a histogram or a parallel coordinates plot as it has already been described in Chapter 4. In our VR application, a histogram plot can be triggered by either clicking on one of the probe buttons at the bottom of the histogram plane or placing a probe with the controller in

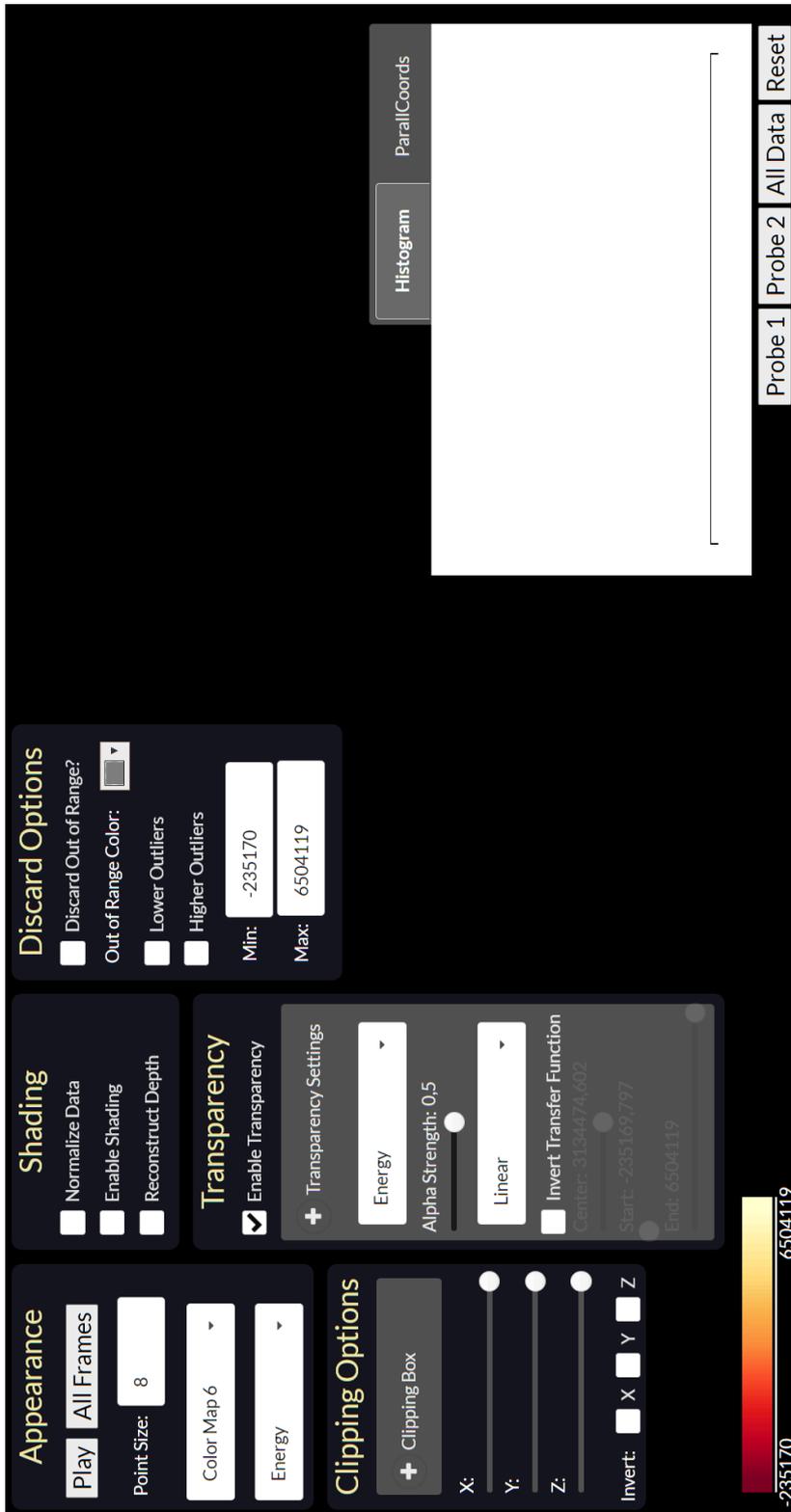
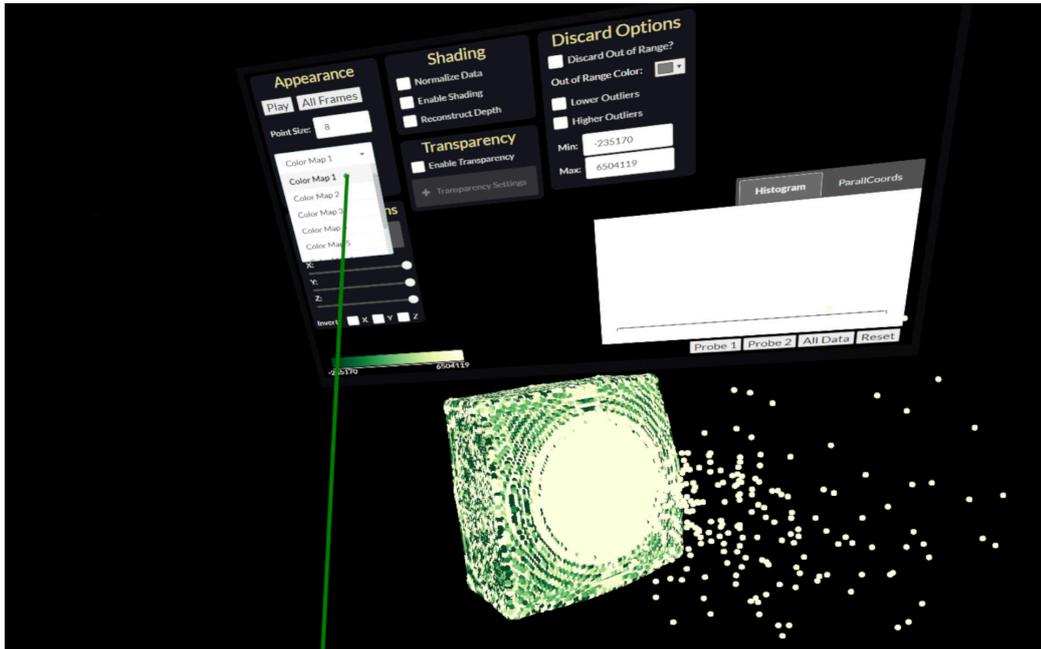
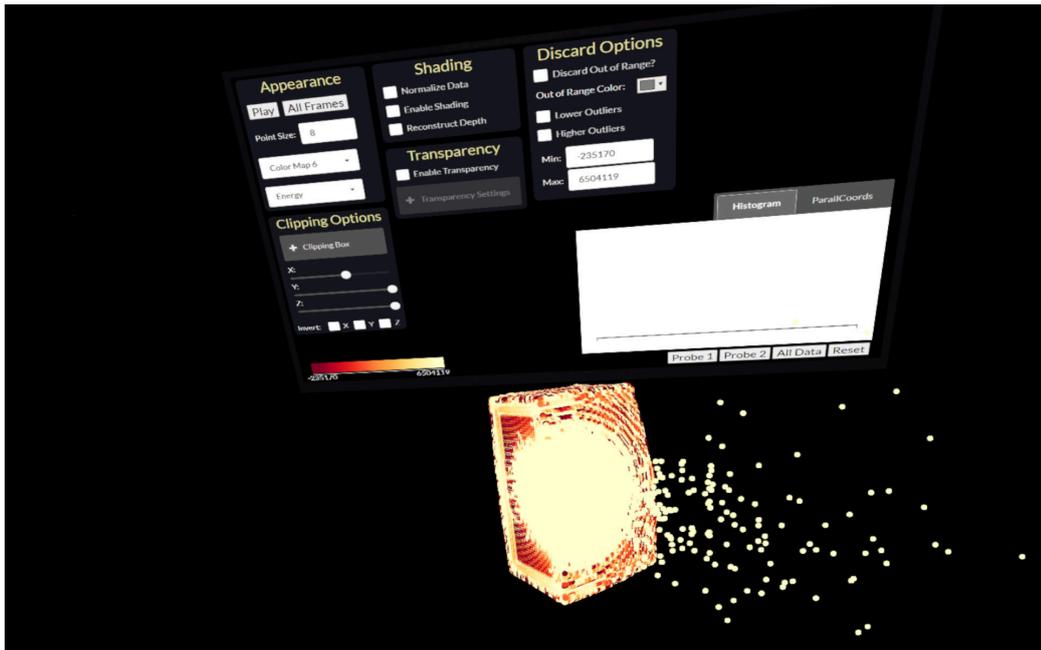


Figure 5.8: The 2D UI that is extended with additional functionality for the Virtual Reality system (see basic version in Figure 4.1)

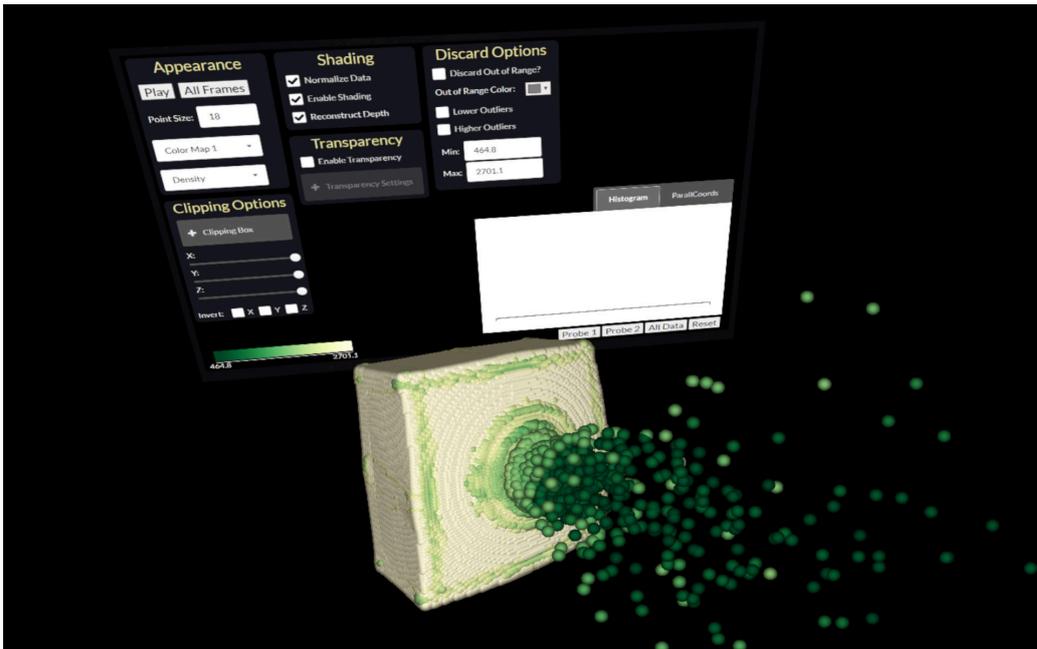


(a) Using the dropdown menu to change the color map from the **A**pppearance panel

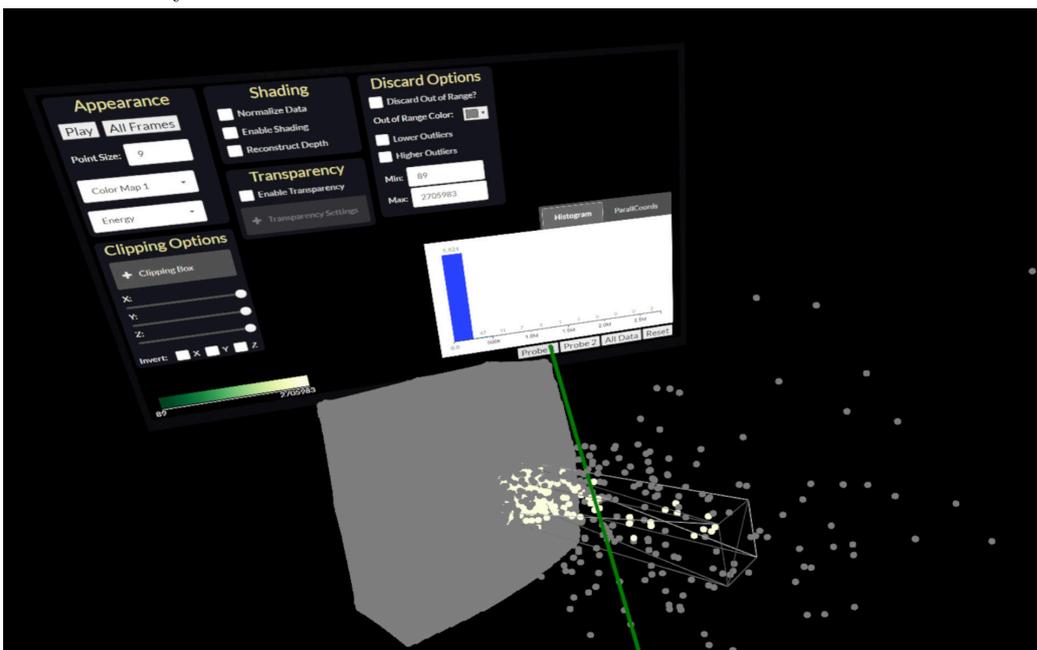


(b) Using the axis aligned clipping planes from the **C**lipping panel to cut the visualization in X and Y directions

Figure 5.9: Using the TV screen 2D UI to modify various properties of the point cloud visualization.



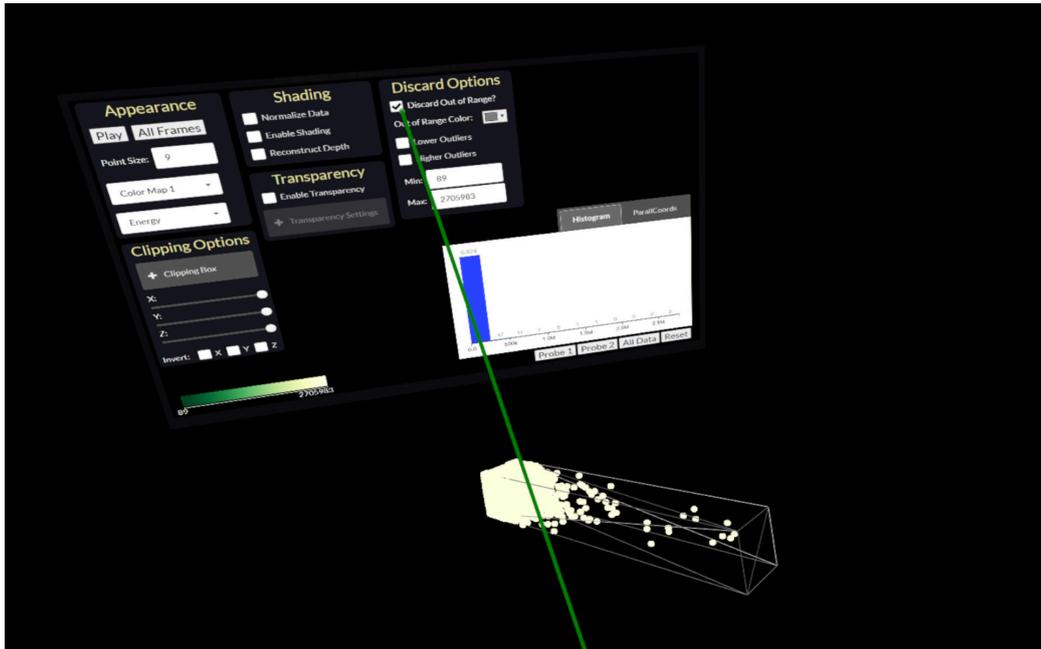
(c) Enabling shading and reconstruct depth. Data is also normalized and point size is increased for better visibility



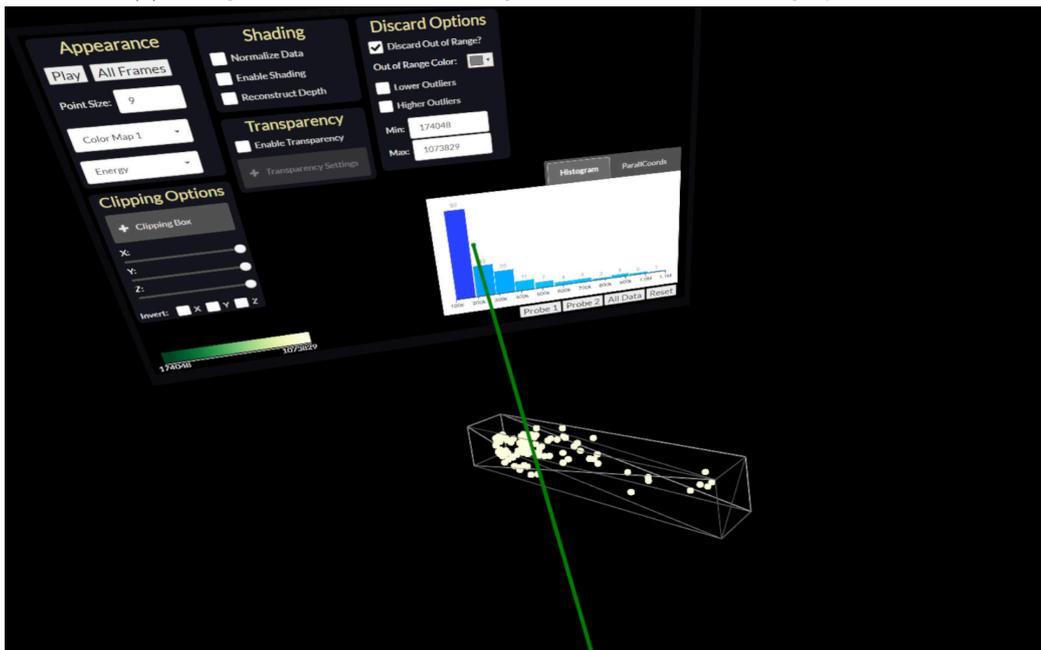
(d) Probe 1 button is activated to display a predefined fixed cuboid probe. The histogram is computed for points inside the probe and all points outside the cuboid are colored in gray

Figure 5.9: Using the TV screen 2D UI to modify various properties of the point cloud visualization.

5. IMPACT SIMULATION DATA IN VIRTUAL REALITY



(e) Using the discard out of range option to remove the gray points



(f) Applying histogram brushing to filter out data in a specific range according to the attribute Energy

Figure 5.9: Using the TV screen 2D UI to modify various properties of the point cloud visualization.

the scene. In the former case, the probes are box-shaped with a fixed position at the crater of the visualization, as can be seen in Figure 5.9d. In the latter case, a histogram is automatically displayed once a sphere probe is placed in the 3D scene. The fixed probes are cuboids since the point cloud itself has a cuboid shape, and domain experts wanted to observe data properties at the crater for the whole depth of the point cloud. For the movable probes in VR, we use a sphere because it does not have edges and does not require rotation transformations, therefore, it is more simple and intuitive to place and transform. The histogram displayed on the 2D UI shows only the distribution of the points of the probe placed last. This histogram, differently from the histograms above each probe, is not static, meaning that it can be brushed to filter out data and reduce points (**T3**). Applying histogram brushing to reduce the points can be seen in Figure 5.9f. The parallel coordinates plot is limited only to its basic functionality, as it is not the main focus of our work.

5.5 Interaction with the Impact Simulation Data in VR

5.5.1 Interactions Overview



Figure 5.10: HTC Vive hardware components [Inc15]

Interactions are crucial because they allow the communication and connection between the user's intention and the immersive system's response. For this work, we use an HTC Vive system consisting of a VIVE VR Headset, two controllers, and two base stations to track the devices in the 3D space. Figure 5.10 shows the components of the HTC Vive system. We adapt some of the 3D UI interaction metaphors by LaViola et al. [LKM⁺17], but also extend and add additional features. The interactions are split into three main groups according to the type of interaction - navigation in various menu types, object manipulation, and movement in the 3D space. In the next paragraphs, we provide information about our interaction adaptations and design decisions. Before going into detail, however, it is vital to know the components and name conventions of the controllers' buttons. Figure 5.11 illustrates an HTC Vive controller. Four buttons are used for interactions - menu button, trackpad, trigger button, and the grip buttons.

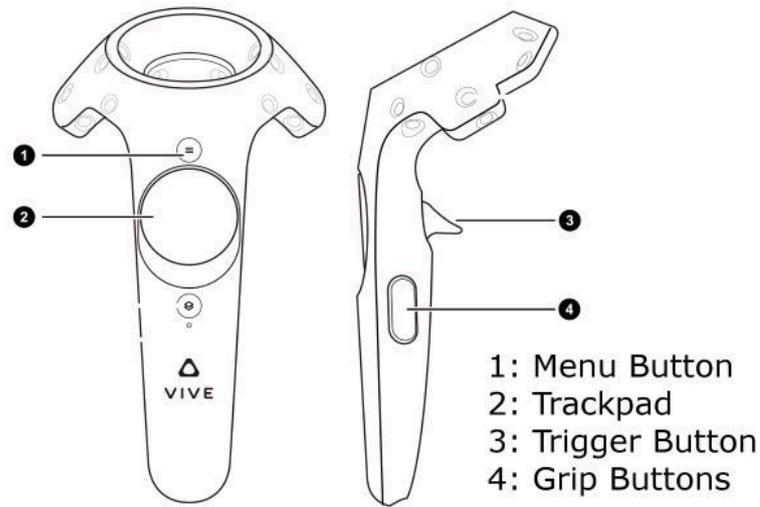


Figure 5.11: HTC Vive Controllers and Buttons [Var19]

5.5.2 Menu Types and Navigation

In our Virtual Reality application, there are two different menu types. The first one is the 2D UI shown on a TV screen in the 3D environment (Figure 5.8). The second menu type is a radial menu attached to the controller (Figure 5.12). The 2D UI provides options for modifying various features of the point cloud visualization. The necessary interactions to use the different options of the 2D UI will be described in detail in the following sections.

In our system, there is a difference between the two controllers since they offer different functionalities. Therefore, we have the main controller and the secondary controller. The main controller is indicated with the "M" letter placed at the bottom part as it can be seen in Figure 5.12. Both controllers have a radial menu which is displayed on the trackpad. The main controller is used to switch between different interaction tools and the secondary controller to change the attribute. The radial menu on the main controller is always opened when the user clicks on the trackpad button. Four different tools are then displayed, and the user can switch between them just by touching the respective field, as can be seen in Figure 5.13. The user needs again to click on the trackpad button once the desired tool is chosen in order to close the menu and be able to use the tool. The menu can always be opened by again clicking on the trackpad. The four different tools the user can choose from are:

- **Probe** (see Figure 5.13a) - this tool is used for creating and manipulating probes in the scene. A probe is a sphere with a modifiable size that can be placed at a desired location and represents statistical information about the point cloud points lying inside it.

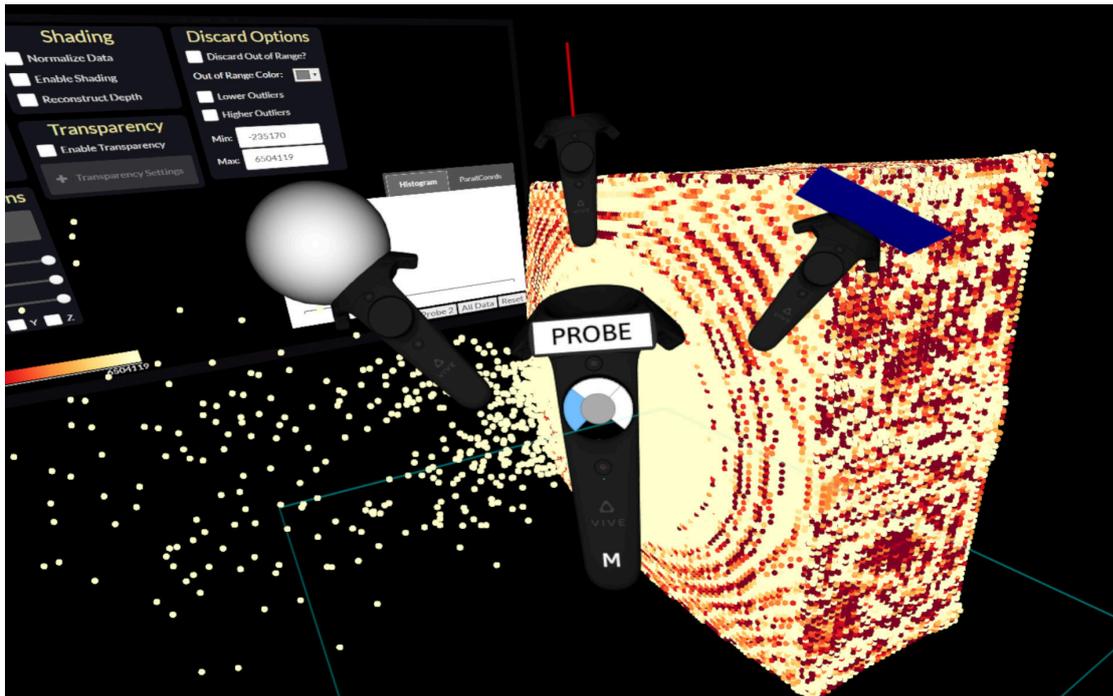


Figure 5.12: Radial menu on the main controller to switch between tools

- **Ray** (see Figure 5.13b) - represents a laser used to select or change options on the panel menu placed on the virtual TV screen. The laser is green when it points to the TV and red otherwise, indicating when it is possible to interact with the TV.
- **Clipping** (see Figure 5.13c) - this tool activates a clipping plane that is always attached to the main controller and can be used to cut through the point cloud visualization. This way, it is possible to observe the inner parts of the visualization.
- **Analyze Probes** (see Figure 5.13d) - offers a tool to compare different regions/probes or time frames at a given region. Accordingly, there are two options - analyze regions and analyze time.

