

INTERACTION AND LOCOMOTION TECHNIQUES FOR THE EXPLORATION OF MASSIVE 3D POINT CLOUDS IN VR ENVIRONMENTS

F. Thiel², S. Discher^{1,*}, R. Richter¹, J. Döllner¹

Hasso Plattner Institute, Faculty of Digital Engineering, University of Potsdam, Germany

¹ (soeren.discher, rico.richter, juergen.doellner)@hpi.de

² felix.thiel@student.hpi.de

Commission IV, WG IV/9

KEY WORDS: Virtual reality, 3D point clouds, Real-time rendering, Locomotion, Interaction techniques

ABSTRACT:

Emerging virtual reality (VR) technology allows immersively exploring digital 3D content on standard consumer hardware. Using in-situ or remote sensing technology, such content can be automatically derived from real-world sites. External memory algorithms allow for the non-immersive exploration of the resulting 3D point clouds on a diverse set of devices with vastly different rendering capabilities. Applications for VR environments raise additional challenges for those algorithms as they are highly sensitive towards visual artifacts that are typical for point cloud depictions (i.e., overdraw and underdraw), while simultaneously requiring higher frame rates (i.e., around 90 fps instead of 30 - 60 fps). We present a rendering system for the immersive exploration and inspection of massive 3D point clouds on state-of-the-art VR devices. Based on a multi-pass rendering pipeline, we combine point-based and image-based rendering techniques to simultaneously improve the rendering performance and the visual quality. A set of interaction and locomotion techniques allows users to inspect a 3D point cloud in detail, for example by measuring distances and areas or by scaling and rotating visualized data sets. All rendering, interaction and locomotion techniques can be selected and configured dynamically, allowing to adapt the rendering system to different use cases. Tests on data sets with up to 2.6 billion points show the feasibility and scalability of our approach.



Figure 1. The virtual tape measure enables the user to take quick measurements in object space.

1. INTRODUCTION

Advances in remote and in-situ sensing technology allow the automatic generation of highly detailed digital twins of real-world assets, sites, cities, or entire countries at minimal cost and time (Eitel et al., 2016). The resulting data sets (i.e., large, unstructured collections of 3D points), are commonly referred to as 3D point clouds. They have become an essential data category for geospatial applications in diverse fields such as urban planning

and development (Nebiker et al., 2010), building information modeling (Stojanovic et al., 2018), or documenting natural and cultural heritage (Rüther et al., 2012).

Since 3D point clouds can consist of several billions of points, resulting in hundreds of gigabytes of raw data, they often exceed available main and gpu memory capacities significantly. Hence, to make full use of their potential and resolution, external memory algorithms need to be applied that limit the memory usage by dynamically fetching and unloading subsets of the data based on their relevance to the task at hand (Discher et al., 2019; Nebiker et al., 2010). As an example, state-of-the-art rendering systems for 3D point clouds determine representative subsets on a per-frame basis through a combination of view frustum culling and detail culling. To ensure an efficient subset retrieval, the data is hierarchically subdivided in a pre-processing step using spatial data structures such as octrees (Elseberg et al., 2013) or kd-trees (Goswami et al., 2013). In combination with web-based rendering concepts, these external memory algorithms allow the interactive inspection and visualization of arbitrary large 3D point clouds on a multitude of devices featuring vastly different CPU and GPU capabilities (Discher et al., 2018b; Schütz and Wimmer, 2015; Butler et al., 2014).

In recent years, *virtual reality* (VR) devices (e.g., Oculus Rift, HTC Vive) have emerged that enable new ways of presenting digital 3D content to the general public (Berg and Vance, 2017; Schütz, 2016). They allow for an immersive visualization of 3D point clouds, granting users the perception of being physically present within the captured site. However, they also raise additional challenges for established 3D point cloud rendering systems that are typically optimized for non-immersive applications (Discher et al., 2018a).

