Explorative Point Cloud Virtual Reality: Immersive Visual Insight

Evaluating User Perception, Interaction, and Immersion with VR and Omnibase

Synthesis Project (GEO1101) - Report

Group 5

Michalis Michalas - 6047378

Lotte de Niet - 5025354

Javier Martinez - 6055613

Zhuoyue Wang - 6093590

Bart Manden - 5236371

Explorative Point Cloud Virtual Reality: Immersive Visual Insight

Evaluating User Perception, Interaction and Immersion with VR and Omnibase Synthesis Project (GEO1101)

Group 5

Bart Manden	-	5236371
Javier Martinez	-	6055613
Michalis Michalas	-	6047378
Lotte de Niet	-	5025354
Zhuoyue Wang	-	6093590

November 15, 2024

Acknowledgments

We would like to thank Geodelta for their support throughout this project, especially for helping us troubleshoot technical issues when we needed it most. We also want to thank our supervisors for their guidance in shaping the project and helping us focus on the important aspects. Their advice and encouragement were invaluable to completing this work.

Contents

List of Tables

List of Figures

1 Introduction

Interpreting spatial data is a challenge faced by both experts and the general public, primarily because of the limitations in the translation from three-dimensional (3D) space to a two-dimensional (2D) screen. These visualizations often lack a sense of scale and hinder depth perception. While various applications attempt to overcome these challenges, there remains uncertainty about their effectiveness compared to a more immersive method, such as Virtual Reality (VR).

One of these platforms is Omnibase (Geodelta, 2024), developed by Geodelta, which integrates various geospatial data including point clouds, panoramic images, nadir and oblique photos, into a single environment. This tool enables access to geo-information for several stakeholders and use cases, such as the upkeep of the cadastral register.

Point clouds are a particularly complex data type, which plays a vital role in areas such as geospatial analysis and surveying. In Omnibase, point clouds primarily support municipal boundary measurements, like the BGT. The addition of VR is intended to deepen the explorative capabilities of the point cloud, especially for users who are unfamiliar with point cloud data. Explorative point clouds refer to the use of point cloud data in immersive environments to enhance the visualization and understanding of spatial data (Nebiker et al., 2010). This method leverages VR technologies to provide users with a realistic and interactive experience, facilitating a deeper exploration of urban models. Building on the new paradigm discussed by Nebiker et al. (2010), this research aims to explore how effectively users can understand and measure in point cloud environments, specifically point clouds—using VR compared to traditional 2D screens and the Omnibase multi-view system.

This project has both theoretical and practical aims. As the theoretical objective, surveys will be conducted to evaluate the user experience of interacting with point clouds in a Virtual Reality (VR) environment compared to traditional 2D platforms. Participants will be selected from two distinct groups to provide a comprehensive perspective: users who have experience dealing with point cloud data, and those without. The aim is to gather feedback on the usability, ease of navigation, and potential learning curve associated with using VR for point cloud interaction. This will create an understanding of the advantages and limitations of VR across different levels of user experience.

The practical aspect will involve the development of a proof-of-concept spatial measuring feature in VR for Omnibase. To achieve this, we will utilize Potree (Schütz, 2016), a web-based point cloud rendering tool, which efficiently handles large point cloud datasets. Potree's integration of Level of Detail (LoD) and octree structure allows us to view point clouds in VR and incorporate our own VR measuring functionality using WebXR.

This project will involve several stakeholders. Geodelta, the developers of Omnibase, who will oversee the integration and ensure that VR functionalities align with their platform and needs. As well as their clients, the municipalities and provinces who benefit from enhanced interaction with geo-information. With the potential to further expand outreach by helping (inexperienced) clients to interact with geo-information in a more intuitive way.

The central research question in this study is: *How does the use of Virtual Reality, compared to Omnibase's multi-view, affect user perception, interaction and relative measurement accuracy, for users that are either familiar or unfamiliar with point clouds?*

2 Research Aim and Scope

In this chapter, the problem statement, research question, and sub-questions are defined to guide the direction of the study. The scope and objectives are outlined, focusing on evaluating user experiences and measurement accuracy in VR compared to Omnibase for point cloud interactions. Key objectives include the development and testing of measurement tools, enhancements to navigation controls, and the collection of user feedback. Moreover, the chapter connects these objectives to relevant courses that provide essential knowledge in VR development, spatial data processing, and user experience design.

2.1 Problem Definition

Accurately interpreting and measuring spatial data remains a challenge to both spatial data experts and the general public. Point clouds, which represent 3D spatial information, are useful for applications such as urban planning, surveying and environmental monitoring. However, when translated to a 2D screen, point clouds can become difficult to interpret, due to issues like:

- Grasping the scale of the data, especially when zooming in and out.
- Depth perception problems due to the point cloud appearing flat, making it difficult to judge relative distances. As Burwell (2016) mentioned "A 2D display will always lack this depth cue, regardless of any other cues present".
- Navigation and interaction difficulties due to the use of a mouse and keyboard, and relying on methods like multiple view ports or rotation to retrieve more insight into the depth.

These problems become even more apparent when point clouds are used for intricate tasks like municipal boundary measurements. Omnibase attempts to overcome these challenges by offering multiple perspectives. It remains, however, a 2D screen that lacks the depth cue. An alternative approach might be better suited to better align with the complexity of point cloud data viewing and interaction.

Additionally, there is a knowledge gap regarding the use of Virtual Reality (VR) for point cloud interpretation and taking measurements. VR is a relatively new tool for spatial analysis, showing promise in overcoming the limitations of 2D visualizations. It is particularly effective for enhancing depth perception and enabling immersive navigation. However, its application to point cloud interactions remains largely unknown.

Currently, Potree offers basic VR functionality for point clouds, allowing users to view and move through any point cloud on the web. However, these implementations are limited to this viewing. Further functionality such as taking measurements and annotating that are present in desktop Potree, are not included in VR.

The current desktop methods of viewing and interacting with point clouds do not take full advantage of VR's ability to help users understand point clouds in a more intuitive way. Research into adding measurements into VR in Potree could help bridge this gap.

2.2 Research Question

This study addresses the central research question: *How does the use of Virtual Reality, compared to Omnibase's multi-view, affect user perception, interaction and relative measurement accuracy, for users that are either familiar or unfamiliar with point clouds?*

2.2.1 Subquestions

To gain more knowledge into the use of VR for point clouds, especially to determine whether VR can provide a more intuitive environment to inexperienced users, the following subquestions will guide the research:

- **Evaluating user comfort and immersion in VR when viewing point clouds.**
  - *How does the physical comfort of VR compare with that of Omnibase's multi-view?*
  - *What are users' subjective experiences of immersion when interacting with point clouds in VR versus a multi-view platform?*
- **Assessing the ease of navigation and control methods in the different platforms**
  - *How do users perceive the intuitiveness of navigation and control in VR compared to Omnibase multi-view?*
- **Assessing the accuracy of spatial measurements across the two platforms**
  - *How does the consistency of spatial measurements in 3D environments differ between VR and multi-view platforms, particularly in terms of depth accuracy and relative measurement discrepancies?*
  - *Do inexperienced users find VR more intuitive for taking point cloud measurements than the multi-view approach?*

2.2.2 Practical Objectives

Objectives to develop a proof-of-concept spatial measuring feature in VR for Omnibase.

- **Intergrate VR functionality into a web environment based on Potree.**
  As part of this research, one key objective is to integrate Virtual Reality (VR) functionality into a web-based environment. The choice to use Potree is further discussed in section 3.3.3 but is also supported by the fact that Omnibase is based on Potree, which would allow for an easier integration.
- **Develop a measuring feature.**
  To compare the accuracy of spatial measurements in point clouds in VR, a measuring feature was developed in the web environment, as this functionality is absent in the current version of Potree.

This research aims to address the challenge of understanding complex spatial environments, specifically point clouds. By comparing the use of VR versus the Omnibase multi-view, it can be determined whether the intuitive nature of VR provides the necessary support needed for in-depth point cloud exploration and interaction. Additionally, by including two levels of user expertise - Users experienced with point clouds and users who have not previously interacted with point clouds - this study aims to assess whether VR is especially helpful to those unfamiliar with point clouds.

2.3 Scope

The scope of this project includes the following:

Primary Objectives:

- **Qualitative and Quantitative Research:** Conduct a survey comparing people with and without experience with point clouds, in Omnibase's multi-perspective and Virtual Reality environments.
- **VR Environment Development:** Adjust the Potree codebase to be able to walk in a web-based VR environment, this includes: adding rotation to the controllers, slowing down the speed and restricting the height.

Essential Objectives:

- **Measurement Functionality:** Display a visual ray from the VR controllers to improve depth perception during interaction and include the ability to make measurements between two points.

Additional Objectives:

- **Collision Detection:** Implement wall collision or restrict the walkable area, in order to stay in the bounds of the point cloud.
- **Polygon Measurement:** Include the potential for polygon measurements and the calculation for its area.
- **Dynamic Height:** Restrict the height not to a fixed number but adjust it dynamically to the ground height.
- **Automatic Starting Position:** Automatically adjust the starting position according to the extent of the point cloud.

Exclusions:

- **Complex Algorithm Implementations:** This study will not include more complex improvements to the rendering or display, such as Gaussian splatting.
- **Real-Time editing:** Real-time editing of point cloud data in VR is not included.
- **Integrate VR Into Omnibase:** This project will not attempt to fully integrate the VR functionality into the Omnibase platform. As this research is based on Potree, like Omnibase, the results do serve as a proof of concept.

2.4 Relevant Courses

The objectives of this project are closely aligned with several courses from the MSc Geomatics program. GEO1001: Sensing Technologies and Geometry provided essential knowledge on handling point clouds, crucial for their efficient rendering and optimization in VR. GEO1006: Geo Database Management Systems introduced octree structures and Level of Detail (LoD) techniques, essential for point cloud rendering, along with advanced topics like 3D modeling and spatial-temporal modeling. GEO1007: Geo Web Technology covered web standards such as WebGL and glTF for rendering 3D geospatial data in VR, along with geo-web service integration to support spatial decision-making in interactive VR applications. Additionally, various other courses provided the programming experience necessary to work on the VR implementation.

3 Theory

In Chapter 2, the challenges of effectively visualizing and interacting with point clouds on 2D screens were discussed, emphasizing the need for an alternative approach such as VR. The chapter outlines the research aim, poses specific questions, and defines the objectives for implementing an interactive VR solution for spatial data. Chapter 3 provides the theoretical background necessary for this implementation, covering tools like Potree and Omnibase that support point cloud visualization, and examining how VR can enhance depth perception and interactivity.

3.1 Omnibase

Omnibase, developed by Geodelta, provides municipalities, provinces, and other institutions with a platform for maintaining various geospatial data sources. Unlike traditional data collection methods such as land surveying or stereo imaging, Omnibase integrates modern data sources to deliver a more efficient and intuitive user experience. For example, one of their datasets, showcased in Figure 1, includes a point cloud representing a section of Vienna.

Figure 1: Omnibase interface (Geodelta, 2024)

This research primarily focuses on the multi-view functionality of Omnibase, which allows users to split screens to view and measure efficiently from different data sources simultaneously. This feature, as shown in Figure 2, is especially relevant when comparing the usability of Omnibase's point cloud navigation and measurement capabilities against VR environments. The multi-view capability of Omnibase allows users to precisely select points for area measurements by providing two separate views, enhancing accuracy in terms of depth perception. By navigating within both views after completing measurements, users can verify if their selected points are depth-consistent or if they mistakenly included points that appeared to be part of the area but were not. This dual-view approach reduces errors and ensures reliable spatial analysis across complex datasets.