To be aware of what tool or option is currently active, the controllers also have a text screen above the menu button displaying the necessary information (Figure 5.13). One of the main functionalities of the secondary controller is to change the attribute. As shown in Figure 5.14, the attribute menu is again a radial menu, which works only with a touch to switch the attribute. The attribute menu is opened automatically in three different cases. First, when the *probe tool* is active, the user can select a *global* attribute on the secondary controller, meaning that all further probes will inspect the data according to the selected attribute (Figure 5.14). Second, when the user intersects a probe with the main controller, he/she can change the respective *probe* attribute. Third,

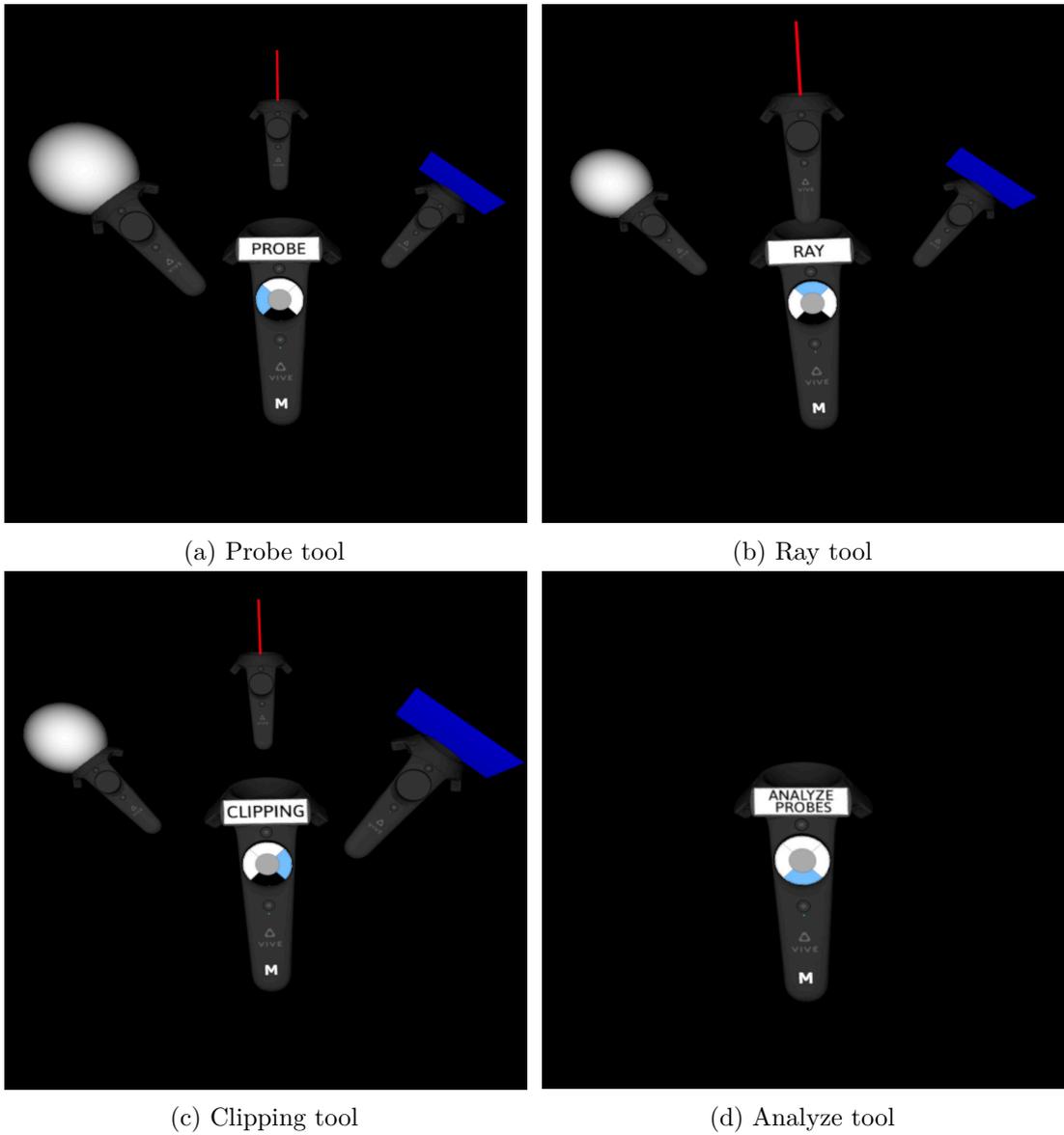


Figure 5.13: The radial controller menu on the main controller to switch between different tools

the *analyze probe tool* activates a *box plot* panel with a particular attribute, which can also be modified.

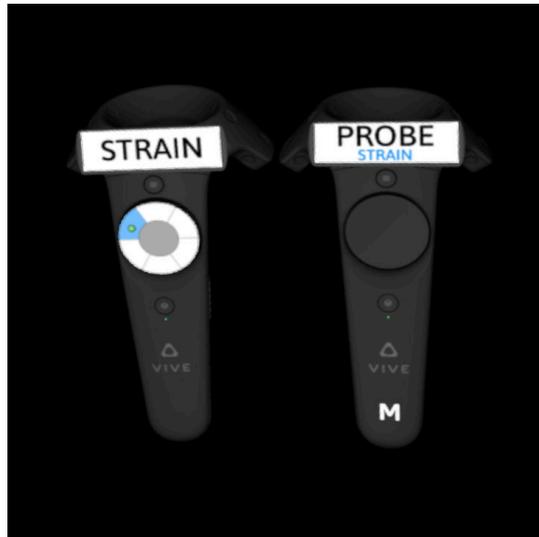
5.5.3 Object Manipulation

Object manipulation is another essential interaction type in the given system. Therefore, we provide and adapt two different types of grasping techniques inspired by LaViola et



(a) No global attribute selected for probes

(b) Changing the attribute to energy

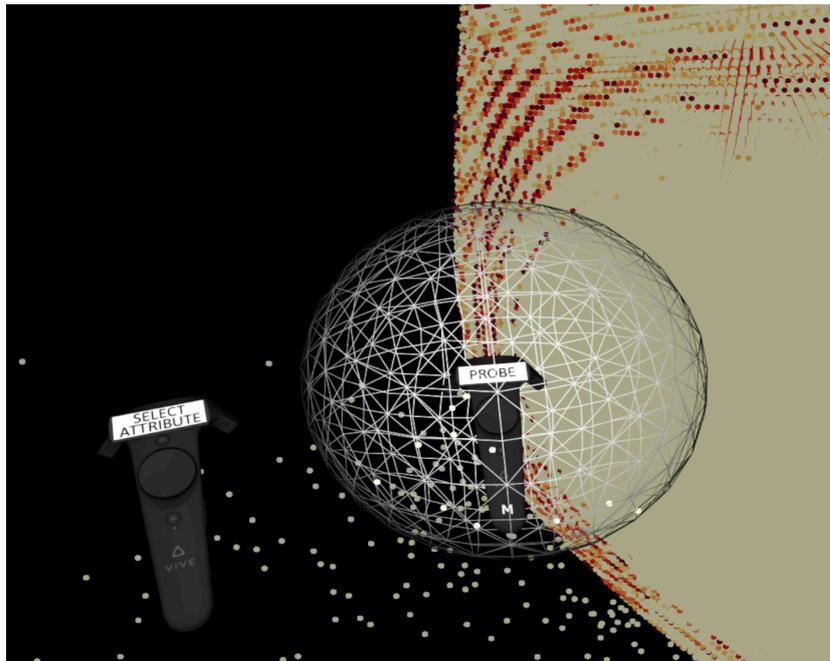


(c) Changing the attribute to strain

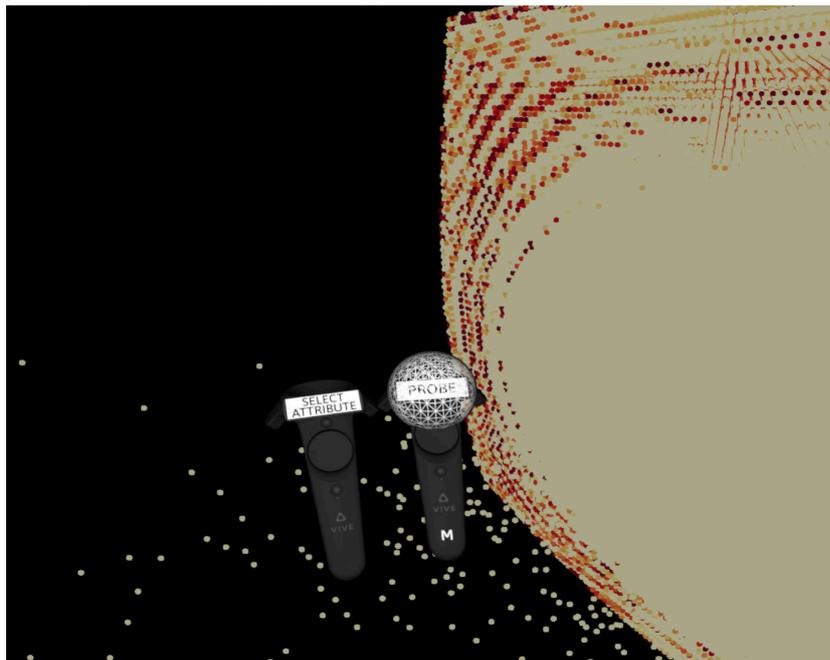
Figure 5.14: The radial attribute menu on the secondary controller to switch between different attributes

al. [LKM⁺17]. We introduce close-range grasping techniques to directly interact with objects in the proximity and long-range pointing techniques involving distant interaction with objects.

The close-range techniques are performed with the trigger. With the trigger button of the main controller, the user can activate and use the currently selected tool. In the case of a *probe tool* - clicking the trigger creates a probe. The user can place it at a

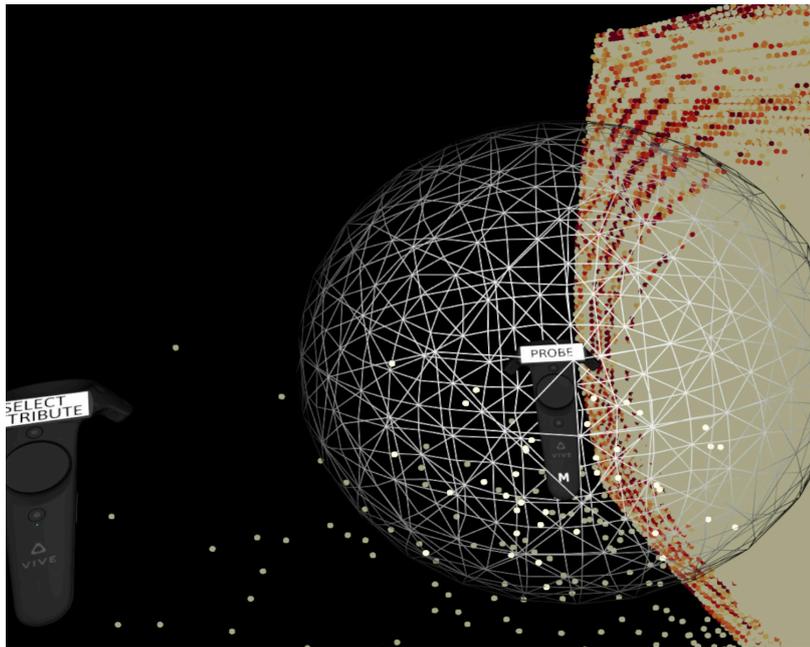


(a) The probe is created, the trigger of the second controller is not pressed



(b) The probe is smaller if the distance between the controllers is smaller

Figure 5.15: Rescaling a probe depending on the distance between main and secondary controller

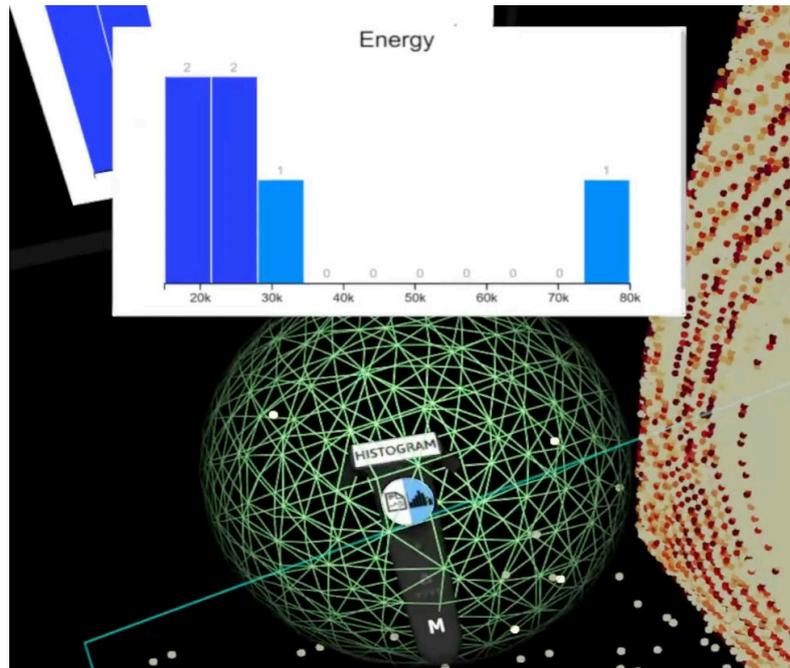


(c) The probe is scaled up if the distance between the two controllers is increased

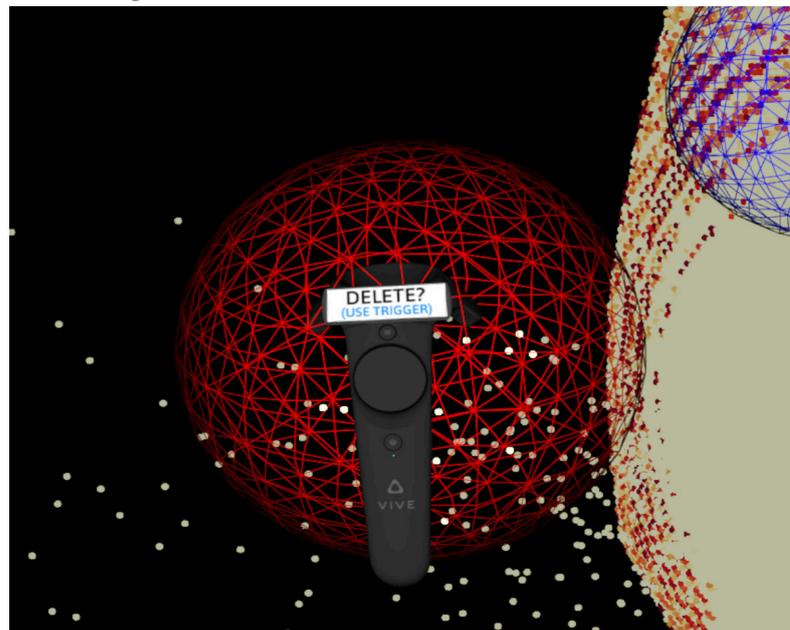
Figure 5.15: Rescaling a probe depending on the distance between main and secondary controller

desired location in the 3D space when releasing the trigger. While holding the trigger on the main controller, the user can also simultaneously click the trigger on the secondary controller to modify the size of the probe. The size depends on the distance between the two controllers - the bigger the distance, the bigger the sphere, and vice versa. Figure 5.15 illustrates the probe resizing depending on the distance between both controllers. Once a probe is placed, the user can always retake it and modify it. This interaction happens if the main controller intersects with the probe (it changes its color to green to identify the intersection) and the user presses the trigger (Figure 5.16a). If the secondary controller intersects with a probe, it is colored in red, indicating the user can delete it by pressing the trigger (Figure 5.16b). The text screen on the controller also indicates that the sphere can be deleted with the trigger. In general, every object in the scene (probe, clipping plane, box plot) except for the point cloud visualization and the TV screen can be deleted if intersected and at the same time the trigger of the secondary controller is pressed.

If the *ray tool* is selected, the trigger on the main controller is used to activate the ray. After the ray is activated, the user can change different settings on the TV screen's 2D UI with it. The trigger mimics the left click on a standard mouse device, and the touchpad is used as the mouse's wheel. If the *clipping tool* is currently selected, the trigger activates a clipping plane attached to the controller as long as the trigger is pressed. Once it is released, the clipping plane preserves its last position in the 3D space. Similar to the



(a) Sphere color is green if it is intersected with the main controller, indicating that it can be grabbed and modified



(b) Sphere color is red if it is intersected with the secondary controller, indicating that it can be deleted by pressing the trigger

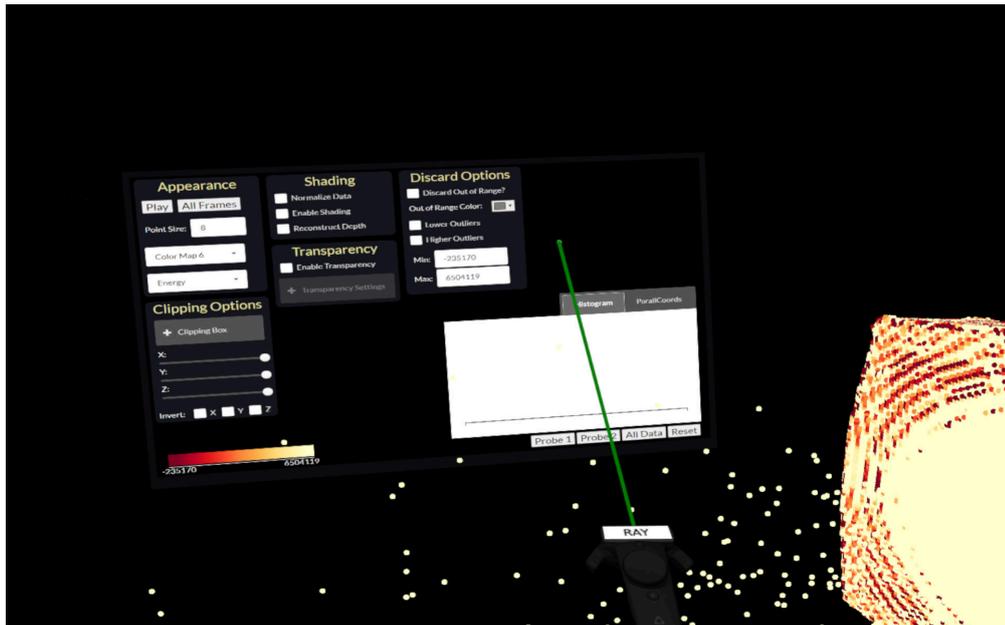
Figure 5.16: Intersection of a probe with a controller indicated by the sphere color

probe tool, the clipping plane can be retaken (no need for intersection, though) with the main controller, and the secondary controller can delete it. Finally, the trigger is used to select probes for the *analyze probe tool*, again when intersecting. A box plot is displayed on a plane attached to the secondary controller for each of the selected probes. After the desired probes are selected, the box plots plane can be placed anywhere in the 3D environment by pressing the trigger on the secondary controller. As with the probe, the user can delete the placed plane by intersecting with the secondary controller and pressing the trigger. If intersecting with the main controller, the user can grab the plane and change its position, but can not modify the information.

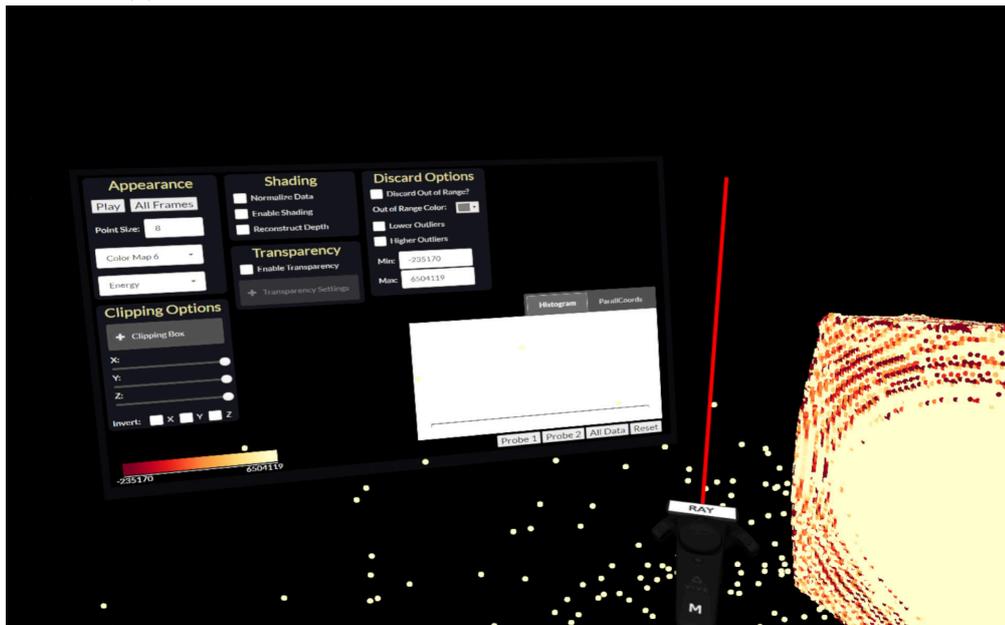
Although the ray is activated with the trigger, the *ray tool* is, in fact, a long-range pointing technique since the user can operate it from any distance. Once the ray is activated and is pointing to the screen, the trigger can be used as the left mouse button to control and modify different options of the 2D UI on the TV screen. The ray changes its color to indicate whether it is pointing to the screen. If it is, then it is colored in green, and screen interactions can be performed. If it is pointing outside the boundaries of the TV screen, it turns red to indicate that no UI modifications are possible (Figure 5.17). The rest of the long-range pointing techniques are performed with the grip buttons of the main controller. Regardless of the currently activated tool, the grip buttons are always used to grab and move the point cloud visualization. Consequently, the controller does not need to intersect with the data points since the grabbing works from any distance and orientation. This way, the user can move the visualization together with the probes and clipping plane from everywhere. In Figure 5.18a the user is pressing the grip buttons of the main controller and can therefore move the point cloud visualization anywhere in 3D space and place it once releasing the grip buttons. While pressing the grip buttons, the scene can also be scaled up and down, which happens by touching the left or right half of the trackpad as can be seen in Figure 5.18b and Figure 5.18c. The closer to the left/right border of the trackpad the user touches, the faster the scaling is performed. The same grabbing and scaling technique is used for the TV screen. However, it is only possible when the *ray tool* is activated, and the ray is pointing to the screen. Only then can the TV be moved and rescaled. In any other case, this interaction technique is used to transform the point cloud visualization.

The close-range object manipulation techniques provide a much more precise movement and fine-grained placement than the long-range pointing techniques. Therefore, they are suitable for objects like the probes and the clipping plane, where accuracy plays a significant role. On the other hand, the long-range pointing techniques allow users to rapidly relocate, rotate, or scale the whole visualization and the TV screen from anywhere in the 3D space. Consequently, each technique in the system is selected and adapted according to the requirements of the specific task so that the combination of both interaction techniques provides more precision, flexibility, and faster object manipulation.