*Corresponding author

First, each scene has to be rendered for two displays simultaneously to generate a stereoscopic image. Second, visual artifacts such as overdraw (i.e., visual clutter) or underdraw (i.e., holes between neighboring points) tend to be more noticeable when seen through head mounted displays and can easily break the immersion. Third, to prevent feelings of motion sickness, state-of-the-art VR devices such as Oculus Rift or HTC Vive operate at 90 Hz. This requires frames to be rendered at significantly higher rates compared to non-immersive applications (i.e., where 30 to 60 fps are typical). Hence, rendering systems have to be optimized with respect to two conflicting goals: an improved visual quality of render artifacts and an increased rendering performance. Furthermore, VR applications require sophisticated locomotion techniques to efficiently navigate through a 3D virtual environment without causing motion sickness or loss of orientation. Intuitive interaction techniques, such as measuring and annotating within point cloud depictions, that have minimal impact on the rendering performance are also required.

In this paper, we present a rendering system that allows for an immersive exploration of arbitrary large 3D point clouds on Oculus Rift and HTC Vive. Our approach is based on a multi-pass rendering pipeline that efficiently combines *performance optimization techniques* speeding up the rendering and *image optimization techniques* improving the visual quality of the render artifacts (Section 3). To enable an in-depth exploration and inspection, we provide different *interaction techniques*, such as, to measure distances and areas (Fig. 1) or to scale and rotate visualized data sets (Section 4), as well as several *natural* and *artificial locomotion techniques* (Section 5), that can be selected and configured at runtime. We evaluated the presented techniques for real-world data sets with up to 2.6 billion points and present initial results of a pilot user study in Section 6. We end with a conclusion and an overview of future work.

2. RELATED WORK

A general overview of point-based rendering techniques is given by Gross and Pfister (2011). External memory algorithms as a means to render arbitrary large 3D point clouds were initially introduced by Rusinkiewicz and Levoy (2000) and have since been adopted by numerous authors (Martinez-Rubi et al., 2016; Richter et al., 2015). Visual optimization techniques for 3D point clouds aim to reduce overdraw and underdraw alike, either preventing such visual artifacts by rendering points with an appropriate size and orientation (Schütz and Wimmer, 2015; Preiner et al., 2012) or eliminating them via image-based post-processing (Dobrev et al., 2010; Rosenthal and Linsen, 2008). While those approaches usually focus on non-immersive applications, Discher et al. (2018a); Schütz (2016) discuss specific challenges and solutions regarding the visualization of 3D point clouds in VR environments. Our system extends their findings, focusing on locomotion and interaction.

Locomotion and interaction in virtual environments has been discussed by several authors. Studies from Wloka and Greenfield (1995) and Sarupuri et al. (2017) have shown that *six-degrees-of-freedom* (6DOF) input devices are preferred over more traditional devices such as keyboards, gamepads or haptic gloves. Objects that can be both seen and touched reinforced the user's sense of presence while locomotion based on the 6DOF input device was much less likely to induce motion sickness. *Walking-in-Place* (WIP) as a technique for locomotion was introduced by Slater et al. (1995a). User studies comparing WIP to joystick flying and real

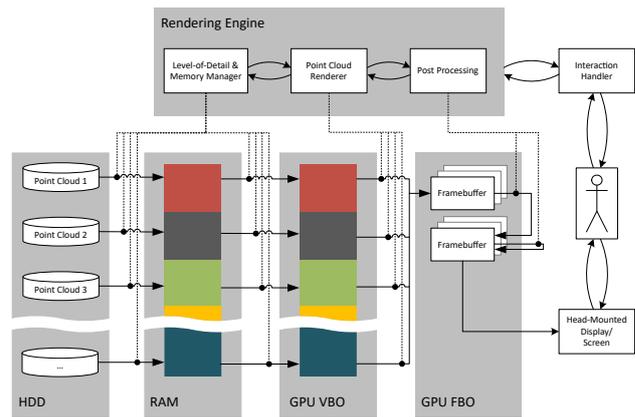


Figure 2. Rendering pipeline and data flow between hard disk drive (HDD), random-access memory (RAM), vertex buffer objects (VBO), and frame buffer objects (FBO). A common memory budget is shared by all rendered data sets.