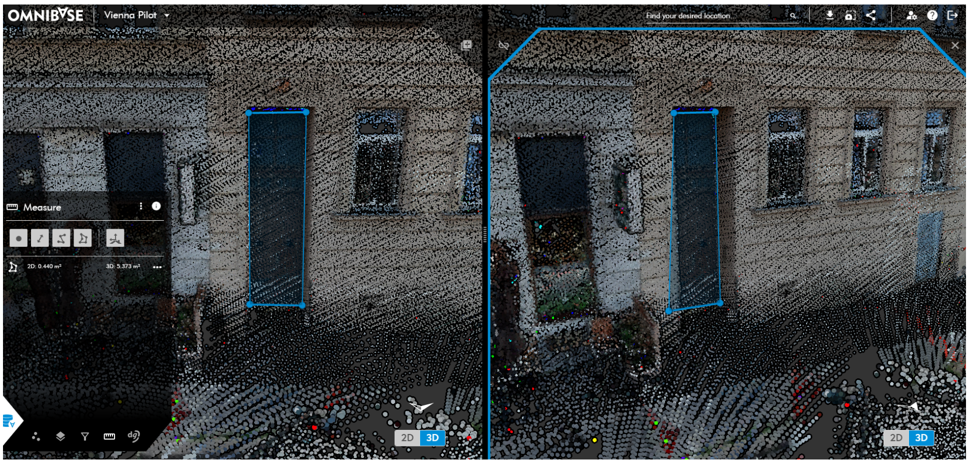


Figure 2: Omnibase Multi-view measuring interface (Geodelta, 2024)

3.2 Point Clouds

In this section the used webtool Potree is further elaborated, including the selection and measuring functionality. Additionally, the explorative capabilities of point clouds in VR are highlighted.

3.2.1 Potree

Potree is a widely used open-source tool for rendering large point cloud datasets directly in web browsers. Developed by Schütz (2016), Potree employs WebGL and Three.js for efficient visualization of point clouds while maintaining high performance on the web. Its octree structure and Level of Detail (LoD) management enable dynamic loading and rendering of only the necessary points based on the user's view, reducing computational load while maintaining visual clarity. This makes it suitable for handling large-scale datasets such as those used in urban planning or geospatial analysis.

The value of point cloud datasets lies in their ability to preserve rich environmental details without requiring intermediate modeling steps. As noted in Meijers' and Verbee's lecture on the direct and explorative use of point clouds (Meijers & Verbree, 2018), these datasets are highly effective for immediate analysis and decision-making due to their authenticity and comprehensiveness. Potree's visualization capabilities align with this potential, providing tools such as measurement functionalities (e.g., length, area, and volume calculations) that are ideal for web-based exploration, although these features are not yet fully extended to VR environments. Figure 3 shows the Potree interface, demonstrating its point cloud visualization and measurement capabilities in a desktop environment.

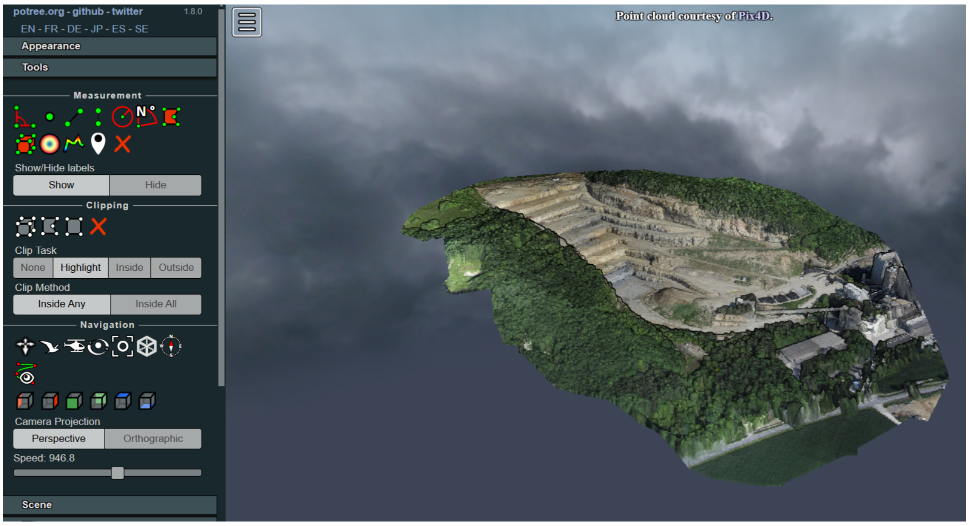


Figure 3: Potree interface (Schütz, 2016)

The concept of continuously managing levels of detail plays an important role in Potree's effectiveness, as described by van Oosterom et al. (2022). The use of continuous LoD allows the visualization to progressively refine as the user interacts with the dataset, providing a smoother experience by loading only the relevant point cloud details in real time. This approach not only optimizes computational efficiency but also enhances user interactivity by preventing excessive resource usage during zooming or panning actions. The insights from this study gave a better understanding of how Potree handles and displays large point clouds. While adjusting the rendering process is beyond the scope of this project, this knowledge helped to work more effectively with point clouds in both VR and web-based settings. It allowed for more informed choices when adapting Potree for VR, especially for adding measurement tools and improving user interactions.

However, although Potree includes built-in tools for spatial measurements that allow users to calculate distances, areas, and volumes within the 3D environment, these features are currently limited to traditional desktop settings and are not yet available in VR. This project aims to extend Potree's functionality by integrating VR-based measurement tools, allowing users to interact with and measure within point clouds more intuitively in a 3D immersive environment. Such integration will address the challenges of managing large datasets in VR, as noted by Papson et al. (2004), where balancing performance, accuracy, and user experience is crucial for handling data such as 3D models and point clouds in real time.

3.2.2 Selection and Measuring

The simplified point selection process in Potree involves using offscreen rendering to interact with large point clouds in a 3D environment. This process utilizes color-based picking by rendering points on an offscreen canvas, assigning unique colors to each point, and determining the selected point based on the pixel color under the user's cursor. The first step in point selection is rendering the scene offscreen. Potree uses the `OffscreenCanvas` API, allowing the rendering process to occur in a background thread, thereby improving performance for large point clouds. Using `OffscreenCanvas` enables Potree to offload the rendering task to a background thread. This is crucial for maintaining a responsive user interface, especially when dealing with large datasets in VR environments or real-time web applications. However, in VR environments, the point selection methodology differs from traditional web-based approaches like Potree, due to the difference in screens, as described further in chapter 5.

3.2.3 Explorative Point Clouds

Point clouds are often used as a means for further processing, but they also offer significant potential for direct visualization to support complex analyses. The rich point cloud paradigm, as articulated by Nebiker et al. (2010), advocates for treating dense, semantically-rich point clouds as dynamic, exploratory models rather than mere inputs for traditional 3D modeling. This shift emphasizes the utilization of point clouds in creating interactive and immersive environments that reflect the detailed reality of urban spaces. Due to the point clouds high accuracy in 3D positioning and the minimal need for human interaction in the data acquisition process, point clouds enable detailed, scalable representations that enhance depth perception and spatial understanding (Nebiker et al., 2010). Potree leverages these capabilities with its efficient LoD management, enabling users to navigate freely through point clouds with minimal geometric distortion. Integrating VR into Potree enhances the explorative capabilities of point clouds, providing an additional visualization technique for understanding complex spatial relationships.

3.3 Virtual Reality

Additional theories on VR point clouds, control techniques and the drawbacks are provided in this section.

3.3.1 VR Point Clouds

The use of VR for interacting with point clouds offers many advantages in terms of perception, interaction, and user experience. One of the most notable benefits is the enhanced depth perception VR provides, allowing users to better understand the spatial relationships within complex 3D datasets, such as large point clouds. In traditional 2D interfaces, perceiving depth and distance within point clouds can be challenging, but VR environments create an immersive experience where users can naturally manipulate, explore, and visualize these datasets using hand gestures and intuitive controls. Garrido et al. (2021) have demonstrated that working with point clouds in VR results in effortless depth perception and more intuitive manipulation of individual points.

Furthermore, VR systems allow for more efficient real-time interaction with large datasets stored on servers, ensuring that the user's local system is not overburdened. This is very useful when dealing with high-density point clouds, which can exceed millions of points, making them difficult to store and process locally. As demonstrated in other implementations (Silva & Sousa, 2020), VR enables detailed interactions, such as painting, removing points, and adjusting lighting in real time, providing a more engaging and efficient workflow for point cloud visualization and manipulation. Overall, VR creates a more engaging and natural user interface, enabling users to interact with large 3D datasets in ways that traditional desktop interfaces cannot match, enhancing both the speed and accuracy of tasks like annotation and exploration (Franzluebbers et al., 2022).

The research from Casado-Coscolla et al. (2023) focuses on the challenges and methods for displaying large indoor point clouds in VR. It explores different techniques for ensuring a smooth and efficient user experience, even when handling large datasets. The authors introduce two pre-computed strategies that leverage the static nature of indoor point clouds: a continuous level of detail data structure (CLOD) and a visibility map (V-Map). Although this project focused on outdoor environments rather than indoors, these insights help in understanding the challenges associated with rendering point clouds in immersive VR settings.

3.3.2 Controls

Movement types in VR play a crucial role in how effectively and comfortably users can interact within virtual environments. According to Bowman et al. (2021), different interaction techniques can be employed using six Degrees of Freedom (DoF) devices, primarily categorized into virtual hand and virtual pointing methods. Virtual hand techniques involve using the entire arm and hand to reach and grasp virtual objects, providing a more natural interaction. However, these techniques can become physically demanding over time. On the other hand, virtual pointing techniques, such as raycasting, allow users to select objects by pointing, reducing the need for full arm movement. By using only wrist rotations, these techniques make it easier to interact with objects that are out of direct reach, ultimately providing a more efficient interaction mechanism. Therefore, our implementation leverages virtual pointing techniques, specifically raycasting with a controller, as it offers an efficient and accessible interaction mechanism. Therefore, allowing users to interact with distant objects using minimal wrist movement rather than full arm extension, this method provides both ease of use and reduced physical strain, making it an optimal choice for a comfortable VR experience.

The movement types in VR explored by Ipsita et al. (2021) were in the context of interacting with scanned point clouds using the VRFromX system. Users employed a brush tool to select regions and manipulate virtual content, enabling free movement and orientation of the environment. Additionally, Oculus controllers facilitated navigation and the movement of objects in the virtual environment, promoting an immersive user experience through embodied interaction and hand gestures.

As Wang et al. (2023) report in their project PointShopAR, movement in VR can be managed through different interaction techniques. In their system, they employ touch-based and grabbing-based object manipulation. Touch-based interaction includes gestures for moving, rotating, and scaling virtual objects, whereas grabbing-based manipulation allows users to map the movement of their physical tablet to the virtual object's movement. This kind of interaction provides a more natural way for users to engage with the virtual environment, closely mimicking real-world behaviors.

Lastly, Gao et al. (2016) reports on different types of VR movement, specifically comparing static front view and oriented view sharing methods. The paper states that in the oriented view, users need to rotate their heads to align with the viewpoint of the remote worker. This approach provides a more immersive experience and allows the user to understand spatial relationships better by following the head movements. However, it also demands physical effort to continuously adjust viewpoints, which can affect the usability of tasks that require frequent adjustments. For this reason, it was decided to implement both options, so that small movements can be made naturally with the head, but more straining motions can also be done using the controller rotation.

3.3.3 Challenges and Considerations

Motion Sickness and Visual Fatigue
Despite its advantages, VR systems present several challenges. One major downside is motion sickness or nausea, most common in environments that involve exploring large datasets or frequent movement. This can occur due to a mismatch between what users see in the virtual world and what their bodies physically experience, leading to discomfort, as highlighted by de Haan (2009). Rapid movements or inconsistent frame rates can exacerbate these symptoms. Extended use of head-mounted displays can also cause visual fatigue, reducing user comfort and overall experience. These physiological effects are crucial considerations when designing user interfaces for VR environments, as they directly impact usability and long-term adoption of VR for complex tasks.

User Fatigue and Interaction Complexity
As introduced by Bowman et al. (2021), VR systems present several challenges, one of which is user fatigue, which is common with handheld devices, leading to muscular strain due to extended hand and arm movements. Techniques to fix hand positions and reduce trembling may increase the overall effort, decreasing user comfort during prolonged use. Another challenge is interacting without physical support, which can be tiring and uncomfortable. Techniques like using virtual hand metaphors can help stabilize user movements and make interactions easier, but often reduce accuracy.

Rendering Challenges and Platform Limitations
Another challenge relates to the rendering process, which originally happened in a web-based setting using Potree. Some projects, such as Garrido et al. (2021), have opted to use Unity Technologies (2024) for VR integration, as it offers open-source utilities for VR and point cloud management and leverages existing tools for handling 3D objects. However, Potree was used for this project because it facilitates easy sharing of point clouds and VR experiences through the web. Additionally, Omnibase, is already based on Potree, making potential integration easier.