5. IMPACT SIMULATION DATA IN VIRTUAL REALITY

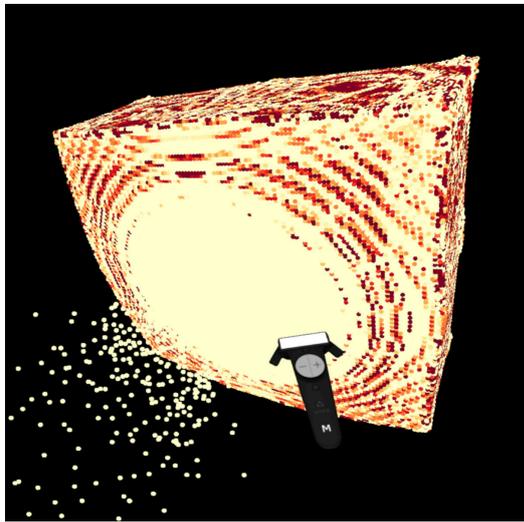


(a) The color of the ray is green if it is pointing to the TV screen

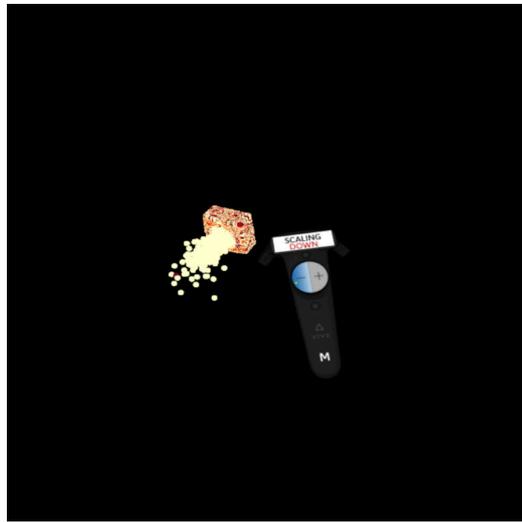


(b) The color of the ray is red if it is pointing outside the boundaries of the TV screen

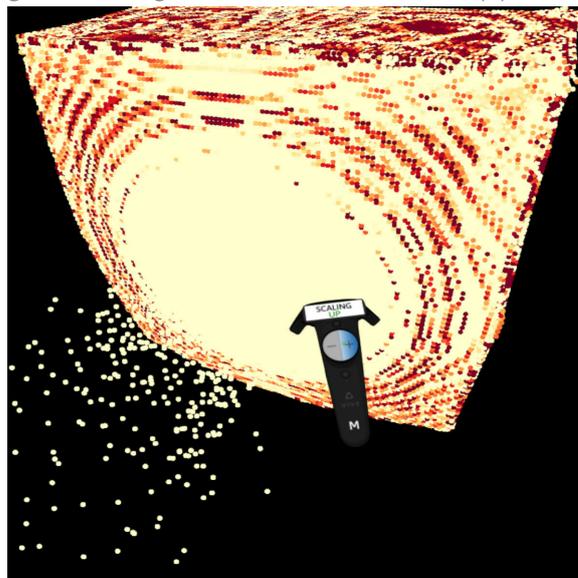
Figure 5.17: Ray color change according to the intersection with the TV screen



(a) Grabbing and moving



(b) Scaling down



(c) Scaling up

Figure 5.18: Grabbing and scaling up and down the point cloud visualization

5.5.4 Movement in 3D Space

Navigation and traveling is another crucial interaction type in immersive environments. Our system provides one of the full gait techniques - physical walking. As LaViola et al. [LKM⁺17] describe, physical walking is the most natural technique since the user can walk and move in the same way as in the real 3D world to travel from one location to another one. This full gait technique is possible because the *Impact simulation data* visualization environment is restricted to the point cloud visualization, the TV screen, and all additional objects placed within and around the point cloud. Therefore, the VR environment does not require specific traveling metaphors. For the application, a physical room with around 9 m² in size is sufficient. Not only is physical walking a natural movement and requires no prior knowledge, but it is also beneficial for reducing the probability of motion sickness occurrence.

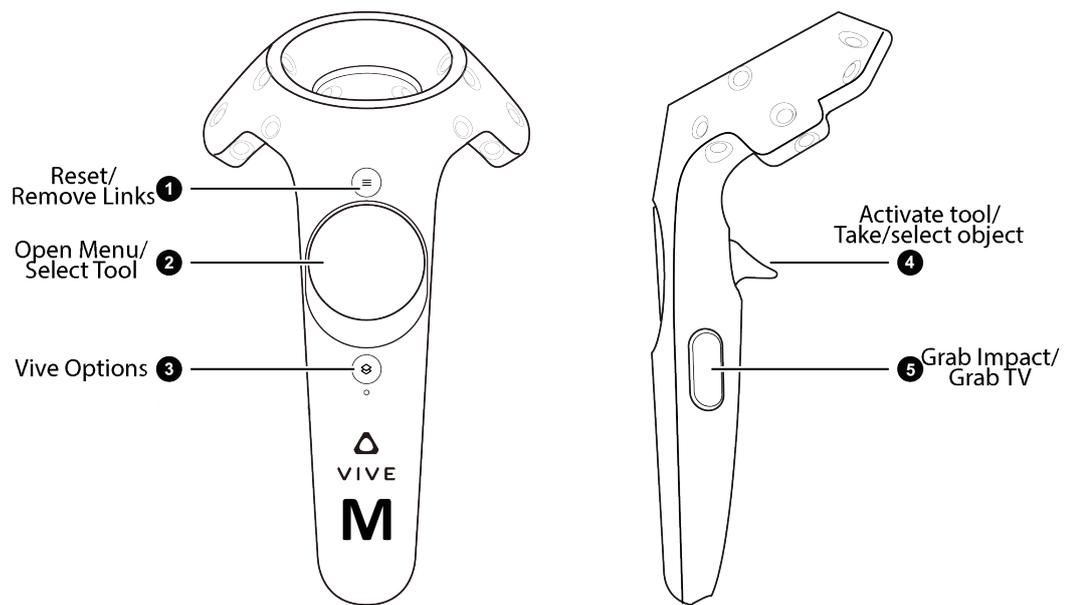
5.5.5 Further Interaction Options

The remaining interactions and options cannot be summarized in one of the categories mentioned above. Most of them are performed with the menu button. The menu button of the main controller is used to reset the whole scene. In case the main controller intersects with a box plots plane, the menu button is used to turn on/off the visual links. When intersecting with a box plot, the menu button of the secondary controller is used to copy the information to the currently active box plot. The grip buttons of the secondary controller are used to turn on/off the billboards above the probe. The trigger button of the main controller can be used as the left mouse button for controlling the 2D UI on the TV screen when the *ray tool* is active. The options on the 2D UI include buttons, dropdown menus, checkboxes, sliders, tabs, and text fields. All of them are clickable and modifiable by the trigger. The trackpad can be used as the wheel button on the mouse to scroll up/down on a dropdown menu or to increase/decrease values on the text field. Table 5.1 summarizes all possible interactions for both the main and secondary controller depending on the button and whether the controller intersects with an object. Figure 5.19 visualizes the possible interactions assigned to the buttons of both controllers.

Button	Intersect	Main Controller	Secondary Controller
Menu Button	<i>Yes</i>	Remove visual links (intersection with box plot)	Copy box plot information (intersection with box plot)
	<i>No</i>	Reset scene	-
Trackpad	<i>Yes</i>	Switch between histogram and billboard (if intersecting with probe)	Change probe attribute (if main controller intersects with a probe)
	<i>No</i>	Open menu Select tool Scale point cloud (while grabbing) Scale TV screen (while grabbing)	Change global attribute Change box plot attribute
Trigger	<i>Yes</i>	Take and move object Change 2D UI settings	Delete object
	<i>No</i>	Activate tool Create or select object	-
Grip Buttons	<i>Yes</i>	Move TV screen from distance (if ray intersects with TV screen)	-
	<i>No</i>	Move point cloud visualization from distance	Turn on/off billboards above probes

Table 5.1: All interaction options for both main and secondary controller

Main Controller

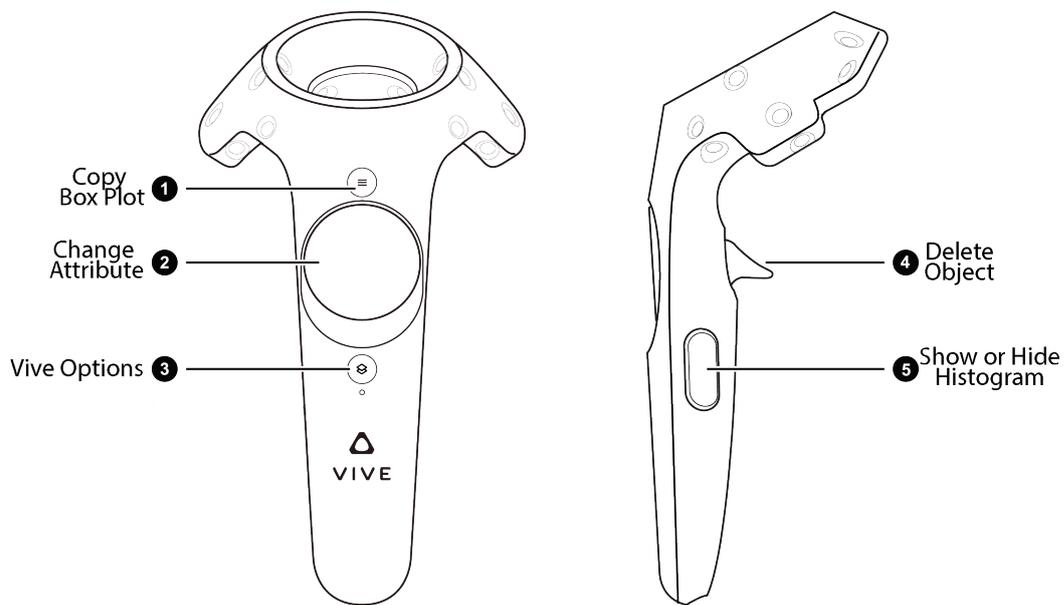


1. Reset system/Remove visual links (when intersecting with a box plot)
2. Open Menu (click)/Select Tool or option (by touching)
3. Vive System Options (not program relevant)
4. Activate currently selected tool/Grab and Place object (sphere, box plot, clipping)
5. Grab Impact simulation data (from distance)/
Grab TV screen (only when ray is active and is pointing at the TV screen)

(a) Possible interactions with main controller

Figure 5.19: The possible interactions with both controllers assigned to different buttons

Secondary Controller



1. Copy Box Plot (when intersecting with a Box Plot)
2. Change Attribute (global, for a probe and for a box plot; by touching)
3. Vive System Options (not program relevant)
4. Delete Object (sphere, box plot, clipping plane)
5. Show or Hide histogram/statistics above a probe

(b) Possible interactions with secondary controller

Figure 5.19: The possible interactions with both controllers assigned to different buttons

Perceptual Considerations and Optimizations in VR

Perception is crucial for understanding and communicating with the surrounding world, whether it is a virtual or a real one. During our implementation process, we received valuable feedback from people testing or viewing our system. An often mentioned issue was the perception of points in the 3D space since initially, we provided only 2D flat circles. As a result of the feedback, we further developed and improved multiple features of our application. These include perception enhancements, immersion improvements, occlusion handling, and motion sickness occurrence reduction. The following subsections present different solutions and rendering modes to address these topics.

6.1 Impostor Rendering of Spheres

In our basic implementation, we only rendered 2D flat circles for our point cloud visualization. Each circle always faces the camera and has a specific color according to the color map or other settings from the 2D UI. In case multiple circles in proximity have the same color, they blend, and it is impossible to observe the structure details and roughness at these regions. For this reason, we decided to render spheres instead of flat circles to depict structure roughness and bumps better.

To render a sphere in the 3D space, we can simply generate a sphere mesh and render it. However, a spherical mesh never represents a mathematically correct sphere even if we use a large number of triangles, as it will still be an approximation and will not look realistically. Moreover, having an environment with a large number of spheres using meshes would require rendering a huge set of polygons, which will cause a time overhead and is not very efficient. Spheres are, though, very simple objects. We can use another powerful, but relatively simple method to render realistic spheres. In computer

graphics, spheres can be visualized by rendering them as impostors. The technique uses view-aligned planes rendered at sphere locations and computing color and extent in a per-pixel operation to make the object look three-dimensional. An impostor is a 2D image texture mapped on a rectangular plane, called a billboard, that creates the illusion of highly detailed geometry.

In order to create realistic spheres through impostor rendering, we first use the vertex shader to compute the vertex positions of each point in the clip space. With the help of a geometry shader, we create a square at each of these positions and with the same width/height as the actual circle, meaning that as an output, we receive four vertices out of one. These four vertices represent the corners of the square, and in the impostor space, they have the coordinates $(-1, -1)$, $(-1, 1)$, $(1, -1)$, $(1, 1)$. They are stored in a variable passed to the fragment shader. We will refer to this variable as *uv_mapping*. The square is called a billboard and is always facing the camera. To form realistic 3D spheres, we then use the fragment shader. In the first step, the fragment shader discards all fragments outside the sphere, thus, forming a 2D flat circle. After that, we need to recompute the normal of each point on the sphere's surface. We do this to find the correspondence of each point in the square to a point located on the sphere surface. For this purpose, we use the *uv_mapping* variable to compute the normal for each position. The normal can be computed by using simple trigonometry. We know the distance from each position in the square to the center. We also know that the radius in the space of the *uv_mapping* is normalized and therefore has a value of 1. We also know that the normal at each point is pointing in the same direction as the vector from the sphere's center to that point on the surface. The normal computation can then be expressed mathematically as: $n = (u, v, \sqrt{r^2 - |(u, v)|^2})$, where n is the recomputed normal, u and v are the first and second coordinates of the *uv_mapping* and r is the radius.

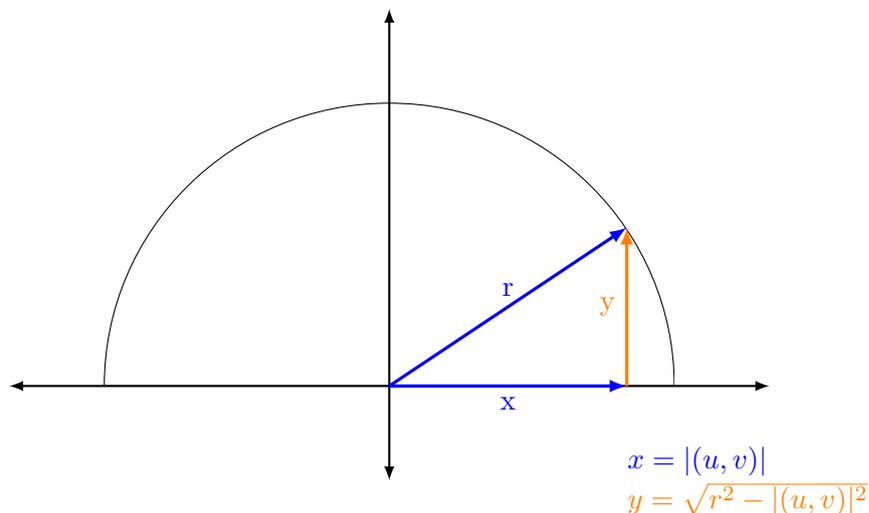


Figure 6.1: Normal computation in 2D space

Figure 6.1 illustrates the computation of the new normal in 2D space for better understanding. We are searching for a 2D direction, so we need x and y , where $x = |(u, v)|$. As we know x , and we also know the length of the vector r , we can easily compute y with the Pythagorean theorem: $y = \sqrt{r^2 - |(u, v)|^2}$, and so we have our new normal $n = (x, y)$ for the 2D case.

After having the correct normal, we can now use the Blinn-Phong shading model to compute the final color for each fragment. For the computation, we use a light source located at the camera position. This ensures that the point cloud is illuminated from every viewing direction. After these computations, we have 3D spheres at each point. However, the point cloud visualization might not look quite correct and realistic. This is because we also need to recompute the depth value in the fragment shader. In order to do that, we compute each position on the sphere in clip space by using the projection matrix. Having the position in clip space, we can use perspective division to transform depth to normalized device coordinates (NDC). By using the shader read-only intrinsic variable `gl_DepthRange`, which describes the range of depth values in the graphics pipeline, we can compute the depth value, as if the spheres were rendered using polygons. The computations of the depth in the shader can be seen in Listing 6.1.

```
uniform mat4 projectionMatrix;

void main()
{
    vec4 posOnSphere = spherePos + normal*sphereRadius;
    vec4 posOnSphere_screen = projectionMatrix*posOnSphere;
    float ndc_depth = posOnSphere_screen.z/posOnSphere_screen.w;
    float depth = ((gl_DepthRange.diff * ndc_depth) +
                  gl_DepthRange.near + gl_DepthRange.far)/2.0;

    gl_FragDepth = depth;
}
```

Listing 6.1: Fragment Shader - Depth Computation

The impostor rendering of the point cloud can be activated through the 2D UI on the TV screen. With the option "Enable Shading", the points are rendered with impostor rendering and look like 3D spheres. The "Reconstruct Depth" option is used to recompute the depth, bringing more realistic results, with correct depth from any viewing angle. Impostor rendering helps the visualization to gain more granularity and brings users a better awareness of the structure and finer details. Figure 6.2 shows the results of rendering the point cloud as flat circles, with enabled shading, and with shading and reconstructed depth.

6.2 Transparency with Alpha Blending

The impostor rendering option helps users receive more visual information about the outer structure of the data. Nevertheless, the inner parts of the point cloud still remain obstructed. Therefore, we provide another rendering mode that can enable insight into the inner parts of the visualization while still preserving the structure. This problem is not new and can be found in volume rendering, which uses a transfer function to map volumetric attributes to RGB colors. This way the opacity of particular parts can be controlled. In our case, we map a color to an attribute value according to a color map and a transfer function. In our application, a color is represented by four components, the first three holding the RGB components, and the fourth is the alpha value. The alpha value indicates how opaque a pixel is and allows alpha blending to combine images, creating the appearance of partial or complete transparency.

There are different ways on how to render transparent objects. One of the most famous methods requires rendering geometry in sorted order (front-to-back or back-to-front), which can cause computational overhead or lead to problems in the case of overlapping geometry for more complex scenes. However, many techniques are already used for transparency that avoid explicit ordering and are summarized as order-independent transparency techniques. One such technique is depth peeling [BM08]. In our system the standard sorting technique works well and produces satisfying results efficiently since only the point cloud is transparent. We compute the distance from the camera for each point and store the new results in the index buffer for every frame. In this way, we know how points are sorted. To achieve transparency, we use back-to-front composition and additive blending.

In the 2D UI, we provide various options to control and modify different features of the transparency rendering mode (Figure 6.3). The mapping for transparency is entirely independent of the color map. The alpha value can be mapped to a different attribute than the one selected for the color map, which happens through the dropdown menu in the transparency panel. Moreover, the user can adjust the alpha strength with a slider. The alpha strength is just a multiplier to increase or decrease the transparency. The dropdown menu below the alpha strength slider is used to change the transfer function (Figure 6.3b). Depending on the selected transfer function, the transparency value will be mapped accordingly to the attribute value of each point. Our system offers four transfer functions for transparency - linear, logarithmic, exponential, and tent function. Each of the transfer functions can be inverted by clicking on the "Invert Transfer Function" checkbox. The tent function implementation is inspired by A. König and E. Gröller [KG01], but is further expanded to provide more flexibility. In our implementation, the user can modify the tent function by changing three components (Figure 6.3c). The tent function has a start value so that all points with a smaller value will have an opacity of 0. It also has a center value, where the peak of the function is located, and values having a value equal to the center value will have the maximum opacity of 1. Finally, the tent function also has an end value, meaning that all points with a bigger value will also have an opacity of 0. Points with values between the start and the center value will have an

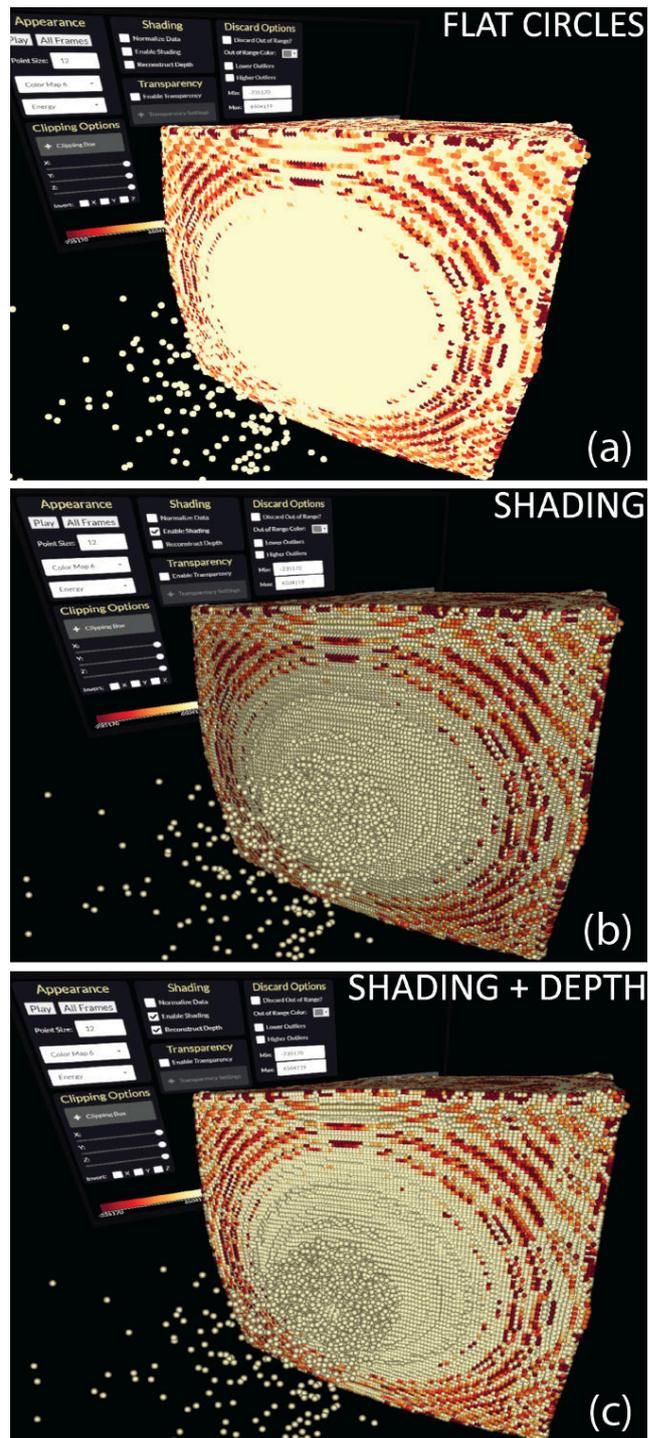
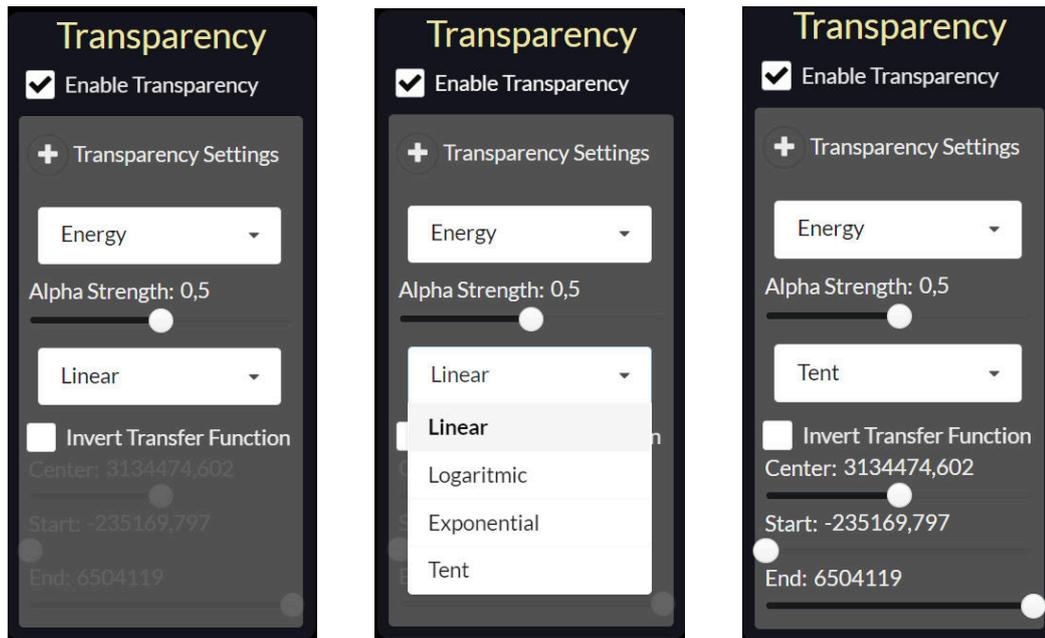


Figure 6.2: Rendering the points as flat circles (a), with enabled shading/impostors (b) and with shading and depth reconstruction (c)



(a) Available options in the transparency panel (b) The mapping functions for transparency (c) Sliders to modify the tent function

Figure 6.3: The images show the available settings for achieving transparency

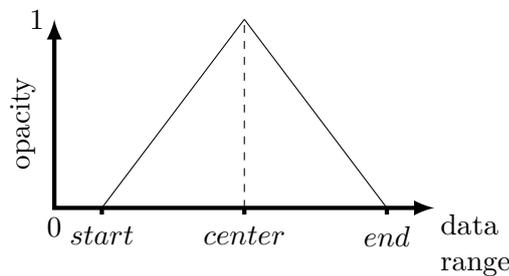


Figure 6.4: Adapted tent function

increasing opacity, like following an increasing linear function. In contrast, points with values between the center and the end value will have a decreasing opacity, respectively, like in a decreasing linear function. The user can modify these three values by using sliders and implicitly following certain rules. The start value can never be greater than the center, the end value can never be less than the center value, and the center value can always accept a value between the start and the end value. Our tent function provides much flexibility and can have various shapes, thus producing different mappings and transparency results for the different attributes. Figure 6.4 shows how our tent function is structured.

With our technique to render semi- or fully transparent points, we can emphasize extreme values and diminish irrelevant ones and observe inner structures while still perceiving the general shape of the point cloud. Using the transparency options, we can produce different rendering results and appearance of the point cloud. Examples of such different results can be seen in Figure 6.5.