walking (Slater et al., 1995b; Usoh et al., 1999) showed that techniques resembling walking in the real world gave the participants a stronger feeling of presence and less discomfort than artificial locomotion. To utilize this finding and cope with the challenge of virtual space being indefinitely bigger than physical space, *Redirected Walking* (RDW), originally proposed by Razzaque et al. (2001), alters the user's path by slightly rotating the virtual world to keep him or her within the available physical space. Studies (Razzaque et al., 2002; Razzaque, 2005) have demonstrated that RDW can be effective while being unnoticeable by participants. This concept has been augmented in the past years. Chen et al. (2018) proposed an algorithm to redirect the user through irregularly shaped environments with dynamic obstacles while Sun et al. (2018) demonstrated that saccadic suppression and the subsequent temporary blindness can be used to increase the rotation gains without the user noticing. The *Point & Teleport* (P&T) locomotion technique was proposed and evaluated by Bozgeyikli et al. (2016). In an experiment with WIP and joystick flying, they found P&T an intuitive, easy to use, and fun technique, though not more immersive or less prone to induce motion sickness as the other two. Still the participants rated it as their preferred technique.

3. SYSTEM OVERVIEW

Our rendering system extends the approach introduced by Discher et al. (2018a). A multi-pass pipeline (Fig. 2) is used to seamlessly combine three distinct render stages: Selecting subsets of representative points; rendering those subsets; applying image-based post-processing on the resulting render artifacts.

3.1 Data Subset Selection

While 3D point clouds often consist of billions of points, only a fraction of that data is actually relevant for a given frame. Subsets of *representative* points, which are manageable by available CPU and GPU capabilities, can be determined by applying (1) *view-frustum culling* (i.e., excluding points outside the current view frustum that would not be visible anyway), and (2) *detail culling* (i.e., by aggregating points based on their spatial position to accommodate the perspective distortion resulting in areas farther away from the current view position so they appear smaller on screen). To allow for an efficient retrieval of those representative points, the corresponding 3D point clouds need to be hierarchically subdivided in a pre-processing step. For that, we use a *kd-tree*. That

is, a binary tree whose splitting planes can be freely positioned alongside the respective coordinate axes. As a result, balanced tree structures can be guaranteed independence from the data's spatial distribution, which in turn minimizes average tree traversal times at runtime.

Furthermore, the hierarchical subdivision allows us to reduce visual overdraw and underdraw, which typically stems from points being sized inappropriately with respect to an area's local point density. Instead of assigning all points a uniform size, we take those local point densities into account by utilizing the kd-tree structure: For each node of the kd-tree, we determine its deepest descendant that has been selected for rendering. Based on that node's bounding box, a point size is calculated that is then assigned to all of its ancestors. As further discussed by Discher et al. (2018a), that approach provides efficient means to minimize overdraw but has a tendency of rendering points slightly too small, thus, creating underdraw in some areas. We fix those holes via image-based post-processing (Section 3.3).

Our pipeline allows to render multiple data sets simultaneously, in which case a different kd-tree is used for each 3D point cloud. While each data set is rendered separately, they all share a common memory budget, limiting the number of points that can be selected for rendering per frame. To account for varying performance rates (e.g., due to changing scene complexity and applied rendering techniques) that memory budget is adjusted dynamically to guarantee 90 fps at any time.

3.2 Subset Rendering

Selected data subsets are rendered into specialized frame buffer objects (FBO) storing multiple 2D textures, for example for color, depth, and normal values. Such *g-buffers* (Saito and Takahashi, 1990) provide efficient means to combine image-based post-processing techniques in the subsequent render stage. A similar approach is also used by the interaction handler to dynamically select and query individual points (Section 3.4). Different point-based rendering techniques can be combined at that stage. This makes it possible to adapt the appearance of each point to the current use case, for example, by applying different color schemes to facilitate context-specific visual filtering and highlighting.

Similarly, we combine different performance optimizing rendering techniques, which aim to prevent unnecessary shader operations by discarding fragments that do not contribute to the final image in a meaningful way as early as possible. While view frustum and detail culling already prevent most of such fragments, they do not exclude them entirely as some selected points may still partly occlude each other. Furthermore, VR devices restrict those parts of the screen actually visible to a circular area due to the radially symmetric distortion produced by the corresponding lenses. We filter fragments outside this visible area as well as occluded fragments by applying early *fragment testing*. This is done by using a separately rendered mesh to mask the hidden parts of the screen and rendering data subsets (i.e., the nodes of the underlying kd-tree) in the order of their distance to the current view position (Discher et al., 2018a).