As discussed in the report by Kumar et al. (2019), Potree's reliance on GPU capabilities creates a challenge for users with limited hardware. Potree's limitation to rendering point clouds restricts its utility in cases requiring mesh or 3D object visualization. Potree was initially developed for the HTC Vive, which relies on external base stations (lighthouse sensors) for tracking. These sensors must be set up and calibrated, making the setup difficult and limiting mobility. In contrast, the Meta Quest 3, with its inside-out tracking, uses built-in cameras to eliminate the need for external sensors. This simplifies setup, enhances portability, and improves user comfort. By adopting the Meta Quest 3, the requirement for additional hardware was eliminated, making the experience more user-friendly and immersive compared to the HTC Vive (VRcompare, 2024).

4 Methodology

With all the theoretical aspects covered, the next step is to put everything into practice and develop the functionalities for VR. Chapter 4 provides an overview of the methods and tools used to explore user interactions with point clouds in both VR and Omnibase environments. It covers the setup process, navigation controls, measurement features, and survey structure designed to assess user experience in each setting. Each section explains the features developed to enhance usability and comfort, along with the methods used to gather and analyze feedback on user preferences and challenges within these environments.

4.1 List of Tools and Requirements

The following tools are required to use the developed program:

- **VR Headset and Controllers:** A VR headset with controllers is essential for users to explore and interact with the point cloud in the VR environment. This project relies on the use of the Meta Quest 3.
- **Website:** A website capable of running the Potree program with added functionalities is needed. For development, a localhost server was created using Apache Software (2024), with HTTPS configured to ensure compatibility with Potree.
- **Browser:** A browser supporting WebXR and WebGL is required. Google Chrome was used during development, and the Meta Quest 3's native browser was used for testing, as it provided a more stable experience.
- **High-Performance PC:** When using a connected computer instead of the Meta Quest 3's built-in browser, the PC should meet Meta's performance specifications. For development, a laptop with an NVIDIA GeForce RTX 4060 GPU was used, though this configuration was deemed underpowered by the Meta app.

For full access to the adapted code and documentation, visit our GitHub repository at `https://github.com/MandenB/synthesis-project`.

The following tools are required for conducting surveys:

- **Laptop Setup:** Participants will use a laptop to interact with Omnibase, allowing navigation and measurement tasks with traditional mouse and keyboard controls, which serves as a comparison to the VR experience.
- **Questionnaires:** After each session, participants will complete questionnaires assessing their experience, comfort, ease of navigation, and perceived accuracy in each environment.

4.2 Navigation and Controls

The original VR framework provided basic movement controls, allowing users to start at a randomly selected fixed location within the point cloud and move freely in any direction. Movement was controlled through the joystick of either VR controller, with the camera moving rapidly in the direction the user was looking. The figure below shows an animation demonstrating how navigation through point clouds appears in VR.

Figure 4: Navigation Interface in VR

4.2.1 Eye-Level Movement

To create a more immersive experience, we implemented an eye-level movement feature within the point cloud. Here, the camera's z-value was set to a fixed value by adding 1.8 meters to the z-coordinate of a ground point, simulating the average eye level. However, a limitation occurs when the point cloud ground is uneven, as a single fixed value fails to adjust to height variations. A dynamic solution, which adjusts the camera height based on nearby ground points, will be discussed in Chapter 5.

4.2.2 Upgraded Joystick Controls

To increase accessibility, particularly for users with restricted movement ability, we adapted the joystick functions to incorporate both movement and rotation. One joystick now controls rotation, while the other controls movement, enabling users to turn without in the VR environment without having to physically rotate. The newly added VR controls can be seen in Figure 5.

Rotation: Rotating the joystick left or right rotates the camera counterclockwise or clockwise, respectively.

Movement: Moving the joystick up or down moves the camera in the direction the joystick points, based on the controller's orientation. This allows sidestepping without needing to rotate the camera. In a future version, implementing full two-dimensional joystick control would further improve this feature by allowing lateral movements as well as forward and backward motion.

4.2.3 Movement and Rotation Speed Adjustments

Finally, we modified the speed of movement and rotation when the joystick is fully pressed, providing a more realistic, walking-like experience at lower speeds than the original high-speed movement. This adjustment also improves precision, making it easier for users to control camera rotation and movement accurately.

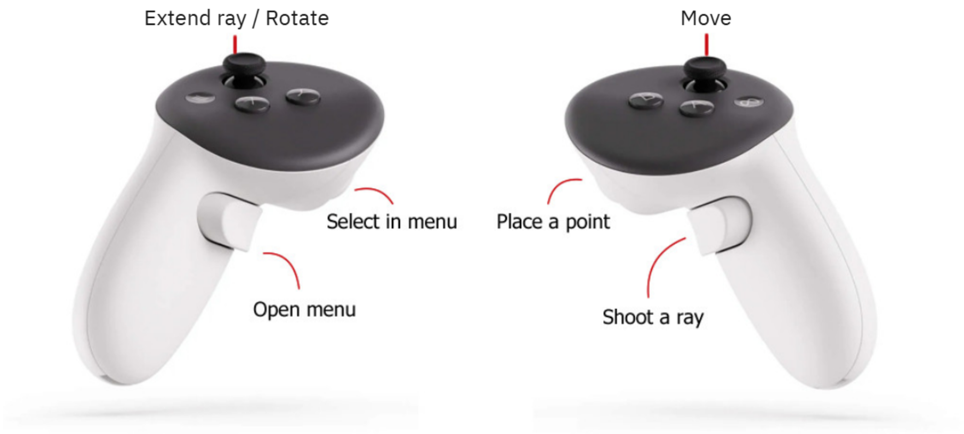


Figure 5: Newly added VR controls (figure adapted from Meta (2024))

4.3 Measurements

The original Potree framework lacks measuring tools in the VR environment, although it does include 2D measurement tools for point clouds. These 2D tools served as the inspiration for the newly implemented measurement features in VR.

4.3.1 Point Selection

The initial design for all measurement tools involved selecting a single point from the point cloud using one of the controllers, with selected points then used for various measurements. However, difficulties with intersecting the point cloud prevented this approach from being completed within the project's timeframe. Details on these issues are discussed in Chapter 5.

In the final approach, users create new points within the point cloud at locations of their choice. To initiate this, the user presses the squeezing trigger on the controller, generating a new vector using the Three.js library. The vector originates at the controller's center, extending in the negative z-direction (forward) relative to the controller's orientation. The joystick on the opposite controller adjusts the vector's length, with forward inputs increasing length and backward inputs decreasing it. The length is calculated by scaling the joystick's output value with an initial length of 2 meters, then dividing by ten to reduce speed of change, and finally inverting the value to make it negative for a forward direction.

After creating this vector, a new sphere geometry (also using Three.js) is positioned at the vector's endpoint. When the user presses the controller's trigger, the sphere is fixed in the scene until the measurements are cleared.

Each point placed in the scene is labeled with its XYZ coordinates. Labels are positioned slightly above each point for visibility and set to always face the camera using a quaternion.

4.3.2 Line Measurements

The first measurement users can perform is calculating the distance between two points. When a second point is placed in the scene, the 3D distance between it and the first point is calculated and displayed. A new vector representing the distance is then rendered between the two points.

After the initial two points, each subsequent point only calculates the distance to the most recently placed point. As with points, a label is created for each distance, positioned slightly above the midpoint of each line for visibility.

4.3.3 Area Measurements

After placing three points, the approximate total area of the resulting 3D shape is automatically calculated. For each of the three coordinate planes (XY, YZ, ZX), the shape is projected onto the plane, and the projected area is calculated by treating the points as vertices of a polygon on that plane. The areas of these projections are then squared, summed, and the square root of the sum is taken, yielding a scalar value that approximates the spread of the shape in 3D space based on its projections. When a fourth point is added, the area calculation is repeated for the new shape, and this process continues with each additional point.

A label is also created for the area, positioned at the shape's center with a slight height offset for visibility.

This method has a few limitations. First, the shape's boundary depends on the order of point placement; if points are not placed in a consistent clockwise or counterclockwise order, the shape's edges may cross, resulting in inaccurate boundaries. An improvement would involve ordering the points systematically to avoid intersecting boundaries regardless of insertion order.

Secondly, this area calculation method provides only an approximate area based on 2D projections, rather than the exact 3D surface area. A more precise approach would be to generate a mesh from the points and calculate the area directly from the mesh. However, due to time constraints, these improvements were not implemented.

4.3.4 Measurement Deletion

The final tool implemented allows the user to delete all measurements created during a session. A small menu can be accessed by pressing a button, which reveals an option to clear all measurements currently displayed in the scene.

The figure below shows an animation demonstrating how measuring work in VR, including point placement and a view of the menu interface.

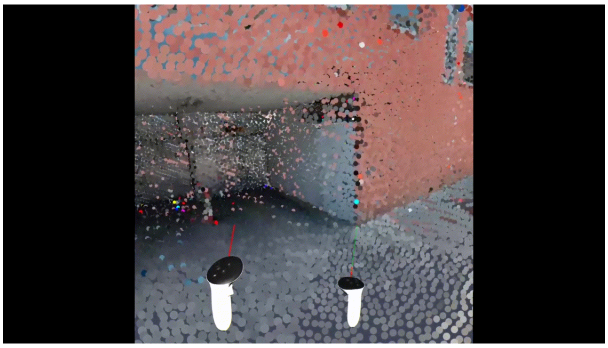


Figure 6: Measuring and Menu Interface in VR

For each measurement tool, created points and measurements are stored in dedicated arrays, which also hold the associated meshes for lines and points. When the delete button in the menu is pressed, each stored element is removed from the scene, and the corresponding arrays are cleared. Currently, this method deletes all measurements simultaneously; potential improvements, such as selective deletion, will be discussed in Chapter 7.

4.4 User Experience Surveys

This survey is structured to compare the impacts of visualization and measurement techniques in point clouds between two distinct environments: **Omnibase's multi-view**, and a **VR environment**. The primary focus is on how each environment affects user perception, interaction, and measurement accuracy. Therefore, aiming to understand these impacts across two distinct user groups: those familiar with point clouds and those unfamiliar. By analyzing the results from this survey, we aim to provide definitive answers to the primary research question of this project, providing insights that could guide future implementations and improvements.

To thoroughly address the research question, the survey incorporates several subquestions aimed at dissecting the user experience into manageable components that will provide a comprehensive understanding of both systems:

- **User Comfort and Immersion:** This subquestion evaluates the physical comfort of users and their subjective immersion experiences when interacting with point clouds in VR versus Omnibase's multi-view. It seeks to identify if the immersive nature of VR adds comfort and deepens the sense of being "present" within a virtual environment compared to a traditional multi-view setup.
- **Ease of Navigation and Control:** The survey assesses how intuitive the navigation and control interfaces are in VR compared to Omnibase. This involves exploring user feedback on the ease of moving through and manipulating the point cloud data within each system.
- **Accuracy of Spatial Measurements:** Focusing on the precision of spatial measurements, this aspect compares the accuracy attainable in VR and Omnibase's

multi-view. It examines the consistency of spatial measurements, particularly looking at depth accuracy and the occurrence of measurement discrepancies between the two platforms. Additionally, it queries whether VR provides an advantage in intuitiveness and accuracy for inexperienced users.

4.4.1 Survey design

Participants engaged with the same point cloud dataset of Vienna in two environments: **Omnibase's multi-view**, and a **VR environment**. They were asked to perform a series of standardized tasks designed to test various aspects of their interaction with point clouds. We first provided how to use the VR controllers and navigate the Omnibase interface on a laptop. Upon entering the VR headset, the participants started at the street view at the same initial location and were first tasked with freely moving to test navigation ease. Subsequently, they performed a precise measurement task by selecting points at the four corners of a doorframe to test the functionalities of point selection and area calculating (see in Figure 7). This process was then repeated using the Omnibase Multiview, allowing for direct comparison of task performance and measurement accuracy in both systems.