Additionally, the shading and transparency options could be combined to receive even more expressive and sophisticated visual results. Highly transparent points will not be visible in the point cloud, while highly opaque points will be visible and will have a 3D shape so that the structure can easily be perceived. Figure 6.6 shows an example for combining both shading and transparency techniques.

6.3 Occlusion Handling for Billboards

We implemented a simple algorithm to reduce billboards' occlusions and hide unnecessary information. The algorithm goes through every probe and checks its position, radius, and depth compared to all other probes in screen space coordinates. If a controller currently intersects an observed probe, all other conditions are disregarded, and its respective billboard is visible. The same applies if a probe is not intersecting any other probe. However, there is a certain tolerance for the intersection condition. We do not want to hide one of the billboards of two probes, when they intersect "slightly", which means we check "how much" they intersect. We quantify that by measuring the distance between the centers of probes. If this distance is larger than the sum of the radiuses of two probes, then this means that they do not even intersect. If the distance is less than the sum of the radius of two probes, it means that they intersect. However, we want to hide one of the billboards not simply if they intersect but if almost half of one of them is occluded by the other billboard. For this reason, we introduce the factor 0.65 so that the allowed minimum distance between the centers of the two probes is defined as $0.65 * (p1_{radius} + p2_{radius})$, where $p1$ and $p2$ are two random probes from all placed probes in the scene, and 0.65 is used as a threshold for the maximum allowed partial occlusion. The number 0.65 was found empirically. This means that if the distance between the centers of two probes is less than their computed minimum distance, this is considered an occlusion. In that case, the algorithm compares the depth of the two probes to test which probe is closer to the screen. The billboard of the probe closer to the screen/headset is displayed, and the other billboard is hidden. We require our own algorithm for that because we want to avoid visual clutter, and, therefore, we are not rendering all those billboards that are only partially visible. This algorithm is called the *Billboard Occlusion Algorithm* (Algorithm 6.1). Figure 6.7 illustrates how different histograms are shown/hidden according to the view position and direction so that occluded billboards are just not rendered.

In addition to the *Billboard Occlusion Algorithm*, there is another algorithm responsible for the occlusion of a billboard and the point cloud. If a probe is placed fully or partially inside the point cloud, its billboard is occluded by some of the points, depending on

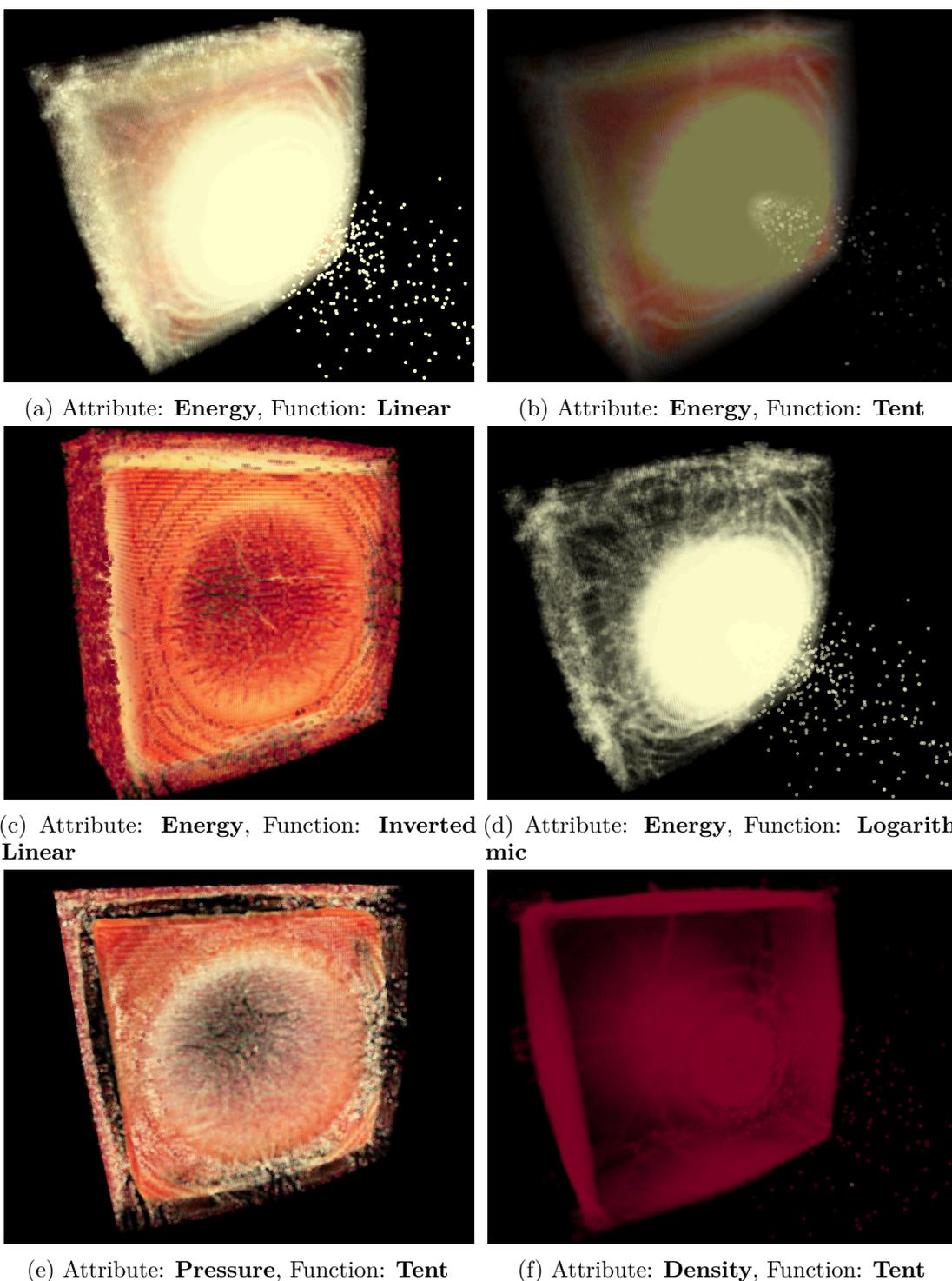


Figure 6.5: Different transparency settings applied to different attributes. Different characteristics and regions can be highlighted or discarded

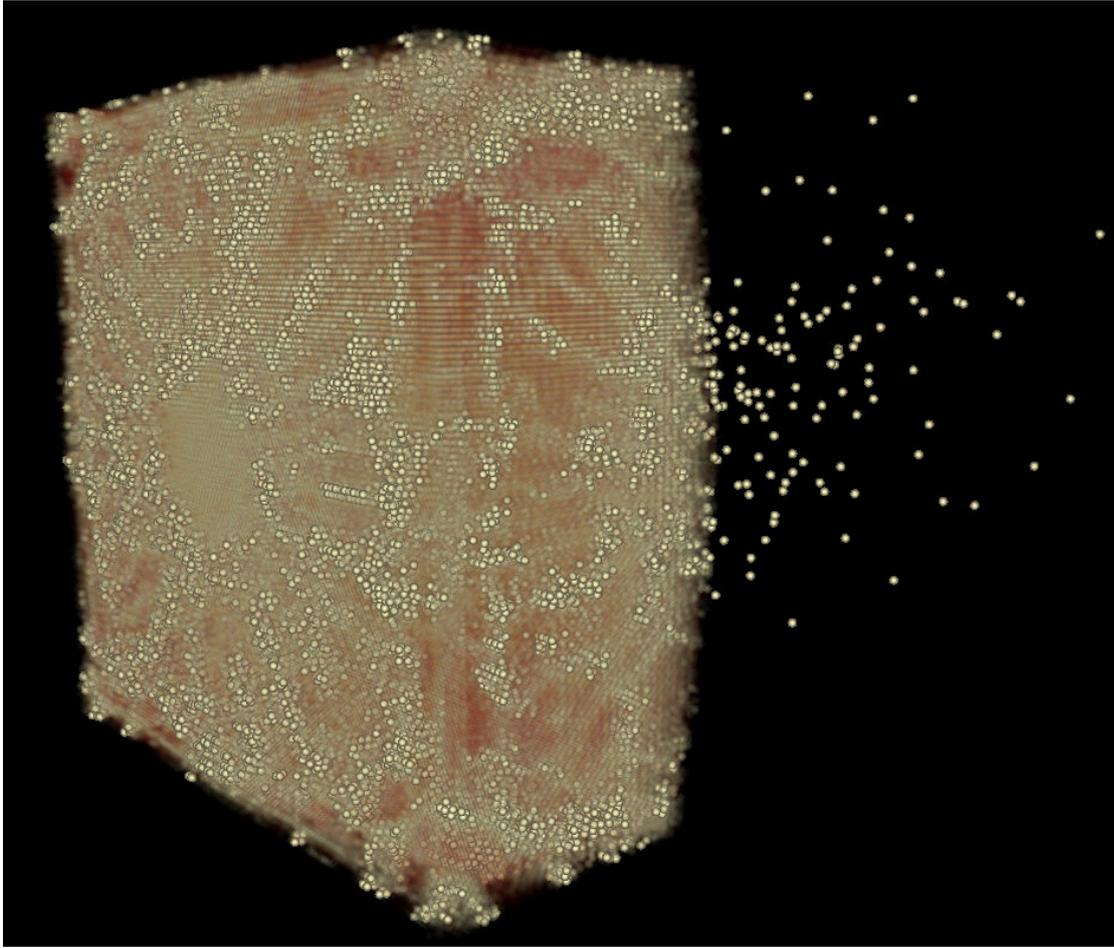


Figure 6.6: Combining shading and transparency options to remove irrelevant points and highlight the structure of the visible points.

the angle from which it is observed. A straightforward solution to this problem is to rotate the scene so that the billboards are visible since they always point to the headset (Figure 6.8c). However, this is not entirely intuitive and might still cause occlusions since the whole scene-setting is changed through such a rotation. Therefore, we introduce an algorithm that checks how far a probe is placed within the point cloud. If the center is inside the point cloud, the algorithm computes an offset with the up-vector so that the billboard is placed slightly above the point cloud (Figure 6.8b). This offset happens only for the billboard of a currently intersected probe. If there are other objects at the new computed location for the billboard, it might still be occluded, which is a limitation of the algorithm since it does not consider other objects. This limitation could be part of future work and improvements. However, this solution was sufficient for the current thesis and the domain experts' requirements. With the help of both algorithms, the users can observe non-occluded statistical information for the placed probes and, on-demand,

Algorithm 6.1: Billboard Occlusion Algorithm

Input: All probes P , model M
Output: Updated all probes P_{new} , where p_{bvis} is the billboard visibility of a single probe

```

1 if  $n(P) \geq 2$  and not  $M_{grabbing}$  then
2   for each probe  $p \in P$  do
3     if  $p_{intersected}$  then
4        $p_{bvis} \leftarrow true$ 
5     else
6       for each probe  $p^* \in P \setminus \{p\}$  do
7          $distance \leftarrow d(p_{posScreen}, p_{posScreen}^*);$ 
8          $allowedMinDistance \leftarrow 0.65 \times (p_{radiusScreen} + p_{radiusScreen}^*);$ 
9         if  $distance < allowedMinDistance$  then
10          if  $p_{posZ} < p_{posZ}^*$  then
11             $p_{bvis} \leftarrow true$ 
12          else
13             $p_{bvis} \leftarrow not\ p_{bvis}$ 
14          end
15        else
16           $p_{bvis} \leftarrow true$ 
17        end
18      end
19    end
20     $P_{new} \leftarrow P_{new}$  with modified  $p$ 
21  end
22  return  $P_{new}$ 
23 else
24   return  $P$ 
25 end

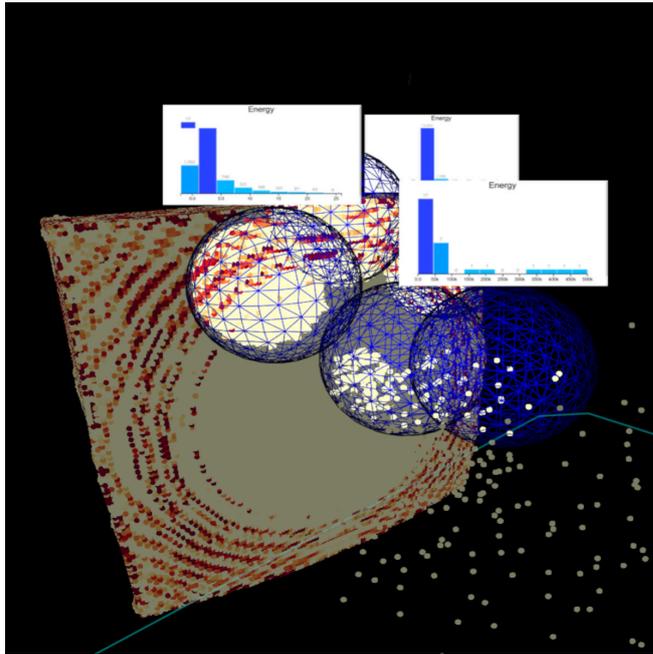
```

show the non-visible billboards by intersecting with a probe of interest.

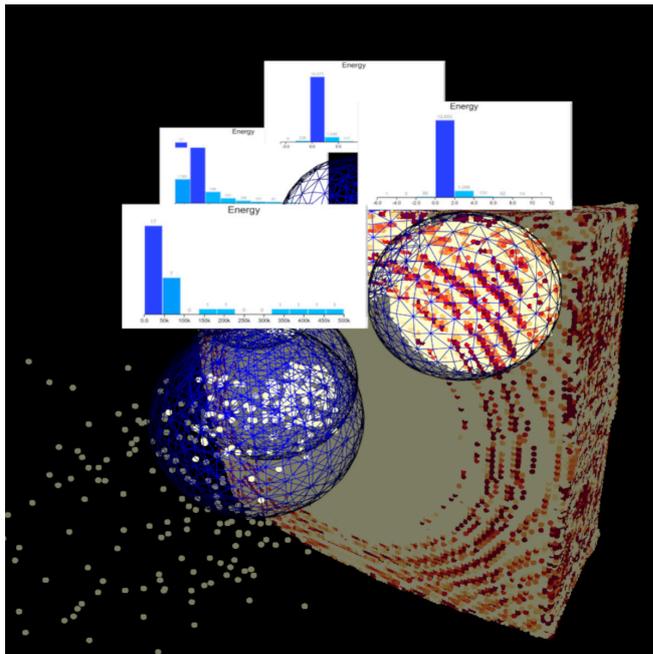
6.4 Performance Overview and Motion Sickness

We implemented and tested our system on a desktop PC with Windows 10 Pro 64-bit, with version 19044.1466 as an operating system. The PC has an AMD Ryzen Threadripper 1900X 8-core processor with a clock speed of 3.80 GHz. The RAM has a storage size of 32 GB. For the graphical processing, the computer is equipped with an NVIDIA GeForce RTX 2080 Ti GPU.

Along with the hardware and software parameters, an efficient implementation is essential

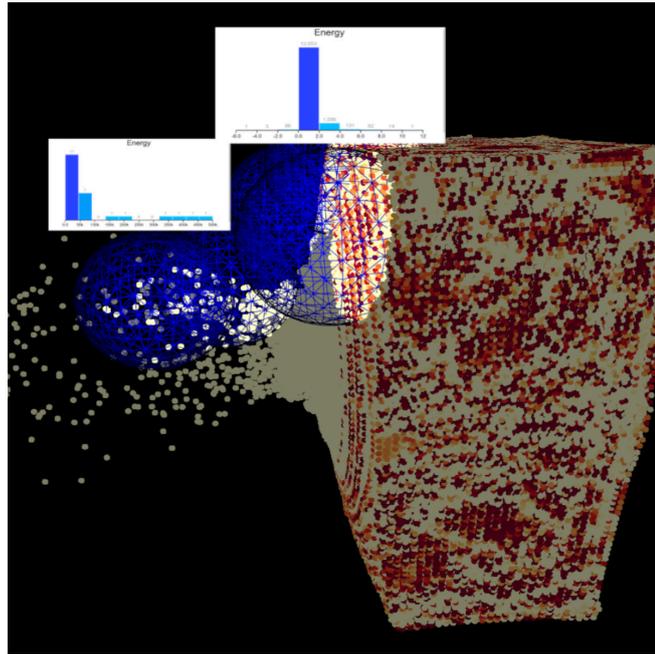


(a) Three out of the five billboards are visible for the given view direction



(b) Four out of the five billboards are visible for the given view direction

Figure 6.7: Billboard Occlusion Algorithm used to hide billboards that are occluded by other billboards from the camera view

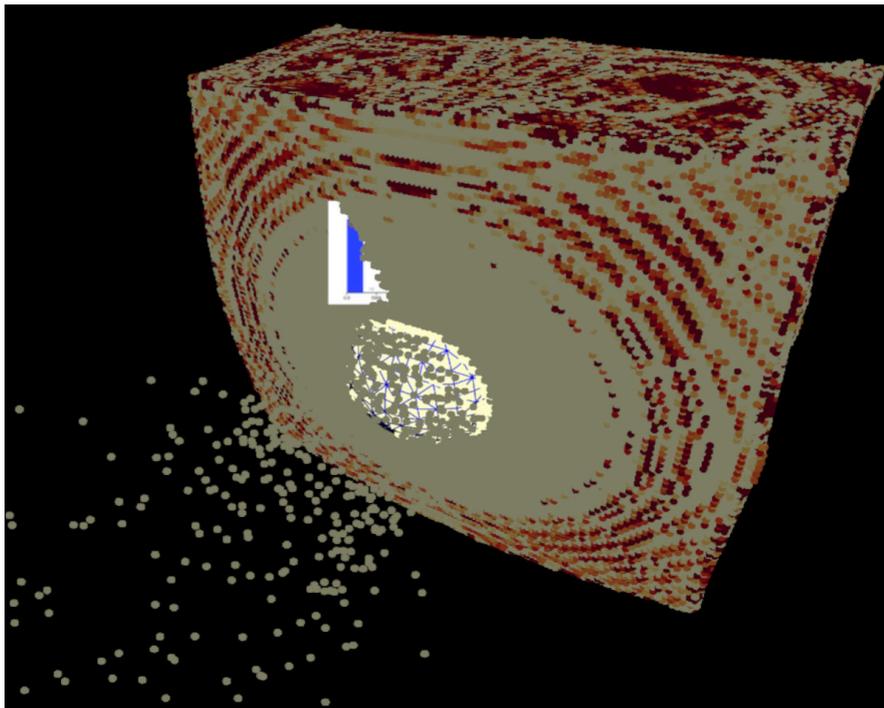


(c) Two out of the five billboards are visible for the given view direction

Figure 6.7: Billboard Occlusion Algorithm used to hide billboards that are occluded by other billboards from the camera view

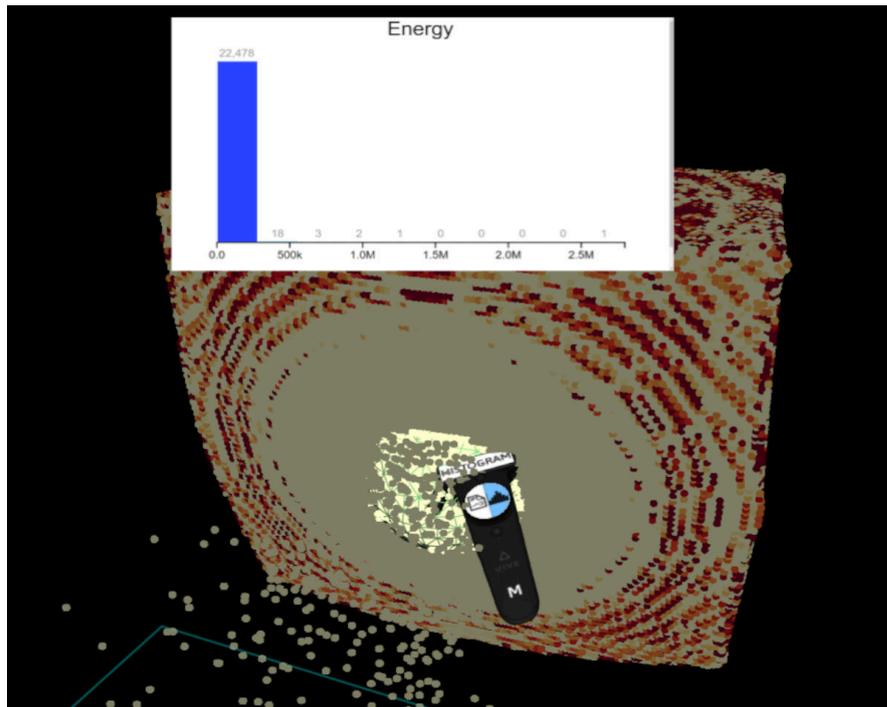
for achieving high performance and a realistic and smooth experience in Virtual Reality. For this reason, we tried to optimize our implementation by providing efficient data structures and algorithms. For instance, we use an octree structure for the traversal of points to identify which of them lie inside a probe needed for the histogram and statistical computations. The points are instantly highlighted visually since the rendering and visual properties are computed on the GPU. For a small number of points (≤ 50000), the histogram is computed within 200 ms. The histogram computation speed decreases for a larger probe and with an increasing number of points. In the worst case of including all points, a computation time of up to 4 sec results. The *Impact simulation data* consists of 400 frames. In our VR system we import only 10 of these frames, since with a larger number of frames, the performance decreases significantly, which hinders a smooth user experience. For this purpose, we take every 40th frame to represent the simulation, which is sufficient for the domain experts since the changes between two successive frames are negligibly small, and they want to gain an overview of the data properties. All controller menus and options are stored as textures on the GPU, so once the user interacts with the menus, the changes are displayed instantly, with no delay. Additionally, all histograms above the probes and placed planes holding box plots are also stored as textures. As a result, they cannot be modified or used for brushing, which would have decreased the performance due to the additional required computations. With the help of all these optimizations, we provide a highly responsive system delivering a real-time experience.

Ensuring good performance and providing a highly responsive system helps also in reducing the probability of motion sickness occurrence since it has been shown in Chapter 2 that performance has a significant impact in this respect. Our design choices are also fulfilling some of the main criteria for decreasing motion sickness emergence. Our system does not have multiple objects constantly moving or changing their spatial positions, which would significantly impact motion sickness. Furthermore, the scene is simple, and the user can naturally move through the space by walking, just like in the real world. Also, most of the interactions are well-established metaphors adapted from the standard 3D interaction techniques by LaViola et al. [LKM⁺17]. In addition, we implemented haptic feedback to notify the user that a particular interaction has been performed and to enhance the feeling of presence. For instance, if the user switches between tools and menu types on the controller or intersects with objects in the scene, the controllers will vibrate slightly to indicate the state changes.

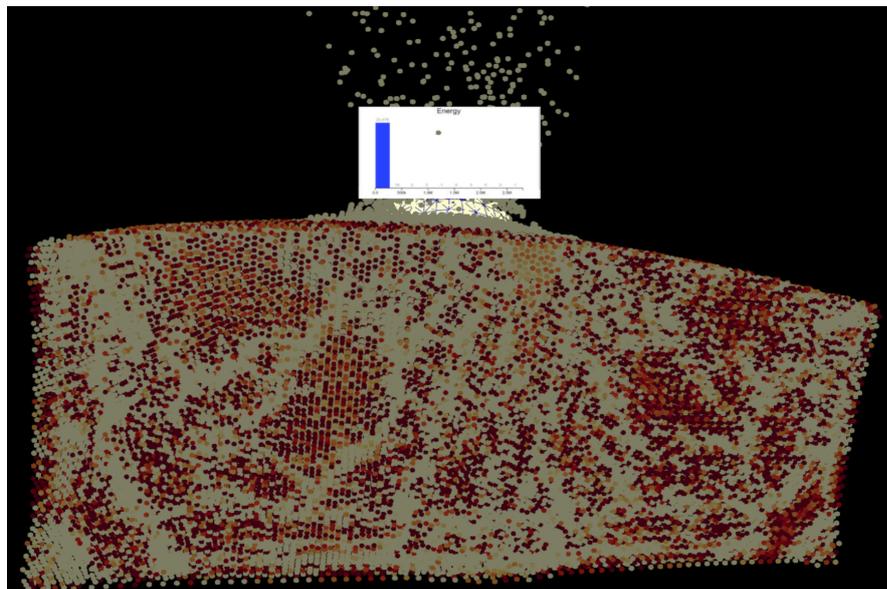


(a) The probe is placed almost fully inside the point cloud and the billboard is therefore obstructed by points

Figure 6.8: Billboard Offset Algorithm used to offset billboards in case the billboard is occluded by the points in the point cloud



(b) By intersecting the probe with the main controller the billboard is offset in the direction of the camera up-vector



(c) Rotating the point cloud can also help in solving the occlusion problem, but it might not work in every case

Figure 6.8: Billboard Offset Algorithm used to offset billboards in case the billboard is occluded by the points in the point cloud

Implementation

The thesis was carried out at the VRVis Research center and heavily utilized its infrastructure and its collaboration with astronomers Thomas Maindl and Christoph Schäfer. VRVis provided the necessary hardware and software for developing the current work. For the hardware and software details refer to the first paragraph of Section 6.4. For the implementation, we used Aardvark, an open-source platform for visual computing, real-time graphics, and visualizations, developed by the VRVis research center. To reduce unnecessary calculations, Aardvark updates the rendering only when the relevant inputs change [HSMT15]. The platform is based on a functional-first paradigm and provides a functional API for F#. For shader programming, Aardvark uses FShade [VRV14], so that shaders are written in F#-syntax, which is then converted to GLSL-syntax and compiled. For implementing VR applications, Aardvark sends the rendering results to the HMD via OpenVR [Ste15]. OpenVR is an API and runtime system used to access VR hardware from multiple vendors without requiring applications to know the hardware they are targeting explicitly. OpenVR is implemented by SteamVR supporting the Oculus Rift and HTC Vive.