3.3 Image-Based Post-Processing

A concluding image compositing stage is used to merge the previously generated g-buffers into a final image. The stage operates recursively and allows combining several image-based rendering techniques. These include highlighting silhouettes and adding

depth cues, smoothing aliasing and z-fighting, or filling holes between inappropriately sized, neighboring points (Dobrev et al., 2010; Boucheny, 2009). Those rendering techniques are typically based on separable *image convolution* filters that we speed up by applying two one-dimensional filter kernels rather than a single two-dimensional one (Dobrev et al., 2010). The applied post-processing techniques can be selected and configured at runtime, thus, introducing many degrees of freedom for graphics and application design.

3.4 Interaction Handling

The interaction handler is tasked with handling all user interaction and updating the visualization accordingly. This includes (1) configuring and selecting applied rendering techniques, (2) locomotion, (3) measuring distances and areas, as well as (4) transforming rendered 3D point clouds.

When required, the interaction handler is also able to communicate with the rendering and post-processing stage. An example of this is the Precise Measurement Tool (Section 4.2) which allows the user to select points for measurement. To minimize the performance impact of that interaction technique, we introduce a separate rendering pass — centering the view at the corresponding controller's location while looking alongside the pointing direction. Using an orthographic projection, the scene is then rendered into a separate g-buffer storing a single depth texture. Combined with the known location of the controller, the stored depth values provide efficient means to calculate the targeted point's position. We opted against a perspective projection to ensure a constant precision of the calculated positions independent of the distance to the view center.

4. INTERACTION TECHNIQUES

We implemented and evaluated three different interaction techniques that enable users to explore and manipulate rendered 3D point clouds. These techniques are an imprecise yet intuitive *virtual tape measure*, a *precise measurement tool* that allows querying distances and areas between dynamically selected points, and a *gesture-based transformation tool* providing means to scale and rotate data sets. All interaction techniques can be controlled with HTC Vive and Oculus Rift interchangeably, requiring only the corresponding motion controllers.

4.1 Virtual Tape Measure

The virtual tape measure is activated by simultaneously pressing the grip buttons on both controllers. When active, a yellow rectangle with equidistant markings is rendered between the controller models, while the measured length is constantly displayed above that rectangle (Fig. 1). Text and tape measure, are rendered using procedural textures and distance fields (Green, 2007) to prevent visual artifacts when viewed up close. The length displayed is calculated in object space. This enables users to measure objects that usually would not fit in an arm span, such as the height of a church tower or the length of a house, by rescaling the data beforehand. Both, the grabbing motion and the rectangle's appearance, contribute to the virtual tape measure resembling its real-world counterpart. In the sense of an interface metaphor (Marx, 1994), this resemblance enables users to transfer their knowledge of using real-world tape measures into the virtual world, making this technique easier to use. While being a quick and intuitive way to measure distances between two positions, the virtual tape's

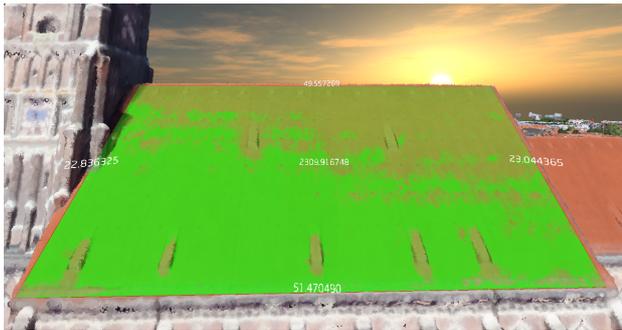


Figure 3. The precise measurement tool allows measuring both distances and areas by point selection.

precision heavily depends on the user's ability to keep his hands steady. Unlike the precise measurement tool, it also doesn't provide functionality to measure areas.