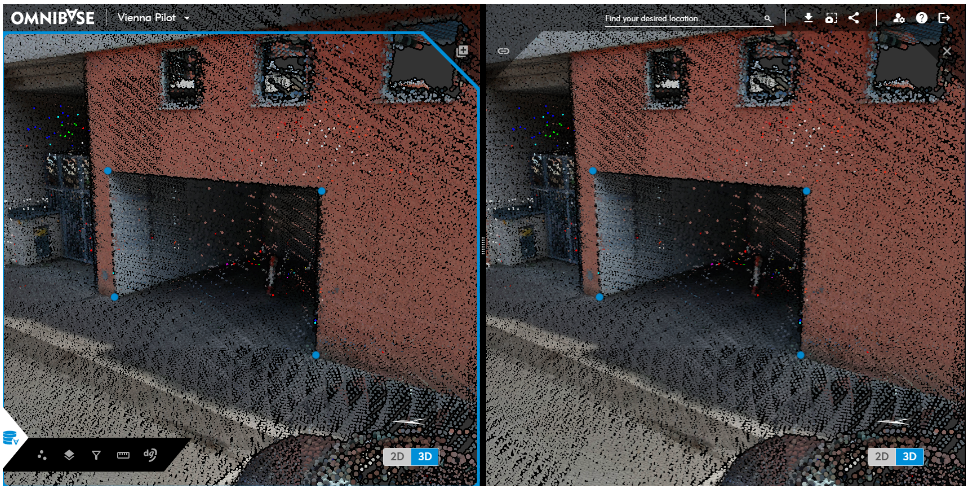


Figure 7: points chosen for measuring the area of the door frame in Omnibase (figure adapted from Geodelta (2024))

Following these tasks, participants completed a structured survey comprising both scale questions and open-ended responses. Although Omnibase offers various functionalities, the survey focused solely on the area measurement task to align with what was implemented in the VR environment. This design choice aims to directly compare the specific strengths and weaknesses of VR and Omnibase's multi-view systems, especially in terms of spatial interaction and user experience.

4.4.2 Participants

The survey will target two main groups to ensure a comprehensive evaluation:

- **Experienced users:** These users are more familiar with point clouds and will assess the technical value of the VR system and its potential applications in their work. This group, consisting of 5 first- and second-year master's students in Geomatics, is familiar with point clouds.

- **Non-experienced users:** This group will provide more insights into the overall usability and accessibility of the VR point cloud experience. This group is comprised of 5 bachelor's and master's students from other field of studies.

Having two groups helps us address the different needs of Omnibase clients, who vary in their experience with point clouds. For this survey, the sample size was chosen to balance statistical value with practical limits, such as time and resources for the project. While larger samples provide stronger statistics, smaller groups can still offer valuable insights, especially when focusing on user experiences rather than purely statistical results (Martínez-Mesa et al., 2014). This approach supports our choice to work with two small groups, prioritizing practical insights over extensive statistical analysis.

4.4.3 Procedure

The study began with a brief introduction to each environment. Both environments provided participants with instructions on how to use the VR, including which triggers to use and how to perform specific tasks. Participants performed identical tasks in both settings, focusing on navigation and spatial measurements. After interacting with each environment, they completed questionnaires and participated in surveys to assess their experience, comparing comfort, ease of use, and measurement accuracy across the two platforms.

Through this data, we can understand how immersive each environment feels to the user, based on comfort and ease of navigation, how accurate the participants' measurements are in VR compared to Omnibase's multi-view interfaces, and how efficiently participants can complete tasks in each environment by analyzing the time taken to perform similar tasks across platforms.

4.4.4 Survey Format and Questions

After the tasks were done in both environments, a Google Form survey was sent to all of the participants. The Google Form can be accessed at this link, and the questions can be found in Appendix A. The survey is comprised of four parts - general questions on experience with point cloud, VR, or Omnibase; Comfort and Interaction; Navigation and Controls; Measurement and Tasks. Within these parts, the data gathered can be categorized into quantitative and qualitative. Using both allows us to capture a well-rounded view of user experiences with VR and Omnibase. Quantitative data provides measurable insights for direct comparison, while qualitative feedback offers deeper understanding and context, highlighting specific areas for improvement.

Quantitative data will be collected through Likert scale ratings from 1 to 5, to assess participants' initial knowledge and confidence in using point clouds, Omnibase, and VR. This provides a baseline measure to interpret subsequent feedback more effectively. Ratings will also cover immersion, comfort, and ease of navigation, allowing for measurable comparisons between Omnibase and VR. Additionally, measurements of a selected area will be recorded in both environments, enabling accuracy comparisons between the two groups of data in two environments.

Qualitative feedback will focus on participants' subjective experiences, exploring immersion and comfort (including any discomfort, such as dizziness) and comparing the perceived pros and cons of each platform. Questions will guide participants to discuss each environment's strengths and weaknesses, particularly in terms of navigation ease, control intuitiveness, and usability. They will also compare point selection and area measurement task completion speed in each environment, indicating which one they felt was faster and easier. While no formal definitions were provided, comfort and immersion were intuitively understood by participants, many of whom described feeling mentally and physically "present" in Vienna, enhancing their engagement with the VR experience.

4.4.5 Ethical Considerations

To mitigate any risks, all data collected will remain anonymous and be securely stored, preventing any personal identification of participants.
Our approach aligns with TU Delft's guidelines on protecting human research participants by considering potential risks and ensuring data security and participant consent.

5 Encountered Issues

Chapter 4 outlined all additions made to Potree's VR environment, which enhance user experience and immersion while providing tools for measurement.
These additions, however, were not the only functionalities explored for implementation. This chapter will discuss other functionalities attempted during development and the issues encountered, which ultimately made these features unsuitable for the final implementation.

5.1 Dynamic height

Point clouds have varying height values in different areas (which is more noticeable in point clouds with a wider area span), especially in the Vienna file used for testing, which presents height differences of up to 7 metres from the highest to the lowest points. Therefore, given the initial height position established for the headset, at some points, the user would either be too high or too low (underground) with respect to the ground floor. Thus, losing part of the "realism" behind the experience.

To make up for this problem, to make it align to make the experience more immersive, the next approach was to implement "dynamic height". A method which would allow users to travel through the point cloud and always stay at 1.8 metres above the ground, matching the constant human view any person has when walking down the street. Therefore, no matter the slope or the height, the camera view would always stay at a certain height above the ground floor, keeping the experience realistic and dynamic.

In order to achieve this, the initial idea was to implement raycasting. This way, the headset could interact with the point cloud and recognize the z-coordinates (height value), or every point below it, to maintain the height constant at all times. However, after many attempts (with the raycaster function in three.js and the group's own attempt to make raycasting), it still could not interact with the point cloud.

Thus, another possibility arose, where raycasting interacts with a mesh of the point cloud. So, using the '.laz' file of Vienna, it was adapted to a '.ply' file, with only x, y and z values, for faster computation and easier handling of the points. Nonetheless, the only requirement for this mesh was the ground points of the point cloud. Therefore, given LAZ files have a classification included ('2' for ground points), the extraction of ground points is easier and faster. While the program correctly selected most of the ground points in the file, some of them were not extracted correctly, as they contained erroneous classification labels.

To solve this, an algorithm was applied, in which from the remaining non-ground points, it would select those with the lowest 18 percent z-value to be included in the ground points file. This percentage selection is due to the fact that at this height, nothing is above the 1.8 meter mark that the headset would rely on, and would not affect any movements. In turn, cars would still be accounted for wall collision (explained in the next subsection) as only their wheels would be included in the ground points (without them becoming possible walking paths as the wall collision from the top of the car would prevent this from happening).

Once the classification process was completed, using another algorithm developed by the team, the meshes were obtained from the point cloud. By using 'trimesh', a python package used to compute triangular meshes through Delaunay triangulation, two meshes were generated for both classified point clouds (ground and non-ground point clouds) in '.glb' format, as it is a more compatible format with WebGL for VR visualization. For this section, only the ground points are needed to compute the dynamic height.

The mesh was imported into the 'VRControls.js' file, so that the headset and the raycasting would interact with it. Importing the mesh into the point cloud was successful and could be visualized in the web environment point cloud. However, some outliers were noticeable when entering the VR scene, so some outlier correction would be needed. This occurred because when the mesh was computed, it would count some sparse points that appeared mid-air, generating areas above the point cloud's ground height (this was solved by fine-tunning parameters in the mesh generation step, however, as the main concept of dynamic height was discarded, this step too was canceled). With the mesh successfully imported into the point cloud, the next step was to make it possible for interaction to occur between the headset and the mesh. Therefore, some methods were accounted for:

- **Raycasting:** By performing a downwards raycasting from the headset into the floor, the ray should hit the mesh and react to the z-value height at that point, stabilizing the user's position at 1.8 metres in the vertical axis. This method failed due to the way Potree deals with meshes, in which it only accounts for the vertices, therefore, it was exactly like the point cloud. Thus, this was still a problem, as some areas would not have an interpolated height value, and the team still could not make it interact with points.

- **KDBush:** The second option was implementing a package called 'KDBush'. A flat KD-tree with zero allocation is used to compute 2-D points in a fast and efficient. This KD-tree is optimized to deal with large trees in order to handle large datasets. Therefore, given the pointcloud contains around 130 million points, it has the ideal assets to handle such en enormous point cloud. The vertices of the mesh were inputted into the KD-tree (which is basically the point cloud) so that the algorithm would traverse the points and select the one the user would point at. Though it managed to work, and return values (meaning it did interact with the point cloud), it is a very demanding process.
  It is so computationally expensive that the scene cannot render properly, thus, offering a very poor experience to the user with non-rendered points and a very low frame rate (which can lead to sickness and dizziness). This occurs because every time the user carried out an action, that was moving, measuring or interacting, the program would revisit the whole KD-tree for each single frame that had to be rendered. Finally, it was decided that this method would not be included. Nonetheless, this option should not be discarded for future considerations, as it is something that can probably be optimized if plenty of time is given to work on it.

5.2 Wall collision

This functionality was one of the main goals at the start of the project. The idea is to make the user stop when moving towards an object that should not be physically possible to go through (such as cars, walls, trees, posts, and so on). This would make the experience more realistic for the user, feeling as if they were actually interacting with the objects in the scene.

The way to achieve this was through raycasting. By shooting a ray from the headset in the forward direction from where the camera is looking at, the ray would interact with the point cloud. The time it would take for the ray to reach a point would determine the distance to which the headset is from that point. If the value was lower than an established threshold (for instance 1 metre), the camera would stop moving forward (in that direction), forcing the user to move in another direction. This way, it would feel as if the user was actually "colliding" with an object, such as a wall or car (however, the user would still be able to move the camera view around, just not advance).

After failing to implement raycasting and its interaction with the point cloud, just like with dynamic height, another solution was attempted. This lead to using the meshes as the second approach for this functionality, as it could be useful for both wall collision and dynamic height (solving two problems at the same time). Therefore, by using the mesh for wall points in a similar procedure as with the ground points mesh, the objective was to make the raycast functionality interact with it. Sadly, the file was not able to load two meshes at the same time. After several attempts at trying to fix this issue with different code variations, it was not possible to integrate two meshes following the simple mesh loading approach from Potree. Thus, with limited time and without the possibility of implementing the wall mesh into the point cloud, we had to come up with another solution.

Eventually, it was decided that the ground mesh could be used as a barrier for the user (as it was the only mesh that would successfully load into the point cloud), which would limit the movement. The headset would be able to traverse through cars, posts and trees, but upon reaching the x and y coordinate limits of the ground mesh, it would not be possible for it to move outside those bounds. Therefore, the user would never "walk through" any wall but could get excessively close. This variation was a simpler wall collision solution the team wanted to implement, to at least have some real sensation that the user cannot physically enter a building through walls. Cars and trees were excluded as it felt that it might not be that determinant in providing an immersive approach (yet, if the possibility was there, it would always be better to make the user "bump" into these objects). Unfortunately, just as with dynamic height, the same issues arose, where the mesh would not interact with the headset.

5.3 Point Selection

As described in Chapter 4, measurements are currently made by creating new temporary points, which are then used to calculate distances and areas. Originally, the plan was to use existing points from the point cloud for these measurements, similar to the approach used in Omnibase.

The intended concept was similar to the implemented method, with the user initiating the selection process by pressing a button. However, instead of placing a point at the end of a vector from the controller, a raycaster would check for intersections with points in the point cloud. The intersected point would then be highlighted, providing visual feedback on the controller's target. As in the current implementation, the user would press the trigger to lock the point for measurement.