Our project is based on the execution model of every Aardvark.Media App [Ort17], which follows the pattern of an unidirectional data flow (ELM architecture). This type of architecture can be seen in Figure 7.1. The **Model** component holds the data. The **View** defines how this data is visualized. The user can interact with the components the view produces, which in turn generates **Actions**. The **Update** is responsible for determining how these actions would modify the data and thus generate a new model.

In our implementation, we have two applications running simultaneously and exchanging data between each other. One of the applications is responsible for the 2D UI running on the 2D desktop, while the other one is the actual VR application. Changes applied on each one of them are instantly applied and visible in the other application. Both applications are running in parallel, which allowed us to load the 2D UI as an interactive texture on the TV screen in the VR environment. In order to integrate the 2D UI in VR,

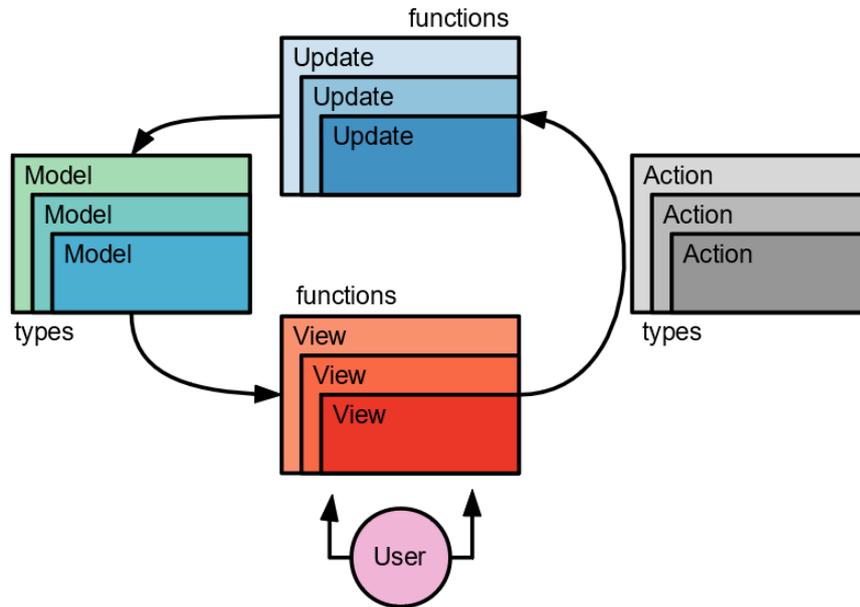
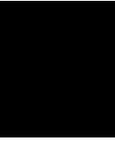


Figure 7.1: Aardvark Media App Composition [Ort17]

we use a specific module from the Aardvark platform, supporting a method to render an HTML file to a texture. We call this a render-to-texture method. We use it to first render UI elements like box plots implemented as HTML/D3 visualizations, but also to expose all 2D user interface elements mapped onto a virtual TV screen in VR.

The render-to-texture method is not used only for displaying the 2D UI on the TV screen, but also for all graphical plots in the VR application. The histograms and box plots implementation is written in JavaScript with the D3.js library. D3 allows the production of dynamic interactive data visualizations in web browsers. In order to display a histogram for each of the placed probes and to display the box plot planes, we use the render-to-texture method. The histogram for the last-placed probe in the scene is rendered on a separate HTML page. This separate page runs in the background and is necessary to save the last histogram as a texture. This texture is then displayed on a plane above the respective probe in our update function. The same principle applies to the box plot visualization. In this way, we only update the texture of the histogram or box plot that is currently modified.



Evaluation and Discussion

In this chapter we describe the evaluation process and discuss the results of the evaluation. In our evaluation, we assess the usability, effectiveness, and utility of the resulting VR system. We perform a user study by incorporating both quantitative and qualitative usability testing to address three concrete questions:

1. Is a Virtual Reality environment suitable for time-dependent multivariate data in general?
2. Is the resulting system intuitive and usable?
3. Can the system support and help domain experts in solving their tasks?

The following sections give a detailed description of our evaluation procedure, the used methodologies and questionnaires, the results of the evaluation and a short discussion about the resulting conclusions.

8.1 Target Group and User Study Design

Since our system was specifically built for scientists in astronomy, the target group are domain experts. However, we could not find enough participants for a solid user study. Therefore, we opted for a twofold evaluation - in an expert interview with the main cooperation partner Thomas Maindl, we evaluated the system from an expert point of view and after that, in a user study, we evaluated the usability.

In the expert interview, we try to answer the question whether the system fulfils the domain expert's expectations and can provide him with the necessary instruments for exploring the *Impact simulation data* in VR. We also use this as an opportunity to receive useful feedback for future expansions and improvements. In the second part of

the evaluation process, we perform the actual user study with several participants and different backgrounds. In the second part we mostly concentrate on usability, interactivity, and general satisfaction of users with the provided experience.

8.2 Evaluation with Domain Experts

In the first part of our evaluation process, our primary goal was to involve a domain expert to test our VR system. It was essential to analyze the effectiveness, usefulness, and delivered value for the domain experts. Therefore, we are thankful that Thomas Maindl took part in this evaluation step and provided helpful feedback for various aspects of the resulting system. This part of the evaluation process is particularly relevant for addressing the first and third questions we set initially.

The procedure consisted of a short introduction to the interactions and components of our VR application. As part of that, the domain expert was given a 10-minutes video showing the system's capabilities. After that, we performed a live demonstration of some of the key features and again explained how to accomplish them. As a next step, Thomas Maindl had to try out our VR system himself. He first was given time to test various options to get used to the techniques, and then he could use the system to explore the data and make insightful observations. We gathered notes during his exploration session and summarized some of his comments. Finally, Thomas had to fill out a questionnaire with several open-ended questions.

The domain expert was, in general, highly impressed by our system and believes that it is a beneficial way to explore data. In the beginning, the system complexity and interactions were a bit demanding and challenging for him, but then he said that he found the exploration process intuitive at some point. He commented that with our system, he could explore data much faster than he usually does because, in our VR application, the tester can instantly see changes and switch back and forth to different frames. Using our system, we observed that Thomas could directly read out various properties and detect patterns and correlations. He also commented on specific characteristics of certain attributes of the data with the help of the box plots and histograms. Furthermore, the domain expert liked the 3D point cloud and all the interactions that allowed him to reposition and rescale it to observe different parts of the data. He is satisfied with the quality of the point cloud, considering that with the color maps, he can quickly gain an overview of the attribute distributions. The two rendering modes were valuable to him, since they provide him with additional information regarding the data structure and distribution. Therefore, he said that all three rendering modes are helpful, and he would use all of them depending on his exploration goal and task.

During testing, the domain expert suggested inverting the axis-aligned clipping planes to obtain more flexibility. Additionally, he suggested enabling multiple controller clipping planes to reduce data to a desired subset and shape. Thomas also reflected on the functionality to export data. He finds it helpful to export data from specific points/regions in text format and also export plots to store new findings and, for example, use them

in future publications. Thomas believes that exploring such type of data in our VR system is better than on a desktop because the user has more space. Even if a person has multiple screens, at some point, they are cluttered, and in VR, the users can basically place everything anywhere in the virtual space and simply move their head around. Last but not least, the domain expert did not report any motion-sickness symptoms or perceptual issues.

In the following, one can see Thomas' answers to our questions:

1. Q: What was particularly useful/helpful in the system and why?
A: It is particularly useful: provides quick overview of the overall physical system, able to move “into” the data and look around, easy definition of areas of interest and plotting statistics/physical properties of SPH particles in these areas. Why: for efficient analysis of the SPH scenario it is important to identify (1) which simulation frames and (2) which areas within those frames are relevant. The VR system helps with both in a quick and intuitive way.
2. Q: Do you want to improve some of the existing features? How and why?
A: Being able to invert the axis-aligned clipping planes, so that they are cutting the point cloud from the opposite side. Also the controller clipping plane should move with the point cloud when translating and scaling.
3. Q: Which of the features of the system do you find most useful and why?
A: Moving around the SPH point cloud; Moving “into” the point cloud; being able to define areas of interest and plotting physical properties (in principle all columns in the data files: density, pressure, alpha-jutzi, internal energy, etc.)
4. Q: Are the interactions intuitive? If not, which of them would you like to improve and how?
A: Given the complexity of operations I think the interactions are intuitive. Of course, there's a learning curve and that's ok.
5. Q: Is the system in general easy to use or is it unnecessarily complex? Why and how would you improve it?
A: I think it is as complex as it needs to be. I would only improve minor characteristics (see Question 2).
6. Q: What advantages/disadvantages do you observe in the provided system?
A: Advantages: The system provides a quick insight into the simulation. The interactions and visualizations are well-designed. The system is very responsive, allowing to quickly create different scenarios and this way explore the simulation. Disadvantages: the only one that comes to mind is an intrinsic problem of current VR which is the necessity of VR goggles (with all their peculiarities: weight, bulkiness, cable, availability, missing standards).

7. Q: Did you have any perception/motion sickness issues and if yes where/to what extent?

A: No, none at all.

8. Q: In what way could a system like that help you explore the data?

A: See answers to other questions.

9. Q: Would you prefer to use such a system in VR instead of traditional desktop and why?

A: VR is a great tool to quickly get an idea about what is happening in the simulation. This includes identifying areas of interest like e.g., the crater rim and the material below the actual crater. For larger collisions it's the global outcome (merging/hit-and-run/erosion) and identifying larger and smaller fragments. For detailed analyses and reproducible plots, a traditional desktop might be better suited because there we need exact borders of our samples and exact ranges for the plotted parameters.

10. Q: Are there any features that are missing in the system and you would like to have? Which and why?

A: I think there is no "reset" and/or "reset view" functionality? For me this is important because I got lost a few times during the demonstration session.

11. Q: What is your general opinion/impression about the system?

A: I am very impressed. Such a tool for interactive, real-time data exploration was just what we needed from the beginning of our collision simulation efforts (real-time in the sense of feeding in a set of simulation output frames and navigating within one frame and also between frames in real-time, not in the sense of simulating collisions in real time). Especially in the early "exploration" phases of a collision simulation-based project it is important to get an overview of the big picture (What happens at all? Are there fragments? If yes which type/consistency/physical properties? How do physical properties such as pressure, alpha jutzi, damage, density, temperature/inner energy/... behave inside the material? Etc. etc.). Before we did this with (a) more or less static scripts that produce figures or predefined videos or (b) interactive exploration in ParaView. ParaView works nicely but it's always a 2D projection that lacks the true 3D experience.

In summary, the feedback from Thomas Maindl gave us a lot of information on what is essential and valuable in our system. We generally received positive comments, with minor remarks and improvement suggestions. His feedback suggests that our system provides options and techniques for exploration that were not possible with the systems and techniques he currently uses to analyze such data. Therefore, he finds our VR system useful and beneficial, especially for the early "exploration" phases that require a quick global overview and insight into the data. We integrated most of the improvement recommendations into our system before the user study with participants, which will be the topic in the next section.

8.3 User Study

In total, 10 testers participated in our user study. Among the participants there were computer scientists, students, and researchers from VRVis. The user study has been carried out for a period of around three weeks. The participants could book a slot for a testing session in advance. We planned 90 minutes for each slot since the system is complex and requires a solid introduction to all features and interaction options. None of the participants has seen or tried the VR system in advance.

One testing session includes several steps. Since some of the participants were not familiar or had minimal experience with Virtual Reality technology, we started with an introductory testing session to familiarize them with our system. We then introduced the participants to VR and how to use the controllers to interact with the immersive environment. Afterwards, we introduced them to our work, the motivation and background information, and the goal and purpose of this thesis. They also watched a 10-minute video of all possible system interactions to get first insights into how the system looks and what possibilities it provides. Before the next part, the participants were asked to fill out a user information form in Google Forms [Goo08] to provide more background information. As part of the next step of the testing session, each participant had the chance to try out our system and get to know the possible interactions and options. We called this step "Guided System Testing", as the participants were given six different simple tasks to interact with the environment. These tasks were addressing the technical and interactive aspects of the application. They were not intended to involve exploration and gaining insight into the data, but to familiarize participants with the system capabilities first. The users were encouraged to ask for help and support whenever they did not know how to perform an interaction or solve a task. Once this step was finished, the testers took a 5-minute break to rest from the virtual experience and reduce the possibility of getting motion sickness.

After the break, the participants' actual testing session began. This step is called "Explorative Tasks", where each task involves exploring different aspects of the point cloud visualization. Most of the tasks can be accomplished in several ways. We are not providing any information on how to complete them. Our intent was to leave room for creativity and flexibility and observe how each participant behaves and what tasks and features of the system might appear challenging. We continue with a free exploration task, where each participant could interact with the system freely by using all available tools. As final part of the user study, the users had to evaluate our system. For the evaluation we used the System Usability Scale (SUS) [Bro95] and a questionnaire in Google Forms, covering different aspects of our VR application and each users' individual experience with the system. The whole evaluation procedure can be found online [Vas21]. The following sections describe all steps of the testing session in detail and provide a comprehensive summary of the process and results.

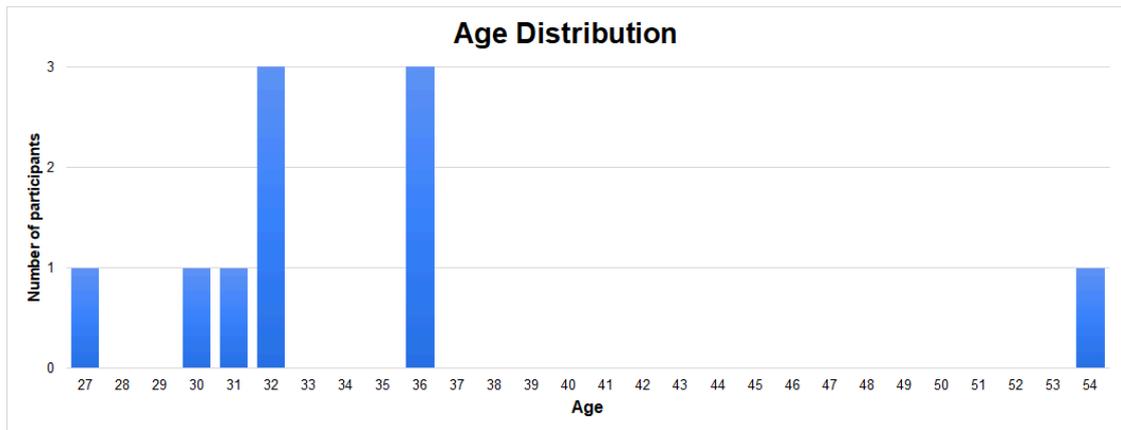


Figure 8.1: Age distribution chart of the user study participants.

8.3.1 User Information and Introduction

In the first step of our user study evaluation, the participants were asked to provide some information and answer a few questions about. The information includes:

1. Name - to identify participants through all questionnaires
2. E-mail - to provide the anonymous results of the questionnaires
3. Gender and Age - to see if a participant is representative for certain groups
4. Frequency of computer, AR, and VR usage - the users were asked to provide information about how often they use computers and AR/VR to determine whether previous experience plays a role in getting used to the system interactions faster than non-experienced users
5. Purpose of VR usage - to identify what kind of experience (if any) users have with VR technology

Out of the 10 participants of the user study, 6 were male and 4 female. Most of the testers were between 30 and 36 years old, one was 27, and one was 54, as shown in Figure 8.1. Figure 8.2 illustrates how often do participants use computers, virtual reality, and augmented reality. As the results show, all of them are using computers regularly. When it comes to virtual reality - three of the participants have used it only once, four of them tried it a few times, one is using it sometimes, and only two people are using this technology regularly. The results for augmented reality are slightly different since three people are using it regularly, two are using it sometimes, four people used it a few times, and only one person has tried it only once. To sum up, all participants have tried virtual reality technology at least once, but only a few of them have plenty of experience. Figure 8.3 displays the purposes for which the participants used VR and AR technology. Almost all of them used it to play a game, while only one of them used it to watch a video. Seven of them implemented a system in VR or AR, and only two participants did research or participated in a user study related to this kind of technology.

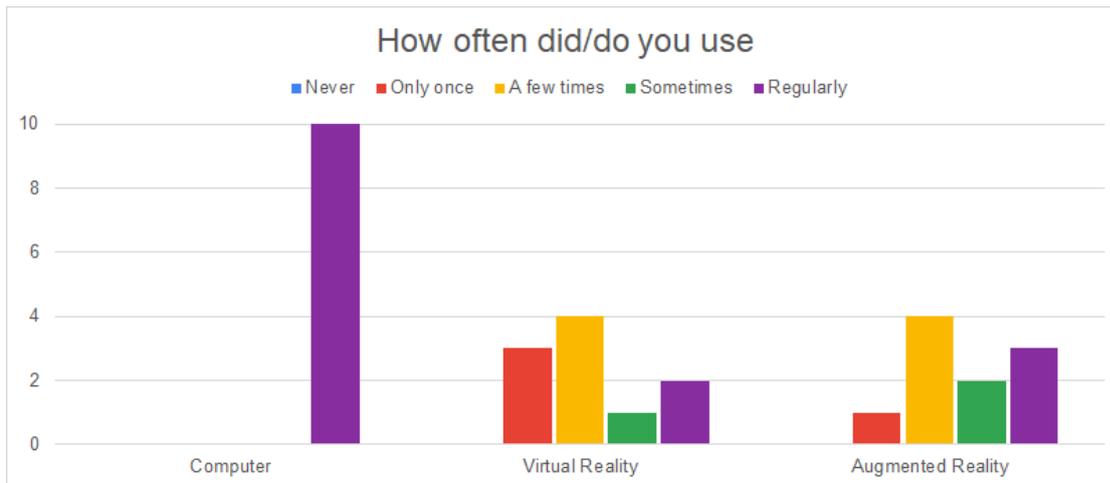


Figure 8.2: Chart showing how often the participants use computers, virtual reality, and augmented reality.

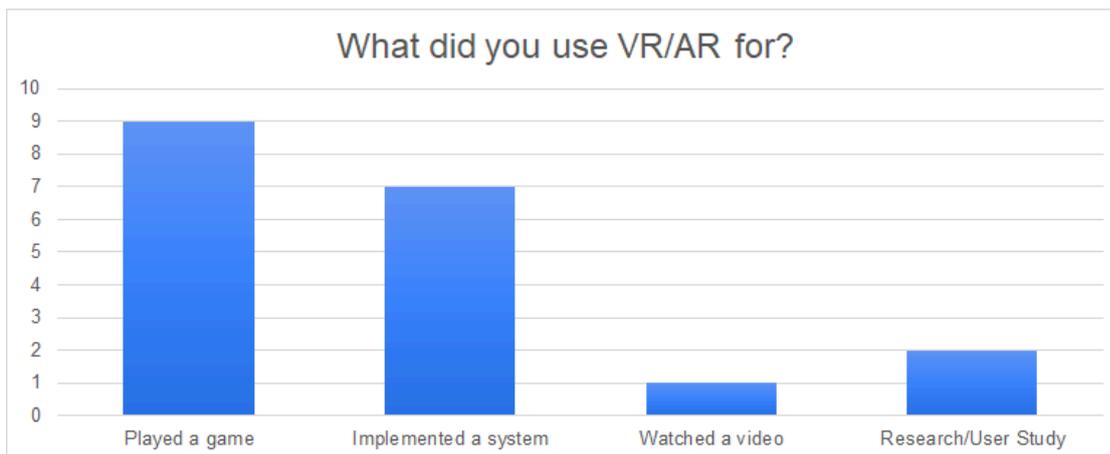


Figure 8.3: This chart shows for what purposes did the participants use virtual reality or augmented reality.

Providing background information is beneficial for identifying the significance of experience for fulfilling the tasks. Accordingly, we would like to figure out whether people with little to no experience can get used to the system quickly or would instead need additional training and learning in the VR technology to get going with our VR application.

As part of this step, the users were also introduced to our work, including background information about the project, the motivation, and the desired goal. The participants also watched a 10-minutes video showing our VR setting, the possible interactions, what they are used for, and how to trigger them. This was intended to provide a general impression of our VR environment, its components and available exploration tools.

8.3.2 Guided System Testing

In the second step of our evaluation procedure, the participants had the chance to try our VR system for the first time and get familiar with the interactions and available features. The users were encouraged to ask for help and guidance throughout the testing. The guided system testing is structured in the form of tasks. The participants had to perform six different tasks, covering diverse aspects of the application. This way, we ensure that all necessary information required for interacting with the system is presented to the users. Moreover, this step serves as a preparation for the next step. The users performed the following tasks:

- **Task 1** - Create four probes with different sizes at random places in the environment.
- **Task 2** - Change the attribute to pressure or density of two of the probes and switch to statistics for one other probe.
- **Task 3** - Activate the clipping plane and cut out part of the points. Grab the whole visualization, scale it down/up and place it somewhere else. Delete the clipping plane.
- **Task 4** - Activate ray. Move the TV to a desired position so you can see both the screen and the visualization. Try to change some settings of the 3D visualization using the 2D floating menu (color map, point size, shading, transparency etc.).
- **Task 5** - Create three probes at the crater. Turn off billboards. Compare the probes with the probe *analyze regions tool* for the density attribute. Place the resulting box plot plane. Copy it and change the attribute.
- **Task 6** - Turn off visual links for one of the created box plot planes. Select one probe and use the animation options to compare three frames of your choice for the pressure attribute.

The goal of the first task is to get used to the *probe tool* since it is the central exploration concept in our system. For this purpose, the participants had to use the controller menu to select the tool before creating probes. They also had to use the secondary controller to rescale the probes. In general, this task was quite simple to perform for most of the participants, especially those who use VR technology regularly. For the testers, who had tried VR only once or a few times, it took a bit longer to get used to the interactions required to open the menu, select the tool, and create a probe. Task 2 is intended to show the participants how to change the attribute of a probe and switch between histogram and statistics. Changing between the histogram and statistics was easy for most participants, while changing the attribute was more challenging, since the user needs to intersect a probe with the main controller while selecting a new attribute on the secondary controller. In Task 3, the participants had to select the *clipping tool* and use it to cut out part of the points. Additionally, they had to use the long-range pointing

technique to grab the whole point cloud, scale it up or down, and place it at another location. Clipping was easy to perform for the participants. The grabbing was also easy. However some users found it a bit difficult to rescale the scene while holding the grip buttons, but they quickly got used to it. Task 4 focused on the *ray tool* and the interactions on the TV screen. After activating the *ray tool*, the users had to modify some of the available options on the 2D UI to change different features of the visualization. There were no difficulties in performing this task. The final two tasks involved testing the *analyze probe tool*. As part of Task 5, the participants had to use the *analyze regions tool* to compare three probes according to the attribute density. They also had to place the plane and use the copy function to duplicate the information of the box plot plane, for which they required instructions and help. Finally, in Task 6, they had to use the *analyze time tool* to compare frames at a particular region. They had to first try out the video player settings on the controller to modify the animation flow and then select different frames for comparison. As part of this task, they also had to turn off visual links. This step of the user study focused on the technical execution and interaction techniques to achieve a particular setting or result rather than interpret the visualizations.

8.3.3 Explorative Tasks

In the "Explorative Tasks" step of the user study, the participants had to explore the data. The goal was to get insights into the structure of the data, observe and explore different parts and regions of the point cloud visualization, and discover trends and patterns. For this reason, there are no specific guidelines on the required interactions to accomplish a certain task. Instead the tester has the freedom to decide how to perform a task. The tasks were:

- **Task 1** - Tell me the distribution and differences of the energy attribute at the impact crater and at the right top edge of the visualization? What are the min/max and average values?
- **Task 2** - Create three probes and analyze them by telling me what differences do you observe in terms of density? Where is the lowest/highest value? What is the average at each probe?
- **Task 3** - How does the energy attribute vary compared to the density for the same three probes?
- **Task 4** - How does the pressure change at the start, the middle, and end of the simulation at the impact crater?
- **Task 5** - Display visually somehow only the highest energy values (Hint: use transparency, histogram or data range).
- **Task 6** - Explore the visualization however you like and use whatever options you want (including 2D screen and controllers). What are your observations? What did you find out?

Observations

While the users performed the tasks, we observed their exploration process and marked down the execution time, complexity, and most importantly, how they performed each task.