To create the raycaster, we followed the steps described in Section 4.3.1: the controller's origin and rotation were used to create a ray extending infinitely in the forward direction. This ray would then be used to check for intersections; however, this step presented several issues. The most significant issues encountered were:

- Multiple coordinate systems
- Incompatibility with point clouds

5.3.1 Coordinate Systems

Potree's VR environment involves two different coordinate systems: the VR or world coordinate system and the point cloud or scene coordinate system. Switching between these two systems requires applying the camera's world matrix to the vector that needs transformation.

This conversion is necessary because the ray from the raycaster is defined in the VR coordinate system, where the VR controllers are positioned. However, to intersect with the point cloud, the ray must be translated into the scene coordinate system. Therefore, before creating the ray, the ray's origin (the controller's center) and direction were converted to the scene coordinate system. When visualized, this translated ray correctly showed the intended position and direction.

However, when we applied Potree's intersection function, the intersections appeared where the user (camera) was looking rather than where the controller was pointing. The cause of this issue remains unclear, though it is suspected that Potree's intersection method was primarily designed for 2D views, which may not account for VR-specific positioning.

5.3.2 Incompatibility

To address the problem of the raycaster mistakenly referencing the camera, we tried using Three.js's intersection function as an alternative. However, this approach also introduced issues. When testing, we found that the intersected points consistently appeared offset from the actual ray's intersection with the point cloud, which was visually verified by rendering the ray. Initially, offsetting the selected point seemed like a potential solution, but it proved ineffective. For example, when selecting points near the edge of a wall, the ray was already adjacent to the wall, but the offset prevented accurate selection.

Further investigation revealed that Three.js's intersection function is optimized for solid meshes, which is likely the reason for this behavior, as it does not interact effectively with point clouds.

6 Results

The results of the team's work are presented in this section. It will cover both the qualitative and the quantitative results obtained, where it will describe the measurements made by those who were interviewed, as well as their opinions on the comparison between 2-D Omnibase and the VR scene.

6.1 Quantitative Results

After asking 10 different users to carry out the surveys and make them take some measurements on the area of the doorframe, the results between Omnibase and VR could be compared.

Table **??** presents all the measurements taken by all 10 users. It contains the area of the door frame obtained both in Omnibase and in VR. Due to privacy reasons, the names of the participants have been replaced with numbers.

Group	Participant Number	Measurement VR	Measurement Omnibase
Experienced Users	1	13.92 m^2	15.90 m^2
	2	12.84 m^2	15.80 m^2
	3	14.89 m^2	17.56 m^2
	4	15.12 m^2	16.75 m^2
	5	14.01 m^2	16.02 m^2
Inexperienced Users	6	18.91 m^2	15.81 m^2
	7	16.54 m^2	16.89 m^2
	8	13.16 m^2	16.11 m^2
	9	13.54 m^2	15.738 m^2
	10	14.36 m^2	15.69 m^2

Table 1: Division of Participants into Geomatics and Non-Geomatics Students and their results to the measurement of the doorframe in VR and Omnibase.

From these results we can perform different metrics to further enhance the analysis of the survey measurements the students obtained. Therefore, we computed the average or mean, the standard deviation and the highest and lowest values for both Omnibase and VR results. The metrics obtained are as follows for point cloud users:

Metric	Experienced	Inexperienced
Average (Mean) VR	14.61	14.82
Average (Mean) Omnibase	16.06	15.80
Standard Deviation VR	1.79	1.65
Standard Deviation Omnibase	0.62	0.51
Highest Value VR	18.91	16.54
Lowest Value VR	12.84	13.16
Highest Value Omnibase	17.56	16.89
Lowest Value Omnibase	15.693	15.738

Table 2: Metrics for Point Cloud Users and Inexperienced Users

6.2 Qualitative Results

The feedback obtained from the surveys on the experience was quite positive for the most part. The following subsection contains the feedback received from the interviewees during the surveys, regarding some of the answers received in the questionnaires handed to the participants. It will explain both the positive and the negative experiences the users had with both VR and Omnibase, as well as some final improvement proposals made by some of them.

One of the most repeated positive points by users regarding VR is the depth sense they perceived while moving around the Vienna point cloud. All of them agreed that it not only felt more immersive, it also felt more natural. Most of the users mentioned the fact that moving around so freely was also a very positive and enjoyable aspect of VR, as they described it as being "similar to walking in the street". How dense the point cloud is in VR also affects this part of the experience, as one of them mentioned how a sparser point cloud would not have had the same effect in giving that realistic approach to the scene.

Regarding user movement, the majority of the surveys provide positive feedback on both VR and Omnibase. They are both very different yet provide a straightforward and easy-going way of moving around the point cloud, therefore users were very content with how this feature was developed in both cases. Nevertheless, some students indicated, in favor of VR, that rotation and panning felt more intuitive in this system than in Omnibase.

When taking measurements, even though the point positioning is through a ray, the users still felt like it was a very effective way of generating points. As one of the interviewees also pinpointed, it was the fact that the ray length can be modified that adds a very useful aspect for the measurement tool, as it prevents the user from needing to move around when creating an object inside the VR. However, this time, Omnibase was the preferred system in which to generate a point or make an area in a faster and seamless way. All of the users agreed that Omnibase was quicker, as it did not require "complex" controls and everyone is used to the methodology of creating areas in 2D space. In addition, they mentioned that the multi-view option was a complimentary option that aided in obtaining better results.

Moreover, non-experienced students mentioned that they did not feel like they needed any previous experience for these measurements to be carried out (in either point clouds, VR or Omnibase), indicating that the experience is user-friendly and that anyone using any of the softwares can carry out the measurements without point cloud knowledge limitations.

Despite all the positive comments received, there were still some mixed opinions on certain aspects of the VR scene. For instance, questionnaire results displayed very varied opinions on how comfortable viewing in VR is. This is because dizziness played an important factor in this matter. While some people felt more used to it, others struggled with keeping up with the lower frame rates. The same goes for point generation, as some deemed it quite impressive while others considered it to react with a certain delay.

In turn, the project also received some very important feedback on the negative aspects they encountered. All of the interviewees offered some constructive approaches to functionalities that can be improved and challenging moments they encountered during the survey.

Regarding Virtual Reality, the most mentioned issue by the majority of the users was the occasional dizziness produced by the low frame rate that was displayed at times. This occurred due to delayed rendering time of the scene or while moving too fast in the pointcloud after a certain period of prolonged use. In addition to this, they mentioned that occasionally, movement controls like rotation would increase this effect, as sometimes frames would have a hard time keeping up with rotation speed.

Another frequently mentioned issue was the fact that the controllers would swap their button assignation. When swapping the headset from one to another, the gyroscope in the headset would detect this, closing the session and initializing it again once it detected a new user wearing the headset. During this process, the controllers would be inverted and where the left one usually containing the rotation and the menu, would now swap to the right controller. Of course, this is a problem that limits how enjoyable the experience can be, as it delays significantly the adaptability to the controls and how intuitive it is to carry out a measurement. Other bugs like the menu not closing or finding the ray length hard to adjust would also prompt the interviewee to provide negative feedback.

Omnibase also received some slight constructive comments in which some users mentioned that some controls like dragging, moving or zooming in and out did not feel as optimized as they should be. These controls were described as being inaccurate and not as intuitive as it would have been expected.

However, all of the issues expressed by the interviewees can be fixed if more time is dedicated to debugging and fixing these challenges. The overall feedback received indicates that users are generally very pleased with how the VR environment works and that making the measurements is more understandable and enjoyable in VR. It helps them understand better the environment, given the depth it provides makes it more natural. Yet, it does require a longer learning curve than that of Omnibase, as it does need a tutorial and might take some time for several users to get used to (around 10-15 minutes for most of them). This issue arose frequently, in which some people with prior controller experience (for instance, video games) would have less trouble learning the controls in comparison with other non-experienced users who took longer to adapt to the VR's mechanisms. In addition, the general survey feedback indicated that it is still harder to make measurements, for instance, an area, and that it did not feel that accurate. The same was mentioned for navigation, as they deemed it to be faster in Omnibase. However, most of them agreed that with slightly more practice, most of these comments can be tackled.

Most of the participants left some interesting suggestions on improvements that can be made to the VR environment. These seem to pinpoint what most users would be looking for when optimizing the VR feature in the near future if the project carries on.

The great majority of users pinpointed the idea of including a snap feature, in which the ray would detect the points in the point cloud and adhere to it. Therefore, easing the way in which point selection is made while at the same time, making the measurement more accurate. Moreover, some of them suggested it would be a great addition to have a storage option, in which after obtaining a measurement, the user would be able to store it and keep generating new ones (something very limited at this stage of development). One last general comment involving point generation revolves around deleting points. Some users found it important that the deleting points option can allow for the deletion of only one point, instead of all the current ones generated.

Other very common constructive criticisms received involved the idea of designing a user guide, in which before entering the actual point cloud in the VR scene, users can learn the controls in detail. Although this employs some development time and further research on what is the best way to achieve this approach, it is an important consideration to have for future work. Lastly, keeping up with user controls, some suggested that the rotation and translation functions should be included in the same joystick, facilitating the way the user moves around the point cloud, in a much more smooth and intuitive way.

Therefore, at the moment, Omnibase seems to be the most economical and user-friendly option for fast computation with measurement tools. Nonetheless, the general idea was that the positive outweighs the negative for VR measuring, indicating that users were very reluctant to see any new optimizations made in this new approach for measuring areas, lines or points inside a point cloud as well as any other possible functionalities included in the future.

7 Discussion

In this chapter, the results will be put into a larger context and analyzed, while addressing the limitations of this study. Additionally, potential future research topics and improvements are discussed.

7.1 Interpretation of Results

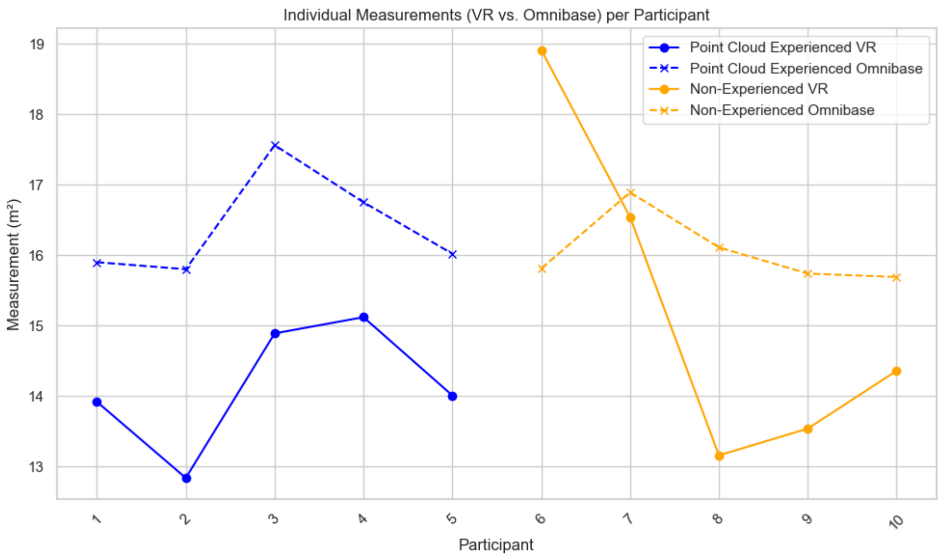


Figure 8: Individual measurements per participant

In Figure 8, several patterns highlight how experience and platform design influence measurement consistency between VR and Omnibase environments:

Experienced Users: For experienced users, measurements in both VR and Omnibase show closer results, with an Omnibase mean of 15.80 and VR mean of 14.82. This similarity suggests that experience with point cloud data helps users adapt to either environment effectively. However, Omnibase's lower standard deviation (0.62) for experienced users implies that those familiar with similar software might achieve more consistent results, likely because they're accustomed to Omnibase-like 2D interfaces where point selection is stable and familiar.

Inexperienced Users: The inexperienced group displays greater variability in VR, with a standard deviation of 1.65 compared to 0.51 in Omnibase. This difference points to a steeper learning curve in VR, where point selection using spatial raycasting can feel less predictable. For these users, Omnibase likely offers a more straightforward setup, with fewer fluctuations in measurement placement and better visual control, resulting in more stable outcomes.

Impact of Familiar Interfaces: For both groups, Omnibase's interface seems to offer a consistency advantage, with lower standard deviations overall. This advantage could stem from Omnibase's familiar structure, resembling software experienced users typically interact with, helping both groups adapt more quickly to the measurement process.

These findings suggest that VR may require additional adjustments or user guidance to achieve Omnibase's consistency, particularly for those new to point cloud interaction. However, with further refinement, VR could potentially match Omnibase's stability for accurate measurements across experience levels.

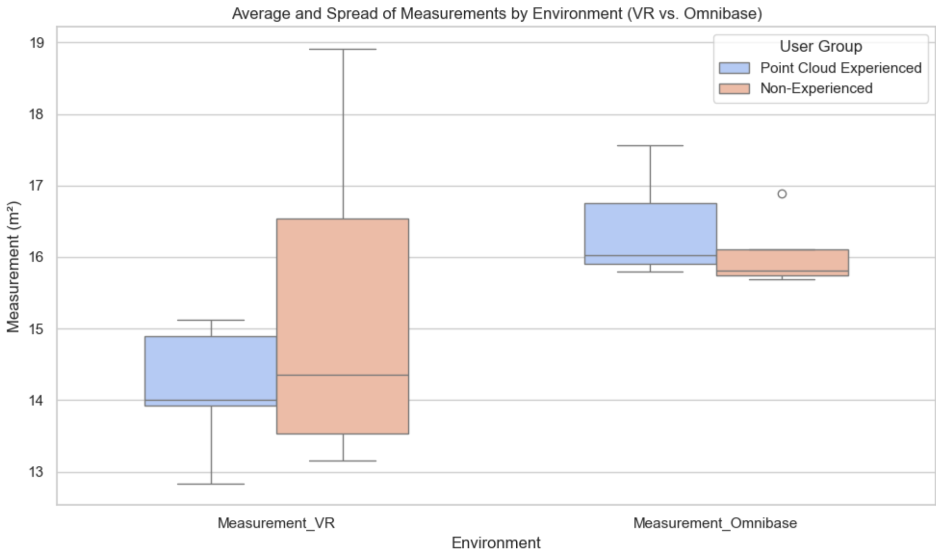


Figure 9: Average and spread of measurements per environment

The average and spread of measurements across VR and Omnibase in Figure 9 show differences based on user experience.

Non-Experienced Users: Non-experienced users have a larger spread in VR measurements, likely due to challenges with VR controls and point cloud navigation. In contrast, their measurements in Omnibase were more consistent, possibly due to its similarity to conventional software.

Experienced Users: Experienced users displayed similar measurement spreads across both VR and Omnibase, suggesting that their familiarity with point clouds allows for stable performance in either environment.

These results indicate that experience with point cloud data significantly impacts measurement consistency, particularly in VR. Non-experienced users may need additional support for consistency in VR, while experienced users perform reliably in both environments.

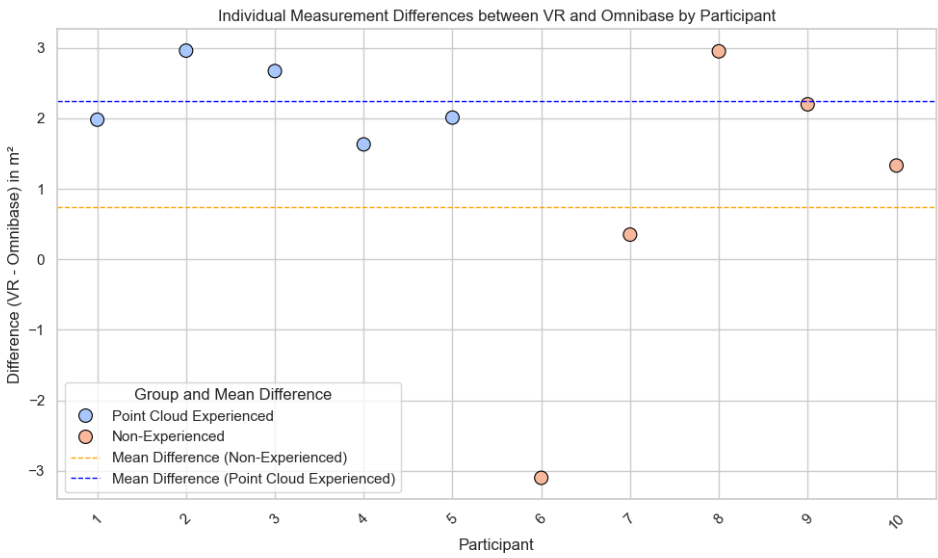


Figure 10: Difference in measurement per participant

Figure 10 shows the differences in measurements between VR and Omnibase for each participant. A positive result in the graph indicates that the measurement in Omnibase is bigger. The consistent positive difference indicates that the users were measuring a larger area in Omnibase. Due to a lack of a ground truth, we cannot for sure make the assumption that this area is closer to the "true value".

However during the interviews it was observed that through Omnibase software, users generally tend to obtain higher area values. This happened frequently due to the fact that the user did not fully understand where they were placing the vertices conforming to the edges of the rectangle. Therefore, some of the points would appear in front or behind where they intentionally intended to place the point. On the contrary, in VR, the users would obtain reduced values for the area, this was because some of them had trouble setting the length of the ray correctly, and ended up placing points closer to the headset camera than they expected.

7.2 Limitations

This research has several limitations that affect its generalizability and constrain the scope of conclusions that can be drawn. These should be carefully considered in future studies. The limitations are discussed below.

7.2.1 Methodological Limitations

The following limitations concern the research approach and flaws in the methodology that have an impact on the results. Firstly, due to a small sample size of survey participants, the findings of this study may not be generalizable to a broader population. The limited number of surveys reduces the ability to draw statistically significant conclusions.

Additionally these participants have a diverse selection of backgrounds. Some participants have prior experience with VR or gaming, making the VR platform easier to navigate than those without, potentially giving the VR a bias in the results. This could be the reason that the experienced users had more consistent measurements overall, due to more general experience with computers and controls.
Another possible bias comes from the VR platform being relatively new and "cool". This novelty bias could affect the feedback, as participants may report a more positive experience due to the uniqueness of VR.

This research is also limited to a short-term evaluation. Therefore no conclusions can be drawn about prolonged use of either platform. Extended use of VR may impact the amount and severity of physical symptoms, which could also impact the preference of the platform.

7.2.2 User Experience Limitations

These limitations relate to factors impacting participant comfort, ease of use, and overall interaction experience in both Omnibase and VR environments. Starting with controls, both platforms have specific controls that have to be learned. The results show that the controls of the VR were confusing to some participants, which then could also impact their comfort and ease in measuring. VR likely has a larger learning curve. This also relates to the background of the participants and the lack of long-term data.

Another impact on the user experience feedback is the physical discomfort in VR. This includes fatigue and motion sickness, as described in section 3.3.3. Which could ensure a negative bias towards VR. However, since these physical symptoms are inherently present in the VR experience, these should be considered in the comparison as subjective feedback rather than bias.

The comfort of the user could also be affected by the sparsity of the point cloud data. This study is limited to a Mobile Laser Scanning (MLS) point cloud, which has more points on the facades of buildings compared to an airborne scanned point cloud, making the street view likely more comfortable.

7.2.3 Accuracy and Data Reliability

These limitations affect the study's ability to establish objective accuracy and reliability in measurements.

During the survey, participants were asked to measure the corners of a building. It should be noted the accuracy of these measurements cannot be truly tested. Firstly, because of a lack of a ground truth in either application. And secondly, because this ground truth cannot be determined. As discussed by Zevenbergen and Bennett 2015, boundaries of the real world can only be approximated, and the concept of accuracy in these situations leads to idealization precision.

A design decision was made about the measuring, namely to utilize point addition rather than selection, as used by Omnibase. This allows users to better guess the boundaries of their chosen object, without restraint to the collected data. However, this also limits the "accuracy" to the real-life situation. By relying on user estimations rather than directly using the data, we are moving further from real-life measurements.

Furthermore, point clouds in any state are mathematical representations of points, appearing larger than the singular coordinate they represent. Making all of the measurements an estimation, regardless of platform.

7.2.4 Known Technical Issues

There are several known bugs that could also negatively impact the VR experience. Since dynamic height adjustment was not achieved (see chapter 5), there is a flaw in the current implementation which could cause the user to clip through the floor in areas with increased elevation, or float higher above the ground than the preferred eye level. The clipping through the point cloud is also true for walls, as wall collision was also not achieved.

Another known bug is the occasionally switching controller functionality. As the user exists and re-enters the VR. The mapped controller buttons can be switched with the other controller, depending on which controller is assigned as the primary controller by WebXR.

7.3 Future Work

Future work could explore several areas to improve both the user experience and technical efficiency of point cloud interaction in virtual reality.

Long-term User Preferences: Future studies can examine how user preferences for VR controls may evolve over time. As users become more familiar with the VR controls, they might prefer it over the mouse and keyboard controls.

Physical Symptoms: Minimizing the physical symptoms of VR is an important research area to make VR a more viable option for extended use.

User Comfort with Different Point Cloud Densities: Another interesting area to explore is how comfortable users feel navigating point clouds with varying levels of detail, especially when comparing data from airborne laser scanning (ALS) to mobile laser scanning (MLS) used in this study. By understanding how users respond to sparse versus dense point clouds, various other applications might also benefit from VR where less dense point clouds are collected and required.

Indoor to Outdoor Transitions: Future work can also focus on enhancing VR experiences that transition between indoor and outdoor environments, especially for applications like virtual monument exploration. Research on lighting adjustments, spatial audio, and visual orientation could help create smooth transitions, adding realism and immersion to the experience.

Augmented Reality (AR) for On-Site Measurements: Augmented reality offers another promising direction for future work, allowing real-time measurements directly within the user's physical environment. Through AR, users could interact with scaled point clouds and view projected dimensions on-site, which may be particularly useful in fields such as construction and architecture. This approach has the potential to make measurement tasks more intuitive and accurate.

Automatic Segmentation and Dynamic Detection: Finally, future work can explore automatic segmentation methods to identify features like ground planes, walls, obstacles, and even vehicles—whether moving or parked. This segmentation could enable automatic wall collision detection, and dynamic height adjustment based on the ground points, enhancing user experience and realism in VR simulations.

Together, these areas of future research offer the possibility for making VR (and AR) applications more practical, accessible, and immersive.

8 Conclusion

The aim of this study was to explore how Virtual Reality affects the user perception, interaction, and measurement accuracy compared to Omnibase, especially for users either familiar or unfamiliar with point cloud data. By analyzing user experiences and the measurement data, the platforms were assessed.

Qualitative feedback showed VR's strength in immersion and depth perception. Users noted that VR felt more natural and engaging. The dense point cloud contributed to this feeling, making research into less sparse and airborne point clouds a useful future research topic.

In terms of controls, the familiarity of Omnibase was preferred. It was noted by both user groups that the 2D interface was easier and faster for making measurements. Many users, especially those less familiar with VR, found the controls initially challenging. VR's controls require a longer learning curve and occasionally cause user errors. This suggests that the controls require more time to fine-tune. As well as how to convey the controls to the user. Both of these adjustments could decrease the learning curve, potentially making VR easier to use in measuring tasks.

The quantitative data showed that Omnibase provided greater measurement consistency across both experienced and inexperienced users. However, this consistency did not necessarily indicate higher accuracy; rather, Omnibase measurements tended to overestimate the area of the doorframe. This was in part caused by depth misinterpretations, leading to consistent overestimations. However, while VR measurements showed slightly higher variability, especially among inexperienced users, VR helped users interpret depth more accurately, according to the qualitative answers. This suggests that VR could potentially offer improved spatial accuracy with additional user training, despite requiring a longer learning curve than Omnibase.

In terms of user comfort, VR came with physical drawbacks, such as motion sickness and fatigue, which were not an issue with Omnibase's desktop setup. The lack of long-term user data suggests that further research is needed to understand how extended use of VR impacts user comfort and platform preference. In Omnibase, the learning curve or control difficulties can still affect user experience, but physical discomfort is generally absent. Therefore, VR's physical effects should be considered integral aspects of the VR experience itself. In practical applications, these factors may lead users to favor traditional 2D interfaces for extended tasks, particularly if physical discomfort is frequent or unavoidable in VR.

Overall, the inexperienced users did find the VR to be more natural, immersive and useful in providing depth cues. But, the controls need time to adapt to. This caused variability in measurements and a general preference towards the more accessible Omnibase controls.