Task 1: Most of the participants managed to complete the task in less than 5 minutes. All of them instantly recognized that they needed to use the *probe tool* to compare the two regions. Only a few needed guidance on the interactions required to open the menu and select the *probe tool*. Four of the participants used the histogram to answer the questions from the first task. Four of them used the statistics to read out differences and min/max values and only two participants started with the *analyze regions tool* and box plots. The participants who had more experience with VR technology took less time to perform the task than those who used it only a few times. Notably, while the participants generally preferred the histogram, two users found extracting the necessary information from the histogram to be complex and, therefore, preferred the statistics. Also, this is related to the precise question - when asking for a min, max, average, the statistics are clearly easier to interpret since it simply contains the answer.

Task 2: The execution of the second task took around 4 to 5 minutes for most of the participants. This task is similar to the first one, with the only difference that the users needed to compare three instead of two probes and had to work with the density attribute instead of energy. As a result of the task similarity, some participants again used histograms and statistics to perform the comparison. However, they quickly came to the conclusion that it was challenging to keep track of all the information and compare several probes with this approach. Consequently, they switched to the *analyze regions tool* and used box plots for the purpose. As for Task 1, two participants had problems with interpreting box plots. We also observed that the participants already in this task started getting used to the system interactions.

Task 3: Before solving Task 3, the participants were explicitly instructed to keep the settings and probes constructed in Task 2, since Task 3 builds on them. On average, the users took about 3 minutes to complete the task, as it required similar interactions to the previous task. All of the participants found the box plots the most suitable solution for comparing different attributes of the same probes. Some participants started selecting all three probes again in the same order and then chose the desired attribute. Others remembered that the system provides a faster approach for this specific case, the "Copy"-feature, so they used it to copy the whole plane and only changed the attribute.

Task 4: To complete Task 4, all participants recognized that they had to use the *analyze time tool* to compare frames and observe simulation progress in time. They took, on average, around 3 to 4 minutes to complete the task. As they were already familiar with the box plots, this significantly reduced interpretation time. However, some of them struggled with selecting the desired frames. For this tool, the user first needs to select a probe, and afterwards, the current frame is directly displayed as a box plot on the plane. All other frames need to be added using the screenshot button, represented by a check

mark icon, which can be found at the video player options on the controller menu. This type of interaction was challenging for some participants. The challenge came from the fact that the video player menu is working only by touching. Therefore, some participants accidentally touched parts they did not intend to or forgot about touching and clicked on the touchpad, automatically opening the main menu. As a result, they had to open the video player menu again.

Task 5: Task 5 was intended to encourage participants to use the 2D UI on the TV screen. There are several ways on how to display the highest energy values, so the users were given the freedom on how to perform the task and also interpret what is considered as a high energy value. Four of the participants used transparency settings, as they find this functionality appealing and also suitable for the given task. Three of the participants used the data range options to filter out data and highlight only the points with higher energy values. Two of the users applied the higher outliers checkbox to display only the points considered to be outliers according to the energy attribute. Finally, only one participant used the histogram to brush data and filter only the highest energy values. The results show that there is not a single preferable approach to solve Task 5 and all users have their own preferences. In general, the participants spent more time on this task compared to the previous one. On average it took around 5 to 6 minutes, as the users wanted to try out different options.

Task 6: The final task was a free exploration task, where the users were given about 5 to 10 minutes to explore the visualization freely and try available options on their own. As Task 5 was about using the 2D UI, most participants continued trying the different options on the TV screen. Some participants mainly focused on shading and transparency, while others tried out the clipping options. Some of the testers combined the settings of the 2D UI with the controllers' tools to explore the visualization. In general, most participants were highly involved in the exploration process and wanted to spend more time testing our VR system.

Most of the participants quickly got used to the interactions and system capabilities. Some of them needed several hints and guidance on how to perform a specific action for some tasks. We did not expect unfamiliarity with interpreting box plots and histograms in our target audience. However, we conjecture that our tasks were relatively simple from a statistical point of view, biasing our evaluation towards the statistics instead of the box plots and histograms. Also, interestingly for some tasks with multiple solution strategies, we did not see clear preferences, which indicates that there is no single best tool for the tasks we asked for.

8.3.4 Usability Evaluation

After performing the explorative tasks, the participants evaluated our system according to different aspects by filling out several questionnaires. In this step, we mainly focused on our system's performance, usability, and general effectiveness. First, the users had to evaluate the usability of our system. For this purpose, we used the System Usability

Scale (SUS) ([Bro95]), which is a popular way to measure the perceived usability of a particular system. The SUS is a 5-point Likert scale consisting of 10 questions that users need to answer. The participants have to rank each question from 1 to 5, depending on how much they agree with the statement. 5 means that the user strongly agrees, while 1 would mean that the user strongly disagrees with the given statement. The 10 standard questions of the SUS are:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very awkward to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

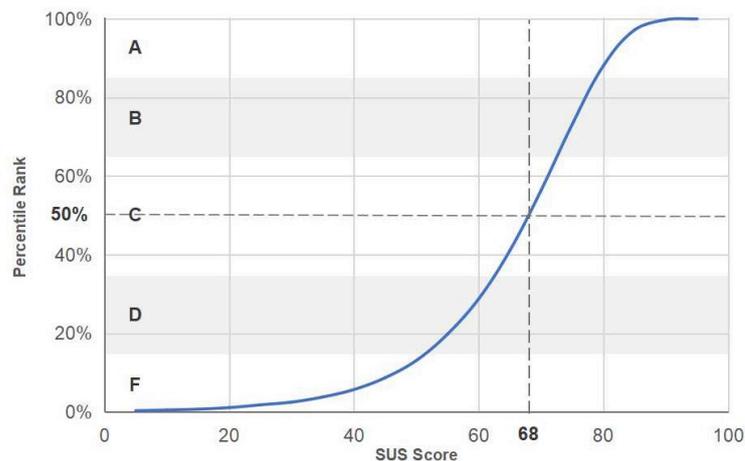
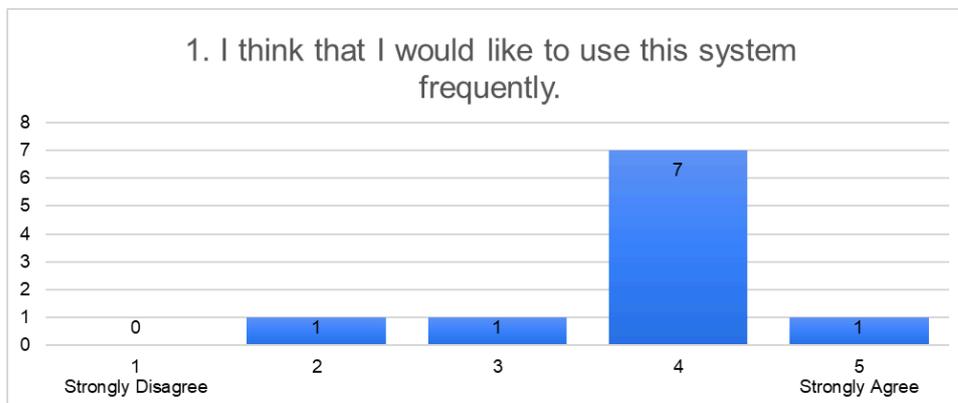


Figure 8.4: SUS Score curve showing the corresponding percentile ranks and grades [Sau18]

After the users have ranked each of the questions based on their level of agreement, the SUS score for each participant can be calculated. The calculation happens by first adding the points for each odd-numbered question and subtracting 5 from the score. Then the points of all even-numbered questions should also be summed up, and the resulting value should be subtracted from 25. The two resulting values should be added up. Finally, the total score should be multiplied by 2.5, and the result gives the SUS score. These calculations are necessary to get a score out of 100. It is crucial to outline that this

score is not a percentage. However, a SUS score can be converted into a percentile rank. Figure 8.4 shows the percentile ranks for the different SUS scores. The average score (at the 50th percentile) is 68, meaning that a SUS score above 68 is above average and below 68 is below average. A SUS score can also be interpreted as a grade, similar to the grades in schools. Grades range from A to F, where A indicates a superior performance, F illustrates a failing performance, and C indicates an "average" performance. Each SUS score corresponds to one of the given grades, as it can be seen in Figure 8.4.

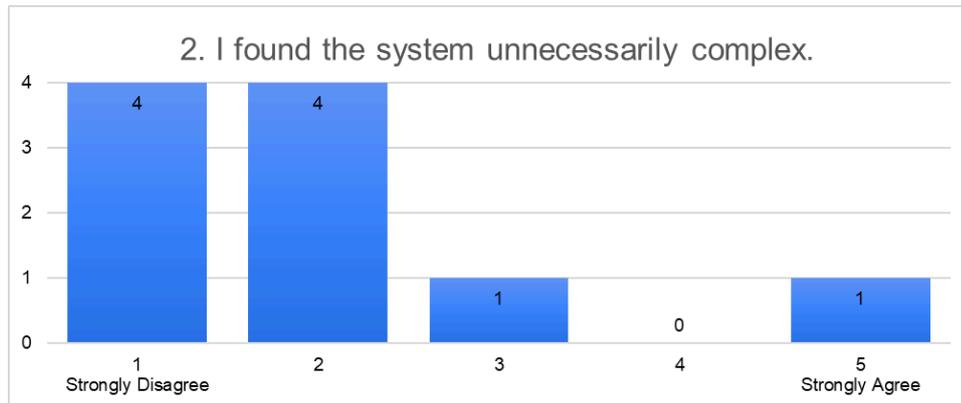
Figure 8.5 represents a bar chart of the participants' ratings for each question of the SUS. In general, most of the participants would use the system rather frequently, do not find it unnecessarily complex, and believe that they would not need the support of a technical person to be able to use the system. They also think that they will learn the system quickly and not need to learn many things before using it. However, some individual answers show that a few participants had a more challenging experience with the system's usability. These participants did not find the system easy to use, so they believe they would need more time to learn how to use it, and, therefore, did not feel confident. Nevertheless, this is an expected result because the system is complex and provides a wide variety of exploration options. Furthermore, interactions are challenging for people with less experience in VR. Also, astronomers and novice users have different approaches to exploration and interactions. Clearly, the domain is specific, and domain experts have a clear advantage given the system's complexity. Nonetheless, according to most of the responses, the system does not seem excessively complicated to interact with and does not require a highly time-consuming learning process to use it.



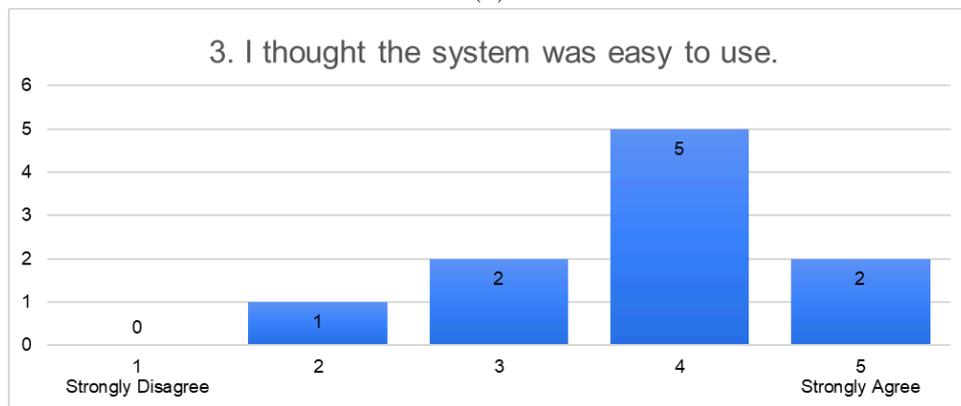
(a)

Figure 8.5: Graphical representations of the responses concerning the SUS for each individual question.

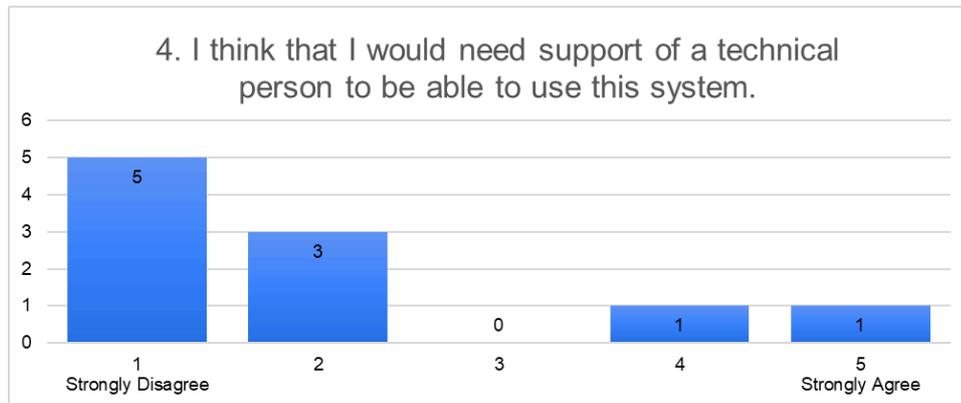
8. EVALUATION AND DISCUSSION



(b)

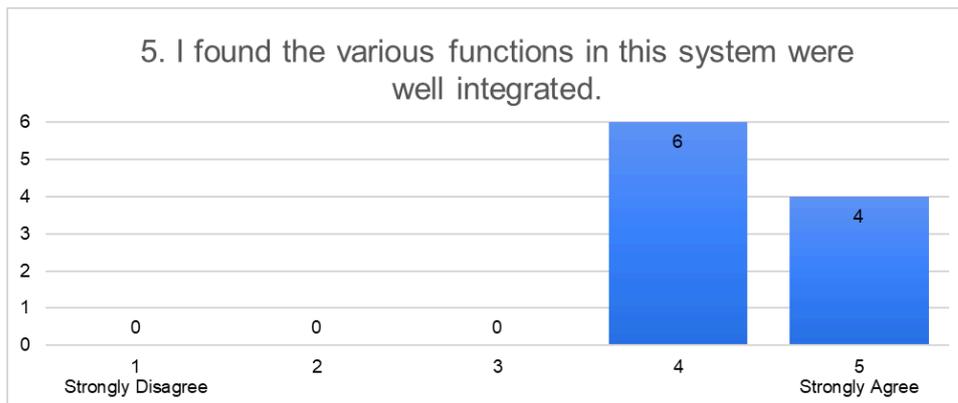


(c)

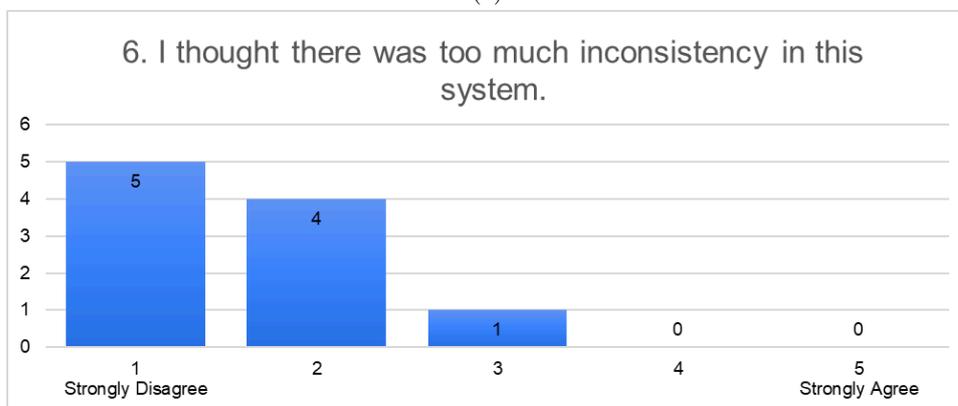


(d)

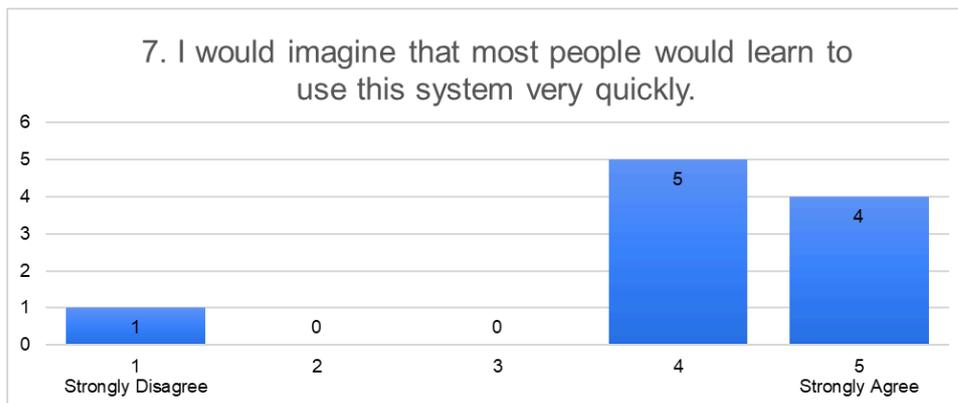
Figure 8.5: Graphical representations of the responses concerning the SUS for each individual question.



(e)

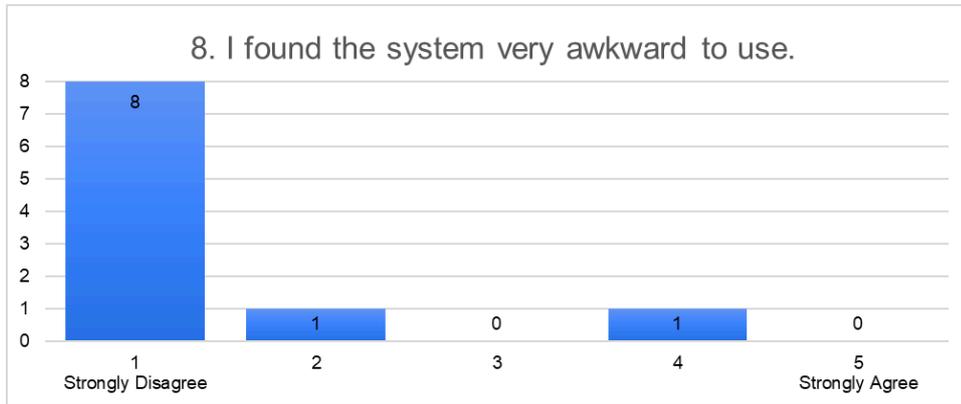


(f)

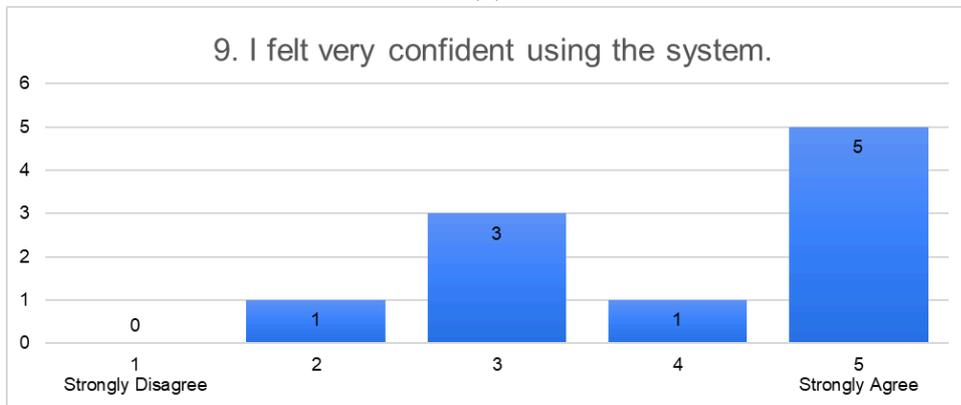


(g)

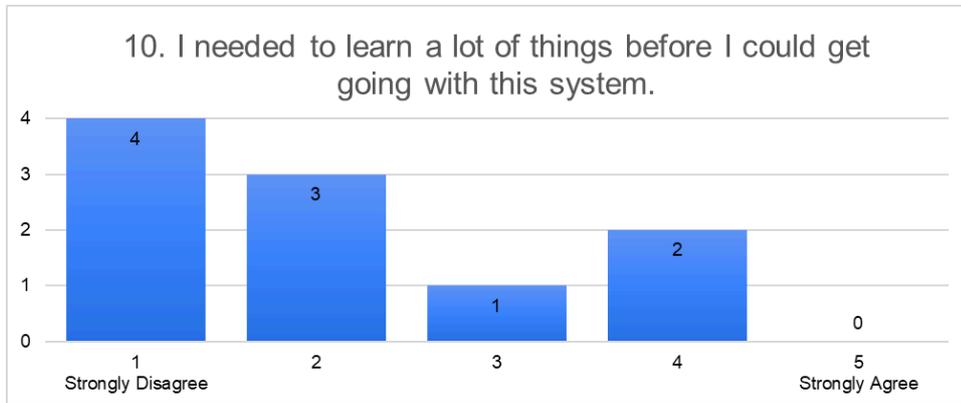
Figure 8.5: Graphical representations of the responses concerning the SUS for each individual question.



(h)



(i)



(j)

Figure 8.5: Graphical representations of the responses concerning the SUS for each individual question.

Participant	SUS Score
Participant 1	82.5
Participant 2	82.5
Participant 3	77.5
Participant 4	85.0
Participant 5	32.5
Participant 6	72.5
Participant 7	62.5
Participant 8	95.0
Participant 9	92.5
Participant 10	92.5
Average	77.5

Table 8.1: The table displays the individual SUS score of each participant as well as the average SUS score of all 10 participants.

Table 8.1 illustrates the SUS score of each participant, as well as the average SUS score of all participants. Most of the participants have SUS scores above 68. Three participants have a score above 90, three have a score between 80 and 90, and two have a score between 70 and 80. Only two participants have a score below 68 - Participant 5 with 32.5 and Participant 7 with 62.5. The average SUS score of all participants is 77.5, with a percentile of around 80 % and a grade B. This result suggests that the perceived usability is above average. Consequently, the results indicate that our system provides a relatively high-quality user experience in terms of effectiveness, efficiency, and overall user satisfaction. Nevertheless, it also suggests room for improvement and that some interactions or system features could be modified or simplified.

8.3.5 General System Evaluation

In the last step of our evaluation procedure, the users had to fill out a more extensive questionnaire. It is split into three different categories, according to which aspect of the system is evaluated. The three categories are - interactions, visualizations, and general questions. Each category consists of around 8 to 10 questions on a 7-Likert scale from 0 to 6, where 0 means "Completely False" and 6 means "Completely True". Depending on the question, the users can rate if and to what extent they find the statement/property true. In each category after the Likert-scale questions, the participants can express their personal opinion about what they found particularly useful in the system and what could be improved, and how.

Interactions Questions

In the first category, we evaluated different aspects of the system interactions:

1. How natural did your interactions with the environment seem?

2. How responsive was the environment to actions you initiated (or performed)?
3. How natural was your navigation/movement through virtual space?
4. Were you able to anticipate what would happen next in response to the actions that you performed?
5. How well could you move or manipulate objects in the virtual environment?
6. How intuitive could you navigate through the controllers menu?
7. How natural could you create probes?
8. How intuitive could you compare regions/time frames?
9. How useful did you find the 2D floating menu and the options it provides?
10. How helpful was haptic feedback for completing interactions?
11. Give your personal opinion about the interactions in the system. What do you find particularly good and why? What would you improve and how?

Figure 8.6 shows a table of the resulting answers to each of the questions. For this category, the participants had to rate 10 questions on a 7-point Likert scale. The results of the questions are represented in the columns, where each black circle corresponds to a participant's response. The rows display the possible ratings of the Likert scale, varying from 0 to 6. Ratings from 4 to 6 portray positive feedback to various aspects of the system. The higher the rating, the more the participant "agrees" with a given question. A response of 3 is neutral, meaning that the user neither agrees nor disagrees with a given question. Ratings from 0 to 2 illustrate negative feedback of different aspects of our system. The lower the rating, the more the user "disagrees" with the given statement. The strength of the responses is also indicated with a color grading. Green color illustrates positive feedback (the higher the saturation, the higher the positivity level), grey indicates a neutral response, and red portrays negative feedback (the higher the saturation, the higher the negativity level).

As shown in the table, the majority of responses represent positive feedback for the various aspects of the system. Few responses are neutral, and several fall into the negative feedback. However, none of the participants responded with a score less than 1. One participant does not find the interactions with the environment completely natural. One participant was somewhat unable to anticipate what would happen next in response to his/her actions. These replies are not surprising since all participants tried the system for the first time and had a limited time for testing it. The responses for two of the questions are distributed over all possible scales, except for 0. One of them is Question 6 regarding the intuitiveness of the navigation through the controllers' menus. The reason most probably lies in the fact that the controllers' menus combine several different types of interaction. For switching between the tools, the user only needs to touch the trackpad, and for selecting the desired tool, the user needs to click on the trackpad button. The video player for controlling the animation, however, works only by touching. Therefore, this might be confusing for the users when they first try the system. The second question with replies over several scales is Question 8, which is about the intuitiveness of comparing regions/frames. The *analyze probe tool* is a rather complex tool, including a variety of options, which is most probably the reason why some participants found the comparison

(6) Completely True	●●●●	●●●●●●	●●●●●	●●	●●●●●	●	●●●●●●●	●●	●●●●●●	●●●
(5) Mostly True	●●●	●●	●●●	●●●	●●●●●	●●●	●●●	●●●	●●	●
(4) Slightly True	●●	●●	●●	●●●●		●●		●●	●	●●●●
(3) Neither True nor False						●●		●		●●
(2) Slightly False	●			●		●		●		
(1) Mostly False						●		●	●	
(0) Completely False										
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10

Figure 8.6: Table illustrating the participants' responses to the interactions questionnaire

challenging. Regarding Question 9, one user does not find the 2D floating menu and the options it provides useful. Other than that, the remaining questions received primarily positive feedback.

Question 11 was open-ended. The participants had to express their personal opinion about the interactions by describing what they found good and what could, in their opinion, be improved. Several participants replied that they found the interactions are well structured and easy to learn and perform. Most of the users also believe that they do not need a long period of training time to get used to the system. Additionally, some users find that the interactions are well integrated despite the system's complexity and the wide variety of options. Some participants also gave suggestions about possible improvements. Participant 2 suggested adding more labels to the buttons to show the available functionality of the controllers. Participant 4 proposed displaying the graphs (histogram or statistics) also while moving and scaling the probes, involving real-time data updates. According to Participant 5, it is possible to make errors, as some of the controllers' buttons were used for several different purposes. Therefore, the user suggested adding a menu with tabs attached to one of the controllers for navigation and tools selections. Participant 6 recommended adding a symbol or 3D object to the *analyze probe tool*. In this way, the tester can quickly identify that this tool is also an option, since it is the only tool that does not have a 3D object representation. Finally, Participant 9 suggested improving consistency regarding touch vs. press controls on the trackpad.

Visualizations Questions

In the second section of our questionnaire, the participants had to evaluate the visualizations and visual aspects of the system according to the following questions:

1. How much did the visual aspects of the environment involve you?
2. How completely were you able to actively survey or search the environment using vision?
3. How well could you examine objects from multiple viewpoints?
4. How well could you perceive the structure of the data?
5. How well could you compare the attributes of different regions/frames using box plots?

(6) Completely True	••	•••	••••••	•••••	•••	••••••	•••••	•••••••
(5) Mostly True	•••••••	•••••	••	•	••••	•••	••	
(4) Slightly True		•••	••	••••	•	•	••	•••
(3) Neither True nor False					•		•	
(2) Slightly False					•			
(1) Mostly False								
(0) Completely False								
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8

Figure 8.7: Table illustrating the participants' responses to the visualizations questionnaire

6. How helpful do you find the visual links for indicating the respective probe for the box plot?
7. How compelling did you find the animation/simulation?
8. How useful did you find the histogram/statistics above the probes?
9. Which visualization technique did you personally find most expressive and useful?
10. Which of the rendering types did you personally find most expressive and informative?
11. Give your personal opinion about the visualizations in the system. What do you find particularly good and why? What would you improve and how?

Figure 8.7 represents the scores for the visualization questions. The participants had to answer 8 questions on a Likert scale. As can be recognized from the table, most of the responses are in the positive feedback range. There are only two neutral responses and one response is somewhat negative. The negative response was given to Question 5, asking how well participants could compare different regions/frames attributes using a box plot. This question is similar to Question 5 in the interactions category. However, in this case, we are asking about the visual aspect of the comparison. As some people faced challenges in interpreting the box plots, it is probably the reason for having a response in the negative segment of the table. Other than that, the results show that the participants were satisfied with the visual aspects of the system and did not have significant difficulties examining the simulation.

Questions 9 and 10 in this category were single-choice questions, where the users had to choose an answer according to their preferences among several options. In Question 9 the participants had to select the visualization technique that they found the most expressive and useful one. They could choose from histogram, statistics, box plot, and a parallel coordinates plot. Figure 8.8 shows the results of the participants' responses. None of the users selected the parallel coordinates plot as most expressive and useful. The parallel coordinates plot was introduced for the basis implementation on a 2D screen and have a limited functionality only (see Chapter 4). In our VR system we do not extend that functionality and do not provide parallel coordinates plots in 3D space like histograms, so that the only interaction with this type of plot is through the 2D UI. This is probably the main reason why most users did not even interact with the parallel coordinates plot. One participant prefers the histogram visualization over the others.

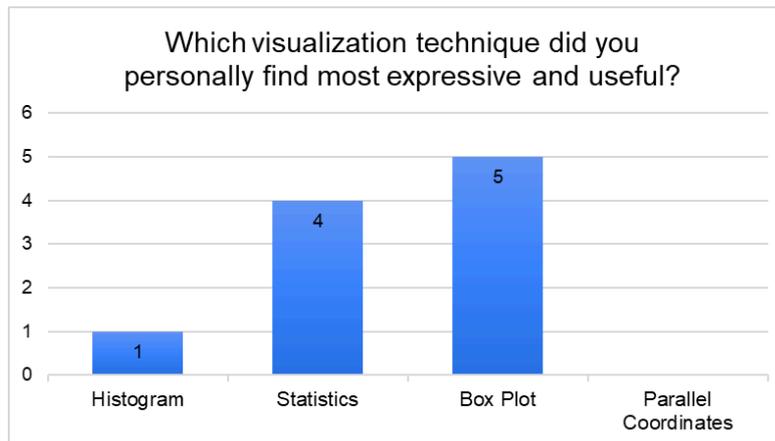


Figure 8.8: Bar chart representing the preference for most expressive and useful visualization technique for each participant.

Four participants personally find the statistics most helpful and expressive. Some of them communicated that they could not interpret the other visualization plots, which presumably explains their choice. Half of the participants preferred the box plots over the other three alternatives. Some of them argued that the box plots are, to a certain extent, a combination of the histogram and statistics and are more suitable for comparing datasets. Several participants were indecisive while answering this question. They said it is hard for them to choose one particular visualization technique since they are all suitable for different cases. Therefore, these users said that they find it good to have a variety of techniques and switch between them, according to the context and the task.

In Question 10 the participants had to select the rendering technique they found most expressive and informative. The results for this question are illustrated on a bar chart in Figure 8.9. 60 % of the users selected transparency. Some of them said that a tester can observe regions with high or low attribute values in transparency mode while still getting a global overview of the structure. Three of the participants selected the flat 2D circles because they said it was the most straightforward technique. However, they would prefer some of the other techniques depending on the use case. Only one participant chose the impostor rendering technique over the others, as he/she believes this is the best way to perceive both the global structure and minor structure roughness.

Finally, in Question 11 the participants had to express their personal opinion regarding the visualizations in the system - what they found particularly good and what could be improved. Some participants mentioned that they found the visualizations well-designed and useful. They helped them to understand how the simulation works. They also do not imagine that this would work so well in 2D desktop environments. Many users expressed positive impressions on alpha blending, which helped them inspect and quickly understand hidden features of the data. One participant found the *clipping tool* particularly helpful,

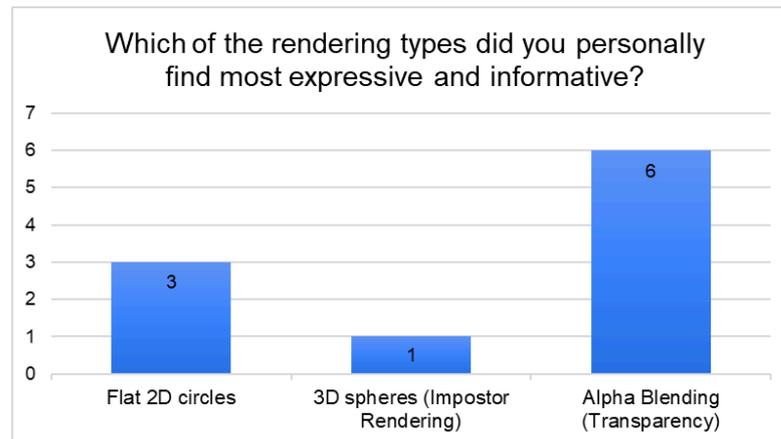


Figure 8.9: Bar chart representing the preference for most expressive and informative rendering technique for each participant.

as this allowed him to get rapid insights into the inner parts of the dataset. According to one participant having a more fine-grained selection of min, max, and alpha values in the transparency panel would improve the system. Another participant said that she imagines that this system would be an excellent tool for experts to explore the *Impact simulation* data. However, for an inexperienced user, there are too many features at once. Therefore, the participant suggested adding an expert and non-expert mode, which would help hide some features for people who are using the system for the first time. According to another participant, it would be beneficial to have also automatic defaults for the transparency mode since it was challenging for him to find suitable parameters for every attribute.

General Questions

In the final section of the questionnaire, the participants had to answer several general questions about the system regarding their experience. These questions include system performance, level of confidence, satisfaction, presence, motion-sickness occurrence, and others:

1. How quickly did you adjust to the virtual environment experience?
2. How confident did you feel while using the system?
3. How efficient did you find the performance of the system (Performance in terms of lagging, delay, jitter)?
4. How present did you feel in the virtual environment?
5. How much did the visual display quality interfere or distract you from performing assigned tasks or required activities?
6. How well could you concentrate on the assigned tasks or required activities rather than on the mechanisms used to perform those tasks or activities?
7. To what extent did you feel motion-sickness symptoms?

(6) Completely True	●●●●●●	●●●●●●	●●●●	●●●●●●●●	●●●	●●	●●●●●●	●●●●	●●●●
(5) Mostly True	●●		●	●●	●●●●	●●	●	●●●	●●
(4) Slightly True		●●●	●●●		●	●●	●●	●	●●●●
(3) Neither True nor False	●		●			●●		●	
(2) Slightly False		●	●		●	●●		●	
(1) Mostly False					●				
(0) Completely False									
	Q1	Q2	Q3	Q4	Q5*	Q6	Q7*	Q8	Q9

Figure 8.10: Table illustrating the participants' responses to the general questionnaire

8. Are you satisfied with the experience the system provides?
9. Were you able to gain insight and find out interesting information about the data while exploring?
10. What were you able to find out?
11. If you have any other remarks about the system, for example what did you mostly like or general improvement suggestions, you can share them here.

The table in Figure 8.10 shows the responses of the participants to 9 general questions on a 7-point Likert scale. For the sake of consistency, negatively formulated questions are converted to positively formulated questions so that they correctly fit into the table responses. These questions are marked with an asterisk sign. Similar to the other two question categories, most of the scores are situated in the positive range of the table. There are five neutral responses, and seven fall into the negative feedback region. The first negative response with a score of 2 is on Question 2. In Question 2 the participants were asked about how confident they felt while using the system. Since all of them are using the system for the first time, it is quite probable that some of them are not feeling too confident due to the large number of features. Question 3 is about the efficiency and performance of the system regarding lagging, delay, jitter. One participant gave a score of 2, and another one gave a score of 3. Although our system provides high performance, there are cases when lagging might occur. One such case is when the user increases the point size. Another case is when the HMD and the controllers are not within the tracking range of the base stations. Questions 5 and 6 received two negative responses each. As the participants are inexperienced testers of our system, it is challenging for them to concentrate solely on the assigned tasks without thinking about how to perform them. Fortunately, none of the participants experienced any severe motion-sickness symptoms. In general, most of the testers are satisfied with the experience of our VR application.

In Question 9 the users were asked if they could find out interesting information about the data, and in Question 10 the participants had to describe what they were able to find out. The findings of each participant are the following:

- **Participant 1** - *"The distribution of some attributes; differences and similarities of different regions; change pressure over time at the impact crater."*

- **Participant 2** - *"That there are differences in density and energy in the center and on the corners or the particles moving away from the crater."*
- **Participant 3** - *"How the energy of the impact spreads out over time and where the extreme values are positioned within the dataset."*
- **Participant 4** - *"The distribution of energy and pressure around and within the impact region and their change over time."*
- **Participant 5** - *"About the different aspects and properties of the simulation and how to get insight into the Impact simulation data with the help of the VR system in an interactive way."*
- **Participant 6** - *"That the density grows at the borders and on at certain point the asteroid surface breaks because the density is getting too high."*
- **Participant 7** - *"Using plots directly in VR does provide some interesting insights into different properties of a simulation."*
- **Participant 8** - *"How heterogeneous data of an impact develops and its correlations."*
- **Participant 9** - *"Cracks in the inner structure of the data; the shape of the impact site (it was a spherical shape with inner impact fragments); changes over time in density and energy of the material; comparison of the energetic properties near and far away from the impact site."*
- **Participant 10** - *"Differences between the attributes of the points inside the crater, at the corner of the visualization, and the furthest points of the explosion."*

In the final question, the participants were asked to express their general impression about the VR system and their experience and suggest what could be improved. Several users mentioned that they genuinely like the resulting system and the exploration capabilities it provides. One participant expressed that he/she finds it particularly beneficial to interact with the objects and the scene directly and is, in her/his opinion, a funnier and more insightful way of studying data. A comment made by another participant was that the combination of 2D UI screen and controller interactions is well-integrated since the controllers provide the central tools for 3D exploration, and the 2D screen expands the functionality further, which results in a sophisticated system for exploration. Another user found the 3D interactions and visualizations particularly effective and appropriate for simulation data like the *Impact simulation data*. As part of future development, this user also suggested replacing part of the 2D visualizations with 3D, such as 3D histograms. One participant communicated that the user interface should not occlude the data when transparency mode is turned on. According to another participant, a help menu might be useful to learn the system without any guidance. Furthermore, one tester suggested providing a reference to the captured frame on the histogram to avoid confusion.

8.4 Discussion

We split our evaluation procedure into two parts. In the first part, we evaluated our system with a domain expert, and in the second part, we conducted a user study with ten participants. The focus of the evaluation with the domain expert was to test the feasibility of our system and review the appropriateness of a VR technology for the given task. The feedback shows that our system is promising and equips the domain experts with the necessary tools to accomplish their exploration objectives. Thomas Maindl also commented on possible improvements and features that he finds valuable and that would enhance their experience and exploration procedure. Furthermore, according to the domain expert, VR technology is a reasonable choice for the given problem domain and data type. It helps users to quickly get an idea about the simulation by identifying regions of interest, examining various properties, comparing areas and time frames, and drawing conclusions about characteristics. Also, it allows a tester to be inside the data and directly interact with components and in this way is more involved in the exploration process.

On the other hand, the user study was intended to mainly investigate the usability, interactivity adaptation, and overall satisfaction of users, testing our system. The results of the SUS and our questionnaires suggest that our system is generally usable, relatively quick to learn, and provides an enjoyable way of exploring data. However, several participants experienced some challenges with the adaptation to our interaction techniques. As a result of that, our primary focus in improving the system will be on the interactions. With our whole evaluation procedure, we were able to address all three research questions we set in the beginning and provide a detailed understanding of the substantial aspects of our VR system. Besides, we discovered parts that should undergo some modifications and improvements and could be part of future developments.

Conclusion and Future Work

The following section gives a conclusion regarding the proposed VR system for visualizing and exploring the time-dependent multivariate *Impact simulation data*. The conclusion is followed by a discussion of improving existing limitations and new possibilities for future developments.

9.1 Conclusion

In this thesis, we proposed a novel Virtual Reality system for the visualization and interactive exploration of the time-dependent multivariate *Impact simulation data*. For visualization, the impact simulation can be exported as point data - one point cloud per time-step, which is the input for our application. The simulation consists of multiple frames and represents an impact with an asteroid. We visualize the data by a 3D point cloud located at the center of the virtual environment. The main objective of our system is to provide domain experts various VR tools to support *Impact simulation data* exploration and get insight into its characteristics. Our system enables interaction with the point cloud and provides diverse options for exploring its properties. The system requirements have been formulated together with domain experts, and the provided capabilities are adjusted explicitly according to their requests and needs.

Our system combines 3D controller tools and a 2D UI integrated into the environment to support exploration. The primary exploration tool is called the *probe*, which is a 3D geometric object used to analyze particular regions of the point cloud in more detail. For our application, we used spheres. For exploring regions, we provide histogram plots and a statistical summary for the points inside each probe. These statistical visualizations allow experts to read out trends and characteristics at a particular location. We also provide tools for comparing different regions or time changes in a particular area by introducing box plot planes. Our system supports several rendering modes, including impostor rendering of spheres and transparency rendering using alpha blending. These

modes are used to discover hidden properties in the point cloud structure and better understand attribute distributions. Moreover, our system provides a variety of additional settings to modify the appearance of the point cloud, filter data or cut it with the integrated clipping planes. To achieve a smooth and natural experience, we combine close-range and long-range object manipulation techniques chosen depending on the task. Last but not least, we provide a simple scene with natural navigation in the environment, resembling the normal physical walking of a person.

Furthermore, we evaluate our system with a user study, to test its feasibility for domain experts and discover potential challenges for participants. The evaluation is split into two parts. The first part involves an evaluation with a domain expert, and in the second part, we conduct an extensive user study with ten participants. The goals we set at the beginning of the thesis were to provide a novel system in Virtual Reality with various exploration tools to support domain experts in analyzing the *Impact simulation data* intuitively and interactively. Therefore, with our evaluation, we want to address three main aspects of our thesis:

1. Is a Virtual Reality environment suitable for time-dependent multivariate data in general?
2. Is the resulting system intuitive and usable?
3. Can the system support and help domain experts in solving their tasks?

The first part is intended to provide an answer mainly to the third question but also addresses the first question. In contrast, in the second part, we test the system usability and general satisfaction of participants, thus addressing the second question primarily. In the testing session with the domain expert we evaluated the features of the system and received feedback. The observations and results suggest that the domain expert is generally satisfied with the given capabilities of our system and sees the benefits of a VR environment for the exploration, as it provides a quick and practical way to get an overview of the simulation. The main advantage for him is that with VR, he can be inside the data and quickly analyze different areas by getting real-time results. With his traditional way of exploring the data using scripts, he has to wait for several hours until he gets the results. The user study gave us helpful feedback regarding usability, interactions, visualizations, and user satisfaction. The results indicate generally positive responses regarding different aspects of our system. Most users were able to quickly adapt to the system capabilities and provide various observations regarding the *Impact simulation data* and its properties. The feedback from the user study also included several critical comments and improvement suggestions that require further research to provide an even better user experience. The improvement recommendations and critical issues were mainly regarding the interactions, their complexity, and partly their inconsistency. The given feedback is highly beneficial for future development and system enhancements.

9.2 Future Work

As of November 2021, there is no comparable system in visualizing time-dependent data in a similar way as our VR system, so there is plenty of room for improvement. One possible improvement could be to adapt the given techniques and approach for other time-dependent and multivariate datasets with properties similar to the *Impact simulation data*. Another possibility is to overcome some of the existing limitations and provide better performance. In our current approach, we limit the frames in the animation to ten, because all frames need to be kept in memory. The system starts lagging with a larger number of frames. We could consider better data handling, memory allocations, and data structures to enhance performance when having more frames.

Furthermore, static plots could be converted into dynamic ones by introducing parallel techniques to update information for several probes simultaneously. Also, we could implement a more sophisticated algorithm for billboard displacement in case points from the point cloud occlude a billboard. The integration of a 2D UI in a 3D immersive system was surprisingly positively accepted and can, therefore, be another aspect to research further.

Another area for further research are the interaction techniques. Several users struggled in the beginning with the information overload regarding the possible interactions. For this reason, we could include labels for the various options on the menus that could be displayed on demand. Another option would be, as suggested by one of the user study participants, to provide a novice and expert mode of the system allowing users a faster adaptation without too much information overload.

Just as importantly, for future work, we can consider further domain experts' requests. Clearly, for making the work even more usable it should be possible to export plots and findings for integrating them in other tools or research. Moreover, we could optimize some of the existing options to incorporate only the needed tools for domain experts.

List of Figures

1.1	Interactive exploration of the <i>Impact simulation data</i> in VR	1
1.2	Single frame from the <i>Impact simulation data</i>	2
1.3	3D point cloud model of the Colosseum in Rome [SCS ⁺ 10]	3
1.4	Region comparison with two spherical probes for the attribute density	5
2.1	3I of Virtual Reality [BC03]	8
2.2	Components of a VR system [Zha17]	8
2.3	Types of Senses in VR [Dan19]	9
2.4	Popular VR Games [Ste18]	10
2.5	Examples for VR in the industry	10
2.6	Examples for VR used for education and medical training	11
2.7	Examples for VR used for rehabilitation and mental health therapy	12
2.8	VR for architecture, modelling, design, and construction planning [DODB20]	12
2.9	Examples for visualizations of 3D trajectories	16
2.10	Multivariate data visualization systems	17
2.11	Collaborative data visualization systems	18
2.12	Filtering time-dependent simulation data with a selection box in VR [HWK09]	19
2.13	Visualizations created with the VR-Viz framework [Sai18]	20
2.14	Different visualizations created with the DXR toolkit [SLC ⁺ 18]	21
2.15	IATK (Immersive analytics toolkit) [CCB ⁺ 19]	22
2.16	Visualizations created with VRIA [BJR21]	23
2.17	Comparison between the Immersive Analytics Toolkits [BJR21]	24
2.18	SSQ profiles for a VR device [KPCC18]	25
3.1	NASA DART and ESA Hera mission concepts [ESA19b]	36
4.1	Overview of the interface for the 2D desktop analysis of the <i>Impact simulation data</i> [VS20]	40
4.2	The <i>Impact simulation data</i> visualization with modified settings using the UI controls	43
4.3	Examples of using the histogram to filter data	44

4.4	In this example we use the parallel coordinates plot to filter out the points according to three dimensions. When discarding the out-of-range values the internal structure is exposed showing regions with moderate energy, low cubic-root, and low strain.	45
4.4	In this example we use the parallel coordinates plot to filter out the points according to three dimensions. When discarding the out-of-range values the internal structure is exposed showing regions with moderate energy, low cubic-root, and low strain.	46
5.1	The two main components of the VR system. In the right part of the image is the 3D point cloud visualization of the data. On the left is a 2D UI placed on a TV screen used to modify properties of the point cloud.	51
5.2	Example of using the <i>probe tool</i> for creating probes, displaying histograms and statistics, and changing individual attributes.	53
5.2	Example of using the <i>probe tool</i> for creating probes, displaying histograms and statistics, and changing individual attributes.	54
5.3	The histogram of the last-placed probe in the scene is also visible on the TV screen. The histogram above the probe is simply a texture, while the one on the TV screen is interactive and can be used for filtering data.	55
5.4	Example of using the <i>analyze regions tool</i> for comparing regions and their characteristics according to different attributes	58
5.4	Example of using the <i>analyze regions tool</i> for comparing regions and their characteristics according to different attributes	59
5.4	Example of using the <i>analyze regions tool</i> for comparing regions and their characteristics according to different attributes	60
5.4	Example of using the <i>analyze regions tool</i> for comparing regions and their characteristics according to different attributes	61
5.5	Example of turning visual links off	63
5.5	Example of turning visual links off	64
5.6	Example of using the <i>analyze time tool</i>	66
5.6	Example of using the <i>analyze time tool</i>	67
5.7	The <i>clipping tool</i> , used to cut through the 3D point cloud and observe the inner structure	68
5.8	The 2D UI that is extended with additional functionality for the Virtual Reality system (see basic version in Figure 4.1)	69
5.9	Using the TV screen 2D UI to modify various properties of the point cloud visualization.	70
5.9	Using the TV screen 2D UI to modify various properties of the point cloud visualization.	71
5.9	Using the TV screen 2D UI to modify various properties of the point cloud visualization.	72
5.10	HTC Vive hardware components [Inc15]	73
5.11	HTC Vive Controllers and Buttons [Var19]	74

5.12	Radial menu on the main controller to switch between tools	75
5.13	The radial controller menu on the main controller to switch between different tools	76
5.14	The radial attribute menu on the secondary controller to switch between different attributes	77
5.15	Rescaling a probe depending on the distance between main and secondary controller	78
5.15	Rescaling a probe depending on the distance between main and secondary controller	79
5.16	Intersection of a probe with a controller indicated by the sphere color . .	80
5.17	Ray color change according to the intersection with the TV screen	82
5.18	Grabbing and scaling up and down the point cloud visualization	83
5.19	The possible interactions with both controllers assigned to different buttons	86
5.19	The possible interactions with both controllers assigned to different buttons	87
6.1	Normal computation in 2D space	90
6.2	Rendering the points as flat circles (a), with enabled shading/impostors (b) and with shading and depth reconstruction (c)	93
6.3	The images show the available settings for achieving transparency	94
6.4	Adapted tent function	94
6.5	Different transparency settings applied to different attributes. Different characteristics and regions can be highlighted or discarded	96
6.6	Combining shading and transparency options to remove irrelevant points and highlight the structure of the visible points.	97
6.7	Billboard Occlusion Algorithm used to hide billboards that are occluded by other billboards from the camera view	99
6.7	Billboard Occlusion Algorithm used to hide billboards that are occluded by other billboards from the camera view	100
6.8	Billboard Offset Algorithm used to offset billboards in case the billboard is occluded by the points in the point cloud	101
6.8	Billboard Offset Algorithm used to offset billboards in case the billboard is occluded by the points in the point cloud	102
7.1	Aardvark Media App Composition [Ort17]	104
8.1	Age distribution chart of the user study participants.	110
8.2	Chart showing how often the participants use computers, virtual reality, and augmented reality.	111
8.3	This chart shows for what purposes did the participants use virtual reality or augmented reality.	111
8.4	SUS Score curve showing the corresponding percentile ranks and grades [Sau18]	116
8.5	Graphical representations of the responses concerning the SUS for each individual question.	117
		137