In summary, while VR has strong potential for point cloud analysis due to its immersive qualities, Omnibase currently offers a more reliable and intuitive experience, particularly for inexperienced users. VR may require further addressing of the learning curve and minimizing the possible physical symptoms to match Omnibase's ease of use and accuracy in point cloud measurements. Future research could address these limitations by exploring long-term preferences and minimizing the physical strain of VR.

References

Apache Software. (2024). *Apache tomcat.* https://tomcat.apache.org/. (Accessed: 2024-09-26)

Bowman, D., Kruijff, E., LaViola, J., & Poupyrev, I. (2021). A survey of 3d object selection techniques for virtual environments. *IEEE Computer Graphics and Applications*, *22*(6), 24-37.

Burwell, C. L. (2016). *The effect of 2d vs. 3d visualisation on lidar point cloud analysis tasks.* https://figshare.le.ac.uk/articles/The_effect_of_2D_vs_3D_visualisation_on_lidar_point_cloud_analysis_tasks/10151543/files/18295010.pdf. (Accessed: 2024-10-28)

Casado-Coscolla, A., Sanchez-Belenguer, C., Wolfart, E., & Sequeira, V. (2023). Rendering massive indoor point clouds in virtual reality. *Virtual Reality*, *27*, 1859–1874. doi: 10.1007/s10055-023-00766-3

de Haan, G. (2009). *Techniques and architectures for 3d interaction* (Unpublished doctoral dissertation). Technische Universiteit Delft.

Franzluebbers, A., Li, C., Paterson, A., & Johnsen, K. (2022). Vr point cloud annotation for agricultural data. *ACM SUI*. doi: 10.1145/3565970.3567696

Gao, L., Bai, H., Lee, G., & Billinghurst, M. (2016). An oriented point-cloud view for mr remote collaboration. In *Sa '16 symposium on mobile graphics and interactive applications.* ACM. Retrieved from http://dx.doi.org/10.1145/2999508.2999531 doi: 10.1145/2999508.2999531

Garrido, D., Rodrigues, R., Sousa, A., Jacob, J., & Silva, D. C. (2021). Point cloud interaction and manipulation in virtual reality. *Proceedings of the ACM*. doi: 10.1145/3480433.3480437

Geodelta. (2024). *Omnibase software.* Retrieved from https://www.geodelta.com/en/software/omnibase (Accessed: 2024-10-25)

Ipsita, A., Li, H., Duan, R., Cao, Y., Chidambaram, S., Liu, M., & Ramani, K. (2021). VRFromX: From Scanned Reality to Interactive Virtual Experience with Human-in-the-Loop. In *Chi conference on human factors in computing systems extended abstracts (chi '21 extended abstracts)* (pp. 1–7). Yokohama, Japan: ACM. Retrieved from https://doi.org/10.1145/3411763.3451747 doi: 10.1145/3411763.3451747

Kumar, Y. A., Noufia, M. A., Shahira, K. A., & Ramiya, A. M. (2019). Building information modelling of a multi storey building using terrestrial laser scanner and visualisation using potree: An open source point cloud renderer. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLII-2/W17*, 421-426. (6th International Workshop LowCost 3D – Sensors, Algorithms, Applications, Strasbourg, France) doi: 10.5194/isprs-archives-XLII-2-W17-421-2019

Martínez-Mesa, J., González-Chica, D. A., Bastos, J. L., Bonamigo, R. R., & Duquia, R. P. (2014). Sample size: how many participants do i need in my research? *Anais Brasileiros de Dermatologia*, *89*(4), 609–615. doi: 10.1590/abd1806-4841.20143705

Meijers, M., & Verbree, E. (2018, April). *Direct and explorative use of point clouds.* Lecture Presentation. East Quad Lecture Theatre 226 / Geography Lab 231. (Presented in Glasgow, Scotland)

Meta. (2024). *Meta quest touch plus-controller.* Retrieved from `https://www.meta.com/nl/quest/accessories/quest-touch-plus-controller/`

Nebiker, S., Bleisch, S., & Christen, M. (2010). Rich point clouds in virtual globes – a new paradigm in city modeling? *Computers, Environment and Urban Systems*, *34*(6), 508-517. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0198971510000414` (GeoVisualization and the Digital City) doi: https://doi.org/10.1016/j.compenvurbsys.2010.05.002

Papson, S., Oagaro, J., Polikar, R., Chen, J. C., Schmalzel, J. L., & Mandayam, S. (2004). A virtual reality environment for multi-sensor data integration. *Sensors for Industry Conference Proceedings*. doi: 10.1109/SFICON.2004.1287073

Schütz, M. (2016). *Potree: Rendering large point clouds in web browsers* (Unpublished doctoral dissertation). TU Wien, Austria.

Silva, D. C., & Sousa, A. (2020). Rendering and interaction with large point clouds in unity. *ScienceDirect*. doi: 10.1016/j.compgraph.2020.01.006

Technologies, U. (2024). *Unity real-time development platform.* Retrieved from `https://unity.com/` (Accessed: 2024-10-25)

van Oosterom, P., van Oosterom, S., Liu, H., Thompson, R., Meijers, M., & Verbree, E. (2022). Organizing and visualizing point clouds with continuous levels of detail. *ISPRS Journal of Photogrammetry and Remote Sensing*, *194*, 119–131. doi: 10.1016/j.isprsjprs.2022.10.004

VRcompare. (2024). *Comparison: Htc vive vs. meta quest 3.* Retrieved from `https://vr-compare.com/compare?h1=0dLRrbMbHgL&h2=0q3goALzg` (Accessed: 2024-10-25)

Wang, Z., Nguyen, C., Asente, P., & Dorsey, J. (2023). Pointshopar: Supporting environmental design prototyping using point cloud in augmented reality. In *Proceedings of the 2023 chi conference on human factors in computing systems (chi '23)* (p. 15 pages). Hamburg, Germany: ACM. doi: 10.1145/3544548.3580776

Zevenbergen, J. A., & Bennett, R. M. (2015, November). *The visible boundary: more than just a line between coordinates.* Kigali, Rwanda: University of Rwanda.

A Survey Questionnaire

Introductory Questions

- **What is your name?**

  ____

- **What is your field of study?**

  ____

- **How familiar are you with spatial data and point clouds in particular?**
  1 2 3 4 5
  Never used them Very familiar

- **How much experience do you have with 3D visualization tools (like Omnibase or others)?**
  1 2 3 4 5
  No experience Very experienced

- **How familiar are you with Virtual Reality (VR) for any tasks or applications?**
  1 2 3 4 5
  No experience Very experienced

- **Are you familiar with any of the following platforms: Omnibase's 2D system, its multi-view system, or any other VR platform for spatial data?**
  ☐ Yes ☐ No

Section 2: Comfort and Interaction

- **How comfortable did you feel viewing and interacting with the dataset in VR?**
  1 2 3 4 5
  Very uncomfortable Very comfortable

- **Did you experience any discomfort while using VR (e.g., dizziness, motion sickness, fatigue)?**
  ☐ Yes ☐ No

- **How did you find interacting with point clouds in VR in comparison to using Omnibase system?**
  1 2 3 4 5
  A lot more difficult A lot easier

- **Did using VR improve your sense of immersion or understanding of the dataset compared to using a 2D screen?**
  1 2 3 4 5
  No improvement Significant improvement

- **Did you encounter any challenges while interacting with Omnibase or the VR environment?**
  ☐ Yes ☐ No

- **If yes, what challenges did you encounter?**

  ____

- **What improvements would make VR more comfortable or user-friendly for viewing and interacting with datasets?**

  ____

Navigation and Controls

- **How easy was it to navigate and control Omnibase system using a mouse and keyboard?**
  1 2 3 4 5
  Difficult Easy

- **How easy was it to navigate and control in VR using hand-tracking or controllers?**
  1 2 3 4 5
  Difficult Easy

- **Do you feel that navigating to specific points in VR is faster or easier than using a mouse in Omnibase?**
  ☐ Yes ☐ No ☐ Other: ________________________

- **How intuitive was it to pan or rotate the view in VR compared to Omnibase's 2D or 3D systems?**
  1 2 3 4 5
  Not intuitive Very intuitive

- **What would you improve in terms of navigation or controls in either VR or Omnibase?**

  ____

Measurement and Tasks

- **How intuitive did you find the process of adding a new point to the point cloud using VR's raycast function from the controller?**
  1 2 3 4 5
  Not intuitive Very intuitive

- **How easy was it to add points in VR for measuring distances and areas between/within them compared to Omnibase?**
  1 2 3 4 5
  More Difficult More Easy

- **How accurately do you feel you could add points for measuring distances or areas in VR compared to Omnibase?**
  1 2 3 4 5
  Not accurate Very accurate

- **Which system (VR or Omnibase's 2D/multiview) felt quicker for performing measurements?**
  ☐ Omnibase ☐ VR

- **What other buttons would you like to be included in the menu?**

  ____

- **What suggestion would you give to improve the measurement function in VR?**

  ____

B Organisation

B.1 Team members

Name	**Background**
Lotte de Niet	BSc Architecture, Urbanism and Building sciences at TU Delft.
Zhuoyue Wang	BSc sciences (earth sciences track) at Amsterdam University College.
Javier Martínez	BSc Geology at Universidad Complutense de Madrid
Bart Manden	BSc Architecture, Urbanism and Building sciences at TU Delft.
Michalis Michalas	Intergrated Master in Rural, Surveying and Geoinformatics Engineering at National Technical University of Athens

B.2 Supervisory team

Name	**Affiliation / Background**
Geomatics:	
Edward Verbree	TU Delft, Expertise in Positioning, Location Awareness, Point Clouds
Martijn Meijers	TU Delft, Expertise in GIS, Geo-information modeling, Map generalization, Geo-database systems, Cartography, and Geo-visualization.
Hans Hoogenboom	TU Delft, Department of Architectural Engineering Technology
Geodelta:	
Martin Kodde	Geodetic Consultant and Director at Geodelta
Fabian Visser	Software Engineer at Geodelta, developing Omnibase
Roeland Boeters	Geodetic Software engineer at Geodelta - Developer of Omnibase

B.3 Responsibilities of team members

During this project, each team member will take on a specific role. While every Work Package will be a collective effort, the roles serve as a way to ensure responsibility. The focus on a certain area of the project will also ensure that the tasks can be efficiently distributed between the team members.

Role	Description
Project Manager and Technical Oversight (PM) Lotte de Niet	Oversees the grand scheme of the project, by ensuring the team keeps within the assigned timeline. As Technical Oversight, they provide support for the Technical Lead and Integration Engineer to make sure that the different parts of the project integrate seamlessly.
Technical Lead (TL) Michael Michalas	Acts as the technical guide for the team, helping with key decisions regarding architecture, tools, and ensuring the technical quality of the project. They also manage the interviews together with the Communication Manager.
Integration Engineer (IE) Bart Manden	Manages the integration of the point cloud data and tools into the VR environment. Together with the Technical Lead, they ensure that the code is of good quality.
Communication Manager and UX Designer (CM) Zhuoyue Wang	Manages internal and external communication, ensuring team members stay informed and stakeholders are kept in the loop. As the UX Designer, they make sure that the VR experience is intuitive and immersive.
Quality Manager (QM) Javier Martinez	Ensures the VR environment functions as expected, identifying and addressing bugs and edge cases. Responsible for the quality of the end products, such as the code, presentations, and report.

B.4 Client

The client of this project is Ingenieursbureau Geodelta B.V., a Delft-based engineering firm specializing in geodesy, photogrammetry, and laser scanning, ensuring that all spatial measurement needs are met with the highest standards of accuracy and reliability. They achieve this by developing user-friendly software and providing trusted, expert advice. One of their key software solutions is Omnibase, which will be the focus of this project. Omnibase allows users to work with point clouds, panoramic images, nadir and oblique images—all in one 3D cloud environment—making the management of large-scale maps effortless. Municipalities, provinces, and other official bodies use Omnibase to maintain up-to-date geo-information, combining modern data sources such as point clouds and panoramic imagery with ease. Built with deep expertise in laser scanning and photogrammetry, Omnibase ensures geodetically accurate processing and seamless integration with existing large-scale maps.

We expect the client to:

- Provide sample point cloud datasets currently in use to test the VR integration.
- Provide feedback on our implementations, offering guidance on potential optimizations for performance, usability, and accuracy. Additionally, assist in identifying any inconsistencies or challenges in the integration to ensure smooth functionality.
- If we were to integrate our work into Omnibase: Support us with understanding the infrastructure details to ensure smooth integration of the VR system.

B.5 Project Meetings

- Group meetings: at least two times per week, depending on the changing schedules and the workload of the team members. The plan for now is every Tuesday and Thursday from 11:00 to 15:30. Additionally almost every weekday we will have work sessions with whoever is available, to collaboratively see to the development.
- Meetings with TU Delft supervisors: Every week. The exact time depends on the availability of the supervisors and team members.
- Meetings with external clients: Every two weeks, more meetings can be planned if help is needed.

B.6 Risks

B.6.1 Internal

- Some of the team members will be working an internship during the project, which will affect the meeting availability. We will try to stick to two meetings per week but the time may change.
- Two team members will be leaving for Intergeo the days before the midterm report. Because of this, their roles can be taken over by the other team members (CM is taken over by PM and FD is taken over by SD).

C Planning

C.1 Work Packages

Since there will be overlapping areas of focus, the work has been divided in work packages.

WP No.	WP Title	Lead members	Start Week	End Week
1	Organisation	PM	Week 1	Week 2
2	VR Development	IE	Week 2	Week 10
3	Interview Testing	TL	Week 3	Week 6
4	Presentation	PM	Week 8	Week 11

C.1.1 Organisation

The first phase involves setting up the project in the first two weeks to get ready for the PID deliverable.

- T1: Setting up a general project workspace (Miro), a technical workspace (GitHub) and initial documents for the deliverables (Overleaf).
- T2: Schedule team and client meetings.
- T3: Writing the PID document to get a clear understanding of the problem, organisation and planning of the project. Incorporating the feedback afterwards.
- T4: Writing the midterm report and incorporating the feedback.

C.1.2 Development

The biggest Work Package is the development. Further insight in these tasks will become clear as the research is getting done.

- T1: Display a pointcloud through a local server using Potree as basis.
- T2: Develop the main functionalities like being able to walk and also the measuring.
- T3: Address the "could" functionalities. Mainly on implementing the feedback from the interviews.
- T4: Fix bugs, clean up the code and finish development.

C.1.3 Interview

- T1: Prepare the interviews by setting up a workflow.
- T2: Perform the interviews on the screens.
- T3: Perform the interviews in VR.

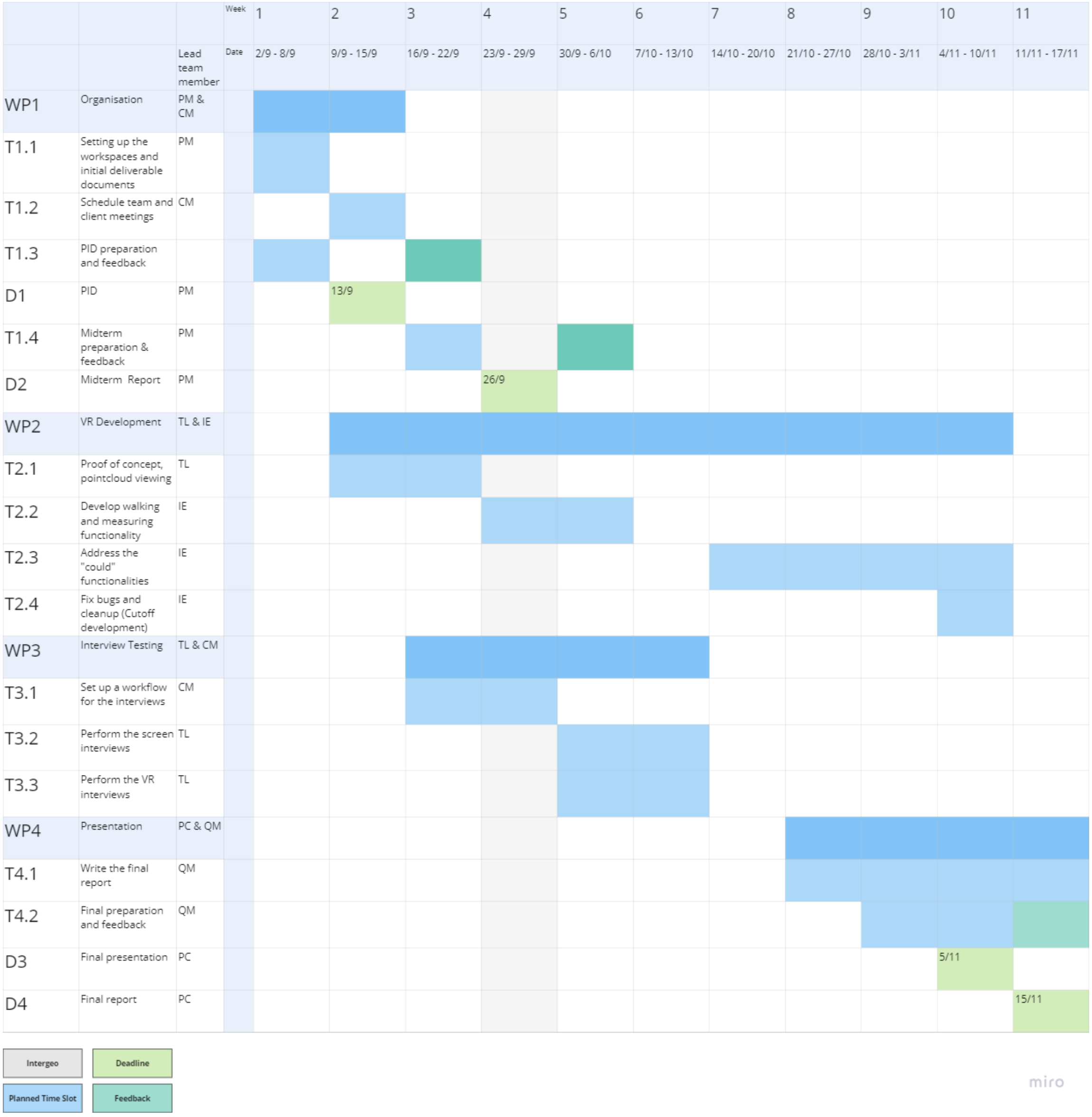
C.1.4 Presentation

The communication of our project through a report and presentation. The private presentation offers a chance to incorporate feedback for the presentation on Geomatics Day and the final report.

- T1: Writing the final report based on the conclusions.
- T2: Making the final presentation to reflect the report and incorporating the feedback to improve the presentation for Geomatics Day.

C.2 Deliverables

No.	Deliverable Title	WP	Lead member	Due Date
D1	PID	WP1	PC	13/09/2024
D2	Midterm Presentation	WP3	TL	26/09/2024
D3	Team Peer-Review Report	WP3	PC	27/09/2024
D4	Personal Reflection	WP3	All	27/09/2024
D5	Final Draft Report	WP3	PC	01/11/2024
D6	Final Presentation	WP5	PC	05/11/2024
D7	Final Deliverables	WP5	PC	15/11/2024


Week
1
2
3
4
5
6
7
8
9
10
11
Lead team member
Date
2/9 - 8/9
9/9 - 15/9
16/9 - 22/9
23/9 - 29/9
30/9 - 6/10
7/10 - 13/10
14/10 - 20/10
21/10 - 27/10
28/10 - 3/11
4/11 - 10/11
11/11 - 17/11
WP1 Organisation PM & CM
T1.1 Setting up the workspaces and initial deliverable documents PM
T1.2 Schedule team and client meetings CM
T1.3 PID preparation and feedback
D1 PID PM
13/9
T1.4 Midterm preparation & feedback PM
D2 Midterm Report PM
26/9
WP2 VR Development TL & IE
T2.1 Proof of concept, pointcloud viewing TL
T2.2 Develop walking and measuring functionality IE
T2.3 Address the "could" functionalities IE
T2.4 Fix bugs and cleanup (Cutoff development) IE
WP3 Interview Testing TL & CM
T3.1 Set up a workflow for the interviews CM
T3.2 Perform the screen interviews TL
T3.3 Perform the VR interviews TL
WP4 Presentation PC & QM
T4.1 Write the final report QM
T4.2 Final preparation and feedback QM
D3 Final presentation PC
5/11
D4 Final report PC
15/11
Intergeo
Deadline
Planned Time Slot
Feedback
miro

C.3 Gantt Chart

C.4 Rich Picture

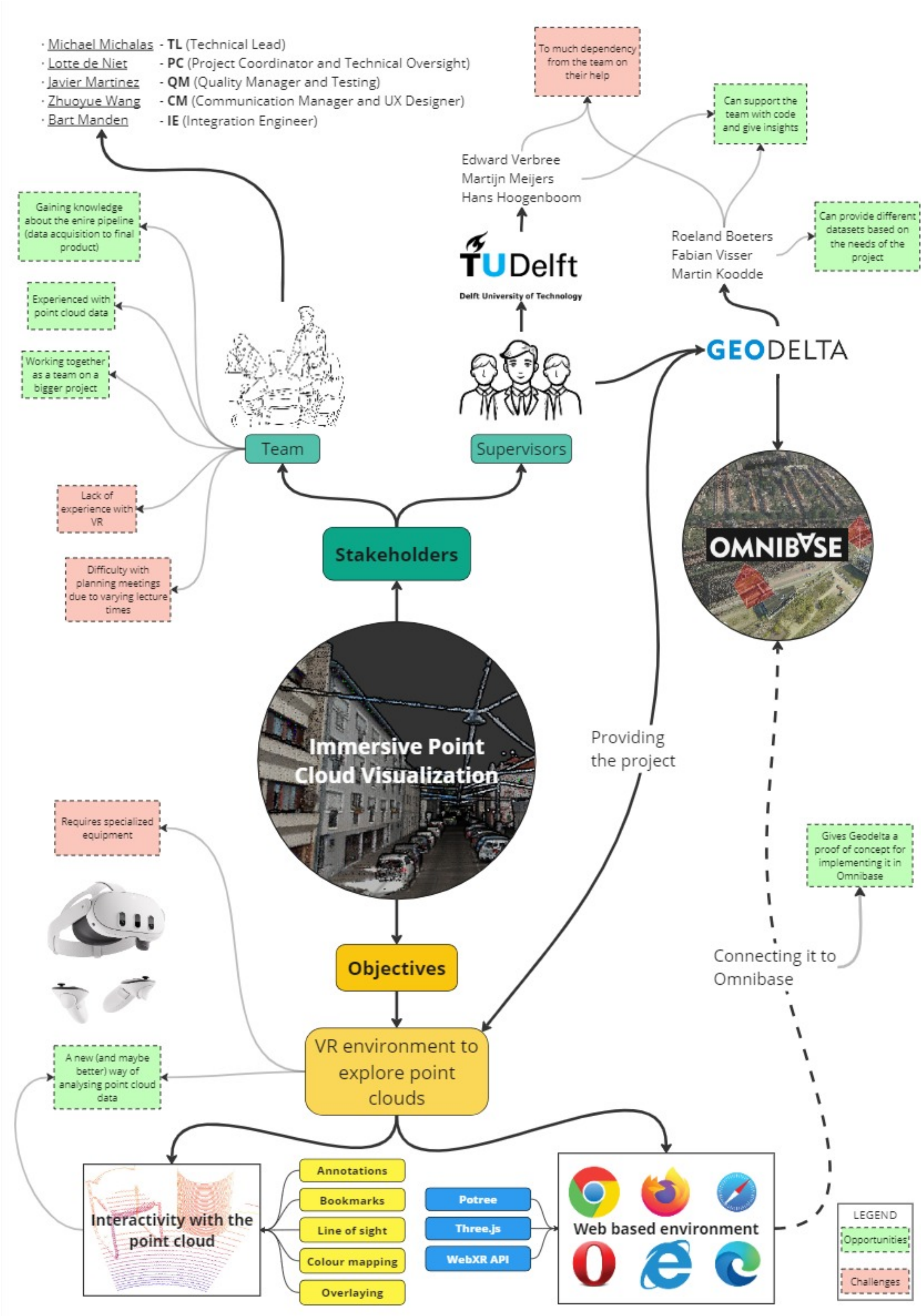


Figure 11: Rich Picture