8.5	Graphical representations of the responses concerning the SUS for each individual question.	118
8.5	Graphical representations of the responses concerning the SUS for each individual question.	119
8.5	Graphical representations of the responses concerning the SUS for each individual question.	120
8.6	Table illustrating the participants' responses to the interactions questionnaire	123
8.7	Table illustrating the participants' responses to the visualizations questionnaire	124
8.8	Bar chart representing the preference for most expressive and useful visualization technique for each participant.	125
8.9	Bar chart representing the preference for most expressive and informative rendering technique for each participant.	126
8.10	Table illustrating the participants' responses to the general questionnaire	127

List of Tables

5.1	All interaction options for both main and secondary controller	85
8.1	The table displays the individual SUS score of each participant as well as the average SUS score of all 10 participants.	121

List of Algorithms

6.1 Billboard Occlusion Algorithm	98
---	----

Acronyms

AIDA Asteroid Impact and Deflection Assessment. 35

API Application Programming Interface. 22, 23, 103

AR Augmented Reality. 17, 110

CSV Comma-Separated Values. 23

DART Double Asteroid Redirection Test. 35, 36, 135

DOF Degrees of Freedom. 5, 8

DOM Document Object Model. 20

ESA European Space Agency. 35, 36, 135

fMRI Functional Magnetic Resonance Imaging. 11

FOR Field of Regard. 13, 14

FOV Field of View. 13

GLSL OpenGL Shading Language. 103

GPU Graphics Processing Unit. 98, 100

GUI Graphical User Interface. 22

HMD Head-Mounted Display. 21, 25, 103

I/O Input/Output. 8

IE Immersive Environment. 13, 14, 26

JSON JavaScript Object Notation. 20, 22, 23

- MRI** Magnetic Resonance Imaging. 13
- NASA** National Aeronautics and Space Administration. 35, 36, 135
- SSQ** Simulator Sickness Questionnaire. 24–26
- SUS** System Usability Scale. 109, 116–121, 129, 137–139
- UI** User Interface. 4, 6, 18, 39, 42, 43, 46, 50, 51, 55, 65, 69–74, 79, 81, 84, 89, 91, 92, 103, 104, 113, 115, 128, 131, 133, 135, 136
- VR** Virtual Reality. 1, 2, 4–26, 32, 33, 36, 37, 39, 48, 50–52, 55, 65, 68, 84, 100, 103–112, 114, 115, 117, 124, 127–129, 131–133, 135, 136

Bibliography

- [ACC99] Laura L. Arns, Dianne Cook, and Carolina Cruz-Neira. The benefits of statistical visualization in an immersive environment. In *IEEE Virtual Reality 1999 Conference*, pages 88–95. IEEE Computer Society, 1999.
- [APSK20] Vladislav Angelov, Emiliyan Petkov, Georgi Shipkovenski, and Teodor Kalushkov. Modern virtual reality headsets. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–5, 06 2020.
- [AWK⁺08] Claudia Armbrüster, Marc Wolter, Torsten W. Kuhlen, Will Spijkers, and Bruno Fimm. Depth perception in virtual reality: Distance estimations in peri- and extrapersonal space. *Cyberpsychology and behavior: the impact of the Internet, multimedia and virtual reality on behavior and society*, 11(1):9–15, 03 2008.
- [AYE20] Evangelos Alexiou, Nanyang Yang, and Touradj Ebrahimi. PointXR: A toolbox for visualization and subjective evaluation of point clouds in virtual reality. In *Twelfth International Conference on Quality of Multimedia Experience*, pages 1–6. IEEE, 05 2020.
- [BC03] Grigore Burdea and Philippe Coiffet. *Virtual reality technology*. Wiley, 2003.
- [BJR21] Peter Butcher, Nigel John, and Panagiotis Ritsos. VRIA: A web-based framework for creating immersive analytics experiences. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3213–3225, 01 2021.
- [BM07] Doug Bowman and Ryan McMahan. Virtual reality: How much immersion is enough? *Computer*, 40:36 – 43, 08 2007.
- [BM08] Louis Bavoil and Kevin Myers. Order independent transparency with dual depth peeling. *NVIDIA*, 01 2008.
- [Bro95] John Brooke. SUS: A quick and dirty usability scale. *Usability Evaluation In Industry*, 189, 11 1995.

- [BV17] Leif Berg and Judy Vance. Industry use of virtual reality in product design and manufacturing: a survey. *Virtual Reality*, 21, 03 2017.
- [CCAL12] Jian Chen, Haipeng Cai, Alexander Auchus, and David Laidlaw. Effects of stereo and screen size on the legibility of three-dimensional streamtube visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18:2130–2139, 12 2012.
- [CCB⁺19] Maxime Cordeil, Andrew Cunningham, Benjamin Bach, Christophe Hurter, Bruce H. Thomas, Kim Marriott, and Tim Dwyer. IATK: an immersive analytics toolkit. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, pages 200–209. IEEE, 03 2019.
- [CCD⁺17] Maxime Cordeil, Andrew Cunningham, Tim Dwyer, Bruce H. Thomas, and Kim Marriott. ImAxes: immersive axes as embodied affordances for interactive multivariate data visualisation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 71–83. ACM, 2017.
- [CDH⁺19] Marco Cavallo, Mishal Dholakia, Matous Havlena, Kenneth Ocheltree, and Mark Podlaseck. Immersive Insights: A hybrid analytics system for collaborative exploratory data analysis. In *25th ACM Symposium on Virtual Reality Software and Technology*, pages 1–12. ACM, 11 2019.
- [CGBL98] Sarah S. Chance, Florence Gaunet, Andrew C. Beall, and Jack M. Loomis. Locomotion mode affects the updating of objects encountered during travel: The contribution of vestibular and proprioceptive inputs to path integration. *Presence*, 7(2):168–178, 1998.
- [CTL17] Rory M. S. Clifford, Nikita Mae B. Tuanquin, and Robert W. Lindeman. Jedi force extension: Telekinesis as a virtual reality interaction metaphor. In *2017 IEEE Symposium on 3D User Interfaces*, pages 239–240. IEEE Computer Society, 01 2017.
- [Dan19] Ninad Janardan Dani. Impact of virtual reality on gaming. *International Research Journal of Engineering and Technology (IRJET)*, pages 2033–2036, 12 2019.
- [Dav11] Jason Davies. Parallel coordinates with reordable axes. <https://bl.ocks.org/jasondavies/1341281>, 2011. Accessed: 2021-12-04.
- [DDC⁺14] Ciro Donalek, S. George Djorgovski, Alex Cioc, Anwell Wang, Jerry Zhang, Elizabeth Lawler, Stacy Yeh, Ashish Mahabal, Matthew J. Graham, Andrew J. Drake, Scott Davidoff, Jeffrey S. Norris, and Giuseppe Longo. Immersive and collaborative data visualization using virtual reality platforms. In *2014 IEEE International Conference on Big Data*, pages 609–614. IEEE Computer Society, 10 2014.

- [DJK⁺06] Çağatay Demiralp, Cullen Jackson, David Karelitz, Song Zhang, and David Laidlaw. Cave and fishtank virtual-reality displays: A qualitative and quantitative comparison. *IEEE Transactions on Visualization and Computer Graphics*, 12:323–30, 05 2006.
- [DMS⁺18] Sören Discher, Leon Masopust, Sebastian Schulz, Rico Richter, and Jürgen Döllner. A point-based and image-based multi-pass rendering technique for visualizing massive 3d point clouds in vr environments. *26th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2018*, 06 2018.
- [DODB20] Juan Manuel Davila Delgado, Lukumon Oyedele, Peter Demian, and Thomas Beach. A research agenda for augmented and virtual reality in architecture, engineering and construction. *Advanced Engineering Informatics*, 45:101–122, 2020.
- [DRST14] Steve Dübel, Martin Rohlig, Heidrun Schumann, and Matthias Trapp. 2d and 3d presentation of spatial data: A systematic review. In *2014 IEEE VIS International Workshop on 3DVis*, pages 11–18. IEEE, 11 2014.
- [DST19] Tor Dagsson Sundman and Joel Thunander. Investigating how impostor techniques are used to optimize the performance of video games. Degree work in technology - basic level, KTH, School of Electrical Engineering and Computer Science (EECS), 2019.
- [EDC⁺19] Mohamed El Beheiry, Sébastien Doutreligne, Clément Caporal, Cécilia Ostertag, Maxime Dahan, and Jean-Baptiste Masson. Virtual reality: Beyond visualization. *Journal of Molecular Biology*, 431(7):1315 – 1321, 2019.
- [ESA19a] ESA. Asteroid impact and deflection assessment (AIDA). https://www.esa.int/Safety_Security/Hera/Asteroid_Impact_Deflection_Assessment_AIDA_collaboration, 2019. Accessed: 2021-12-01.
- [ESA19b] ESA. Asteroid impact and deflection assessment (AIDA) mission overview. https://www.esa.int/ESA_Multimedia/Images/2019/01/Asteroid_Impact_Deflection_Assessment_AIDA_collaboration, 2019. Accessed: 2021-12-01.
- [ESA19c] ESA. Esa’s planetary defence mission. https://www.esa.int/Safety_Security/Hera, 2019. Accessed: 2021-12-01.
- [FHF⁺18] Daniel Freeman, Polly Haselton, Jason Freeman, Bernhard Spanlang, Sameer Kishore, Emily Albery, Megan Denne, Poppy Brown, Mel Slater, and Alecia Nickless. Automated psychological therapy using immersive

- virtual reality for treatment of fear of heights: a single-blind, parallel-group, randomised controlled trial. *The Lancet Psychiatry*, 5(8):625–632, 2018.
- [For15] Ford. Virtual manufacturing technology. <https://fordmediacenter.nl/wp-content/uploads/2015/07/Virtual-Manufacturing-Forum-Fact-Sheet1948.pdf>, 2015. Accessed: 2021-11-30.
- [FP21] Adrien Fonnet and Yannick Prié. Survey of immersive analytics. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2101–2122, 2021.
- [FSB16] Mauricio Frigo, Ethel Silva, and Gustavo Barbosa. Augmented reality in aerospace manufacturing: A review. *Journal of Industrial and Intelligent Information*, 4:125–130, 01 2016.
- [GGH⁺18] Jože Guna, Gregor Geršak, Iztok Humar, Jeungeun Song, Janko Drnovšek, and Matevz Pogacnik. Influence of video content type on users’ virtual reality sickness perception and physiological response. *Future Generation Computer Systems*, 91, 09 2018.
- [Goo08] Google. Google forms. <https://www.google.com/forms/about/>, 2008. Accessed: 2021-12-02.
- [Gro14] David Gross. Technology management and user acceptance of virtual environment technology. In *Handbook of Virtual Environments - Design, Implementation, and Applications, Second Edition*, pages 493–504. CRC Press, 2014.
- [GSB⁺18] Ali Garavand, Mahnaz Samadbeik, Peivand Bastani, Donyaa Yaghoobi, Shahabeddin Abhari, and Rita Rezaee. The applications of virtual reality technology in medical groups teaching. *Journal of advances in medical education and professionalism*, 6, 07 2018.
- [HBV20] François Homps, Yohan Beugin, and Romain Vuillemot. ReViVD: exploration and filtering of trajectories in an immersive environment using 3d shapes. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, pages 729–737. IEEE, 03 2020.
- [HHRD⁺18] Christophe Hurter, Nathalie Henry Riche, Steven Drucker, Maxime Cordeil, Richard Alligier, and Romain Vuillemot. FiberClay: sculpting three dimensional trajectories to reveal structural insights. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):704–714, 08 2018.

- [Hol18] Yan Holtz. Histogram with tooltip in d3.js. https://www.d3-graph-gallery.com/graph/histogram_tooltip.html, 2018. Accessed: 2021-12-04.
- [HPPS18] Vinutha H. P., Basavaraju Poornima, and B. Sagar. *Information and Decision Sciences*, chapter Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset, pages 511–518. 01 2018.
- [HRB⁺06] Hunter Hoffman, Todd Richards, Aric Bills, Trevor Oostrom, Jeff Magula, Eric Seibel, and Sam Sharar. Using fmri to study the neural correlates of virtual reality analgesia. *CNS spectrums*, 11:45–51, 02 2006.
- [HSMT15] Georg Haaser, Harald Steinlechner, Stefan Maierhofer, and Robert Tobler. An incremental rendering VM. In *Proceedings of the 7th Conference on High-Performance Graphics*, pages 51–60. ACM, 08 2015.
- [HVV21] Filip Hácha, Petr Vanecek, and Libor Váša. A Virtual Reality Platform for Immersive Education in Computer Graphics. In *Eurographics 2021 - Education Papers*. The Eurographics Association, 2021.
- [HWK09] Bernd Hentschel, Marc Wolter, and Torsten Kuhlen. Virtual reality-based multi-view visualization of time-dependent simulation data. In *IEEE Virtual Reality Conference 2009*, pages 253–254. IEEE Computer Society, 03 2009.
- [Inc15] InstaVR Inc. Htc vive. [https://www.instavr.co/articles/general/how-to-make-and-play-htc-vive-apps#iLightbox\[gallery15157\]/null](https://www.instavr.co/articles/general/how-to-make-and-play-htc-vive-apps#iLightbox[gallery15157]/null), 2015. Accessed: 2021-12-05.
- [Iri14] IrisVR. The importance of frame rates. <https://help.irisvr.com/hc/en-us/articles/215884547-The-Importance-of-Frame-Rates>, 2014. Accessed: 2021-11-30.
- [JKBWG18] Shaun Jerdan, Maged Kamel Boulos, Hugo Woerden, and Mark Grindle. Head mounted virtual reality and mental health: Critical review of current research. *Journal of Medical Internet Research*, 6:14, 07 2018.
- [KG01] Andreas König and Eduard Gröller. Mastering transfer function specification by using volumepro technology. In *Spring Conference on Computer Graphics*, volume 17, pages 279–286, 04 2001.
- [KLBL93] Robert S. Kennedy, Norman E. Lane, Kevin S. Berbaum, and Michael G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, 1993.

- [KPCC18] Hyun K. Kim, Jaehyun Park, Yeongcheol Choi, and Mungyeong Choe. Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment. *Applied ergonomics*, 69:66–73, 05 2018.
- [LaV00] Joseph J. LaViola. A discussion of cybersickness in virtual environments. *Association for Computing Machinery*, 32(1):47–56, 01 2000.
- [LBS14] Bireswar Laha, Doug Bowman, and John Socha. Effects of vr system fidelity on analyzing isosurface visualization of volume datasets. *IEEE Transactions on Visualization and Computer Graphics*, 20:513–22, 04 2014.
- [LHC⁺21] Benjamin Lee, Xiaoyun Hu, Maxime Cordeil, Arnaud Prouzeau, Bernhard Jenny, and Tim Dwyer. Shared surfaces and spaces: Collaborative data visualisation in a co-located immersive environment. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1171–1181, 2021.
- [LKM⁺17] Joseph LaViola, Ernst Kruijff, Ryan P. McMahan, Doug A. Bowman, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley usability and HCI series. Addison-Wesley, 2017.
- [LPF⁺09] Joseph LaViola, Prabhat, Andrew Forsberg, David Laidlaw, and Andries van Dam. *Trends in Interactive Visualization: State-of-the-Art Survey*, chapter Virtual Reality-Based Interactive Scientific Visualization Environments, pages 225–250. Springer London, 10 2009.
- [LTRCB19] Thibault Louis, Jocelyne Troccaz, Amélie Rochet-Capellan, and François Bérard. Is it real? measuring the effect of resolution, latency, frame rate and jitter on the presence of virtual entities. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, page 5–16. Association for Computing Machinery, 2019.
- [MG99] Tomasz Mazuryk and Michael Gervautz. Virtual reality - history, applications, technology and future. 12 1999.
- [MJPR16] Omar Mures, Alberto Jaspe, Emilio Padrón, and Juan Rabuñal. *Handbook of Research on Visual Computing and Emerging Geometrical Design Tools*, chapter Virtual Reality and Point-Based Rendering in Architecture and Heritage, pages 549–565. IGI Global, 04 2016.
- [MM11] Jason D. Moss and Eric R. Muth. Characteristics of head-mounted displays and their effects on simulator sickness. *Human Factors*, 53(3):308–319, 2011.
- [MS19] Thomas I. Maindl and Christoph M. Schäfer. Momentum enhancement for the DART kinetic impactor. In *EPSC-DPS Joint Meeting 2019*, volume 13, 09 2019.

- [MSD⁺18] Kim Marriott, Falk Schreiber, Tim Dwyer, Karsten Klein, Nathalie Henry Riche, Takayuki Itoh, Wolfgang Stuerzlinger, and Bruce H. Thomas, editors. *Immersive Analytics*, volume 11190 of *Lecture Notes in Computer Science*. Springer, 2018.
- [MV19] Miroslav Musalek and Lubomir Vasek. Possibilities of using virtual reality as a means for therapy from fear of spiders. *MATEC Web of Conferences*, 292:01041, 01 2019.
- [NAS17] NASA. Nasa’s double asteroid redirection test. <https://www.nasa.gov/planetarydefense/dart>, 2017. Accessed: 2021-12-01.
- [OJC⁺17] Patrick O’Leary, Sankhesh Jhaveri, Aashish Chaudhary, William R. Sherman, Ken Martin, David Lonie, Eric T. Whiting, James H. Money, and Sandy McKenzie. Enhancements to VTK enabling scientific visualization in immersive environments. In *2017 IEEE Virtual Reality*, pages 186–194. IEEE Computer Society, 01 2017.
- [OOKO15] Ekaterina Olshannikova, Aleksandr Ometov, Yevgeni Koucheryavy, and Thomas Olsson. Visualizing big data with augmented and virtual reality: challenges and research agenda. *Journal of Big Data*, 2, 12 2015.
- [Ort17] Thomas Ortner. Aardvark media. <https://github.com/aardvark-platform/aardvark.docs/wiki/Aardvark.Media>, 2017. Accessed: 2021-12-05.
- [PCC92] Randy Pausch, Thomas Crea, and Matthew Conway. A literature survey for virtual environments: Military flight simulator visual systems and simulator sickness. *Presence*, 1:344–363, 01 1992.
- [PLE⁺19] Arnaud Prouzeau, Antoine Lhuillier, Barrett Ens, Daniel Weiskopf, and Tim Dwyer. Visual link routing in immersive visualisations. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, pages 241–253. ACM, 11 2019.
- [PNFG15] Michelle Portman, Asya Natapov, and Dafna Fisher-Gewirtzman. To go where no man has gone before: Virtual reality in architecture, landscape architecture and environmental planning. *Computers, Environment and Urban Systems*, 54, 06 2015.
- [PPM15] Bruno Patrão, Samuel Lago Pedro, and Paulo Menezes. How to deal with motion sickness in virtual reality. In *22nd Portuguese Meeting on Computer Graphics and Interaction 2015*. The Eurographics Association, 11 2015.
- [RBLN04] Dheva Raja, Doug Bowman, John Lucas, and Chris North. Exploring the benefits of immersion in abstract information visualization. *8th Int’l Immersive Projection Technology Workshop (IPT ’04)*, 01 2004.

- [RCGMBB13] Mar Rus-Calafell, José Gutiérrez-Maldonado, Cristina Botella, and Rosa Baños. Virtual reality exposure and imaginal exposure in the treatment of fear of flying: A pilot study. *Behavior modification*, 04 2013.
- [RKSB12] Eric Ragan, Regis Kopper, Philip Schuchardt, and Doug Bowman. Studying the effects of stereo, head tracking, and field of regard on a small-scale spatial judgment task. *IEEE Transactions on Visualization and Computer Graphics*, 19(5):886–896, 08 2012.
- [RMG⁺16] Pedro J. Rosa, Diogo Morais, Pedro Gamito, Jorge Oliveira, and Tomaz Saraiva. The immersive virtual reality experience: A typology of users revealed through multiple correspondence analysis combined with cluster analysis technique. *Cyberpsychology, Behavior, and Social Networking*, 19(3):209–216, 2016. PMID: 26985781.
- [RO16] Lisa Rebenitsch and Charles Owen. Review on cybersickness in applications and visual displays. *Virtual Reality*, 20, 06 2016.
- [Sai18] Mustafa Saifee. VR-Viz: visualization system for data visualization in vr. Master’s thesis, Aalto University. School of Arts, Design and Architecture, 2018.
- [Sau18] Jeff Sauro. 5 ways to interpret a SUS score. <https://measuringu.com/interpret-sus-score/>, 2018. Accessed: 2021-12-02.
- [SB07] Philip Schuchardt and Doug Bowman. The benefits of immersion for spatial understanding of complex underground cave systems. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 121–124. ACM, 01 2007.
- [SCS⁺10] Richard Szeliski, Brian Curless, Steven M. Seitz, Noah Snavely, Yasutaka Furukawa, and Sameer Agarwal. Reconstructing rome. *Computer*, 43(06):40–47, 06 2010.
- [Shn96] Ben Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.
- [SLC⁺18] Ronell Sicat, Jiabao Li, JunYoung Choi, Maxime Cordeil, Won-Ki Jeong, Benjamin Bach, and Hanspeter Pfister. DXR: A toolkit for building immersive data visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):715–725, 08 2018.
- [SMWH17] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2017.

- [SNL18] Jan-Philipp Stauffert, Florian Niebling, and Marc Erich Latoschik. Effects of latency jitter on simulator sickness in a search task. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 121–127, 2018.
- [SSB⁺20] Dimitrios Saredakis, Ancret Szpak, Brandon Birckhead, Hannah A. D. Keage, Albert Rizzo, and Tobias Loetscher. Factors associated with virtual reality sickness in head-mounted displays: A systematic review and meta-analysis. *Frontiers in Human Neuroscience*, 14:96, 2020.
- [SSCG17] Jonathan Stets, Yongbin Sun, Wiley Corning, and Scott Greenwald. Visualization and labeling of point clouds in virtual reality. In *SIGGRAPH Asia 2017 Posters*, pages 31:1–2. ACM, 11 2017.
- [Ste15] Steamworks. OpenVR. <https://partner.steamgames.com/doc/features/steamvr/openvr>, 2015. Accessed: 2021-12-05.
- [Ste18] Scott Stein. The 51 best VR games. <https://www.cnet.com/pictures/best-vr-games/>, 2018. Accessed: 2021-11-30.
- [Sut65] Ivan E. Sutherland. The ultimate display. In *Proceedings of the IFIP Congress*, pages 506–508, 1965.
- [UKF⁺17] Will Usher, Pavol Klacansky, Frederick Federer, Peer-Timo Bremer, Aaron Knoll, Alessandra Angelucci, Valerio Pascucci, and Jeff Yarch. A virtual reality visualization tool for neuron tracing. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):994–1003, 08 2017.
- [Var19] Varwin. VR controllers. <https://varwin.readthedocs.io/en/latest/manuals/controllers.html>, 2019. Accessed: 2021-12-05.
- [Vas21] Elitza Vasileva. Evaluation procedure. https://docs.google.com/document/d/1noF4hywESUsd-ksz9SNY_98t334m7mybtae7U5mLupw/edit, 2021. Accessed: 2021-12-02.
- [Vis17] Visualise. Virtual reality in healthcare. <https://visualise.com/virtual-reality/virtual-reality-healthcare>, 2017. Accessed: 2021-11-30.
- [VRV14] VRVis. Fshade. <https://fshade.org/>, 2014. Accessed: 2021-12-05.
- [VS20] Elitza Vasileva and Harald Steinlechner. Impact simulation. https://github.com/aardvark-community/impact-simulation/tree/restructure_for_octree, 2020. Accessed: 2021-11-30.
- [WBB06] Hans Peter Wyss, Roland Blach, and Matthias Bues. iSith - Intersection-based spatial interaction for two hands. In *3D User Interfaces (3DUI'06)*, pages 59 – 61, 04 2006.

- [WPL⁺10] Manuela Waldner, Werner Puff, Alexander Lex, Marc Streit, and Dieter Schmalstieg. Visual links across applications. In *Proceedings of the Graphics Interface 2010 Conference*, pages 129–136. ACM, 01 2010.
- [WQOS19] Florian Wirth, Jannik Quehl, Jeffrey Ota, and Christoph Stiller. PointAtMe: Efficient 3d point cloud labeling in virtual reality. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1693–1698, 2019.
- [ZDK⁺01] Song Zhang, Çağatay Demiralp, Daniel Keefe, Manuel DaSilva, David Laidlaw, Benjamin Greenberg, Peter Basser, Carlo Pierpaoli, Ennio Chiocca, and Thomas Deisboeck. An immersive virtual environment for dt-mri volume visualization applications: A case study. In *12th IEEE Visualization Conference*, pages 437–440. IEEE Computer Society, 01 2001.
- [Zel92] David Zeltzer. Autonomy, interaction, and presence. *Presence: Teleoperators and Virtual Environments*, 1:127–132, 1992.
- [Zha17] Hui Zhang. Head-mounted display-based intuitive virtual reality training system for the mining industry. *International Journal of Mining Science and Technology*, 27, 05 2017.