4.2 Precise Measurement Tool

The precise measurement tool (Fig. 3) allows dynamically selecting points to measure exact distances as well as the areas. The points are selected by pointing at them and pressing the trigger button, using the g-buffer based approach described in Section 3.4. This interaction technique provides two different modes:

1. In *distance mode*, selected points are connected by a line, and a label indicating its length is displayed next to it. After every two points the measurement is automatically completed. Selecting an additional point will start a new measurement.
2. In *surface mode*, selected points form a polygon and its area and the lengths of its outer edges are displayed. The first three selected points define a plane into which every subsequent point is projected. All projected points form a surface that is reconstructed upon each new selection, using either a *Bowyer-Watson triangulation* (Liu and Snoeyink, 2005) or an *advancing front reconstruction* (Mavriplis, 1995). The latter handles concave polygons better, but often leads to unintuitive results, especially when there is a large variance in edge length. We opted to project all points into a common plane to compensate for unavoidable inconsistencies during the data acquisition process, since even perfectly even surfaces will usually not be completely planar (Eitel et al., 2016).

Similar to the virtual tape measure, procedural textures and distance fields are applied to render distance and area labels. Pressing the left grip button starts a new measurement; while the previous lines and polygons are still displayed. The right grip button cancels the current measurement and with every subsequent press deletes previous measurements from last to first. A dedicated menu pane allows switching between distance and surface mode and choosing the method for the surface reconstruction.

4.3 Gesture-Based Transformation Tool

The gesture-based transformation tool (Fig. 4) is operated in a similar way as equivalent tools from other VR applications as Tilt Brush by Google¹ or Oculus Quill². Objects can be grabbed with one or both controllers by pressing the respective grip buttons. The

¹<https://www.tiltbrush.com/>

²<https://www.oculus.com/experiences/rift/1118609381580656/>



Figure 4. A gesture-based transformation tool allows to scale, translate and rotate 3D point clouds interactively.

grabbed objects can then be manipulated by performing gestures with the grabbing controllers. In this specific case, the following gestures are supported:

1. *Translate*: Real-world movement of the controller is directly translated to movement of the 3D point cloud. To the user, the 3D point cloud appears fixed to the moving controllers.
2. *Scale*: Moving both controllers apart or bringing them closer together scales the 3D point cloud up or down.
3. *Rotate*: Rotating both controllers around an imaginary point in between the controllers also rotates the 3D point cloud as well. However, to prevent motion sickness, this rotation is limited to the up-axis.

While scaling and rotating requires the use of both controllers, translating can also be triggered with just one.

5. LOCOMOTION TECHNIQUES

Users can navigate using different locomotion techniques: real walking, joystick flying, *point & teleport* (P&T), dashing, or locomotion based on gamepads and keyboards.

5.1 Keyboard/Gamepad Locomotion

The application can be navigated using a gamepad or a keyboard. On gamepads the left thumbstick controls the movement relative to the gaze direction. The speed of the movement can be controlled by the inclination of the stick, with no movement in the center position and maximum velocity when the stick is completely pushed in one direction. The right thumbstick enables rotating (left and right) or changing altitude (up and down) in the virtual world. The gamepad's right trigger button accelerates any movement. Keyboard controls are similar, with the WASD keys controlling movement and the arrow keys controlling rotation and tilt.

5.2 Real Walking

Position and orientation reported by the tracking system are taken into account so that physical movements are translated to equivalent motions in the virtual world. This enables running and walking, but also movements like bowing down (e.g., to inspect details of an object) crouching, sitting or lying prone on the ground.

5.3 Joystick Flying

The motion controller projects a green ray that indicates the movement direction. Pressing the trigger button allows movement along that direction in the virtual world. The movement speed increases linearly with the force of the trigger pull. Any changes to direction and speed of the movement are applied directly, conforming to the Oculus best practices guide³ that highly discourages gradual accelerations. While the trigger is being pressed, the direction of movement is strictly forward, symbolized by the green ray. We opted for that behavior due to an observation by Bozgeyikli et al. (2016) that humans tend to primarily move forward and rarely backward or to the side. Still, we allow users to explicitly move backwards or sideways in relation to the current forward direction by using the controller's touchpad (i.e., for HTC Vive) or analogue stick (i.e., for Oculus Rift) (Fig. 5). Again, the movement speed increases linearly with growing distance to the touchpad's center and the analogue stick's inclination, respectively.

5.4 Point & Teleport Locomotion

Based on a study by Bozgeyikli et al. (2016), recommending that technique explicitly for applications with an explorative nature, P&T enables an instantaneous movement to interactively selected positions in the virtual world. In addition to the green ray discussed in Section 5.3, a red cross is rendered onto the ray, indicating the position of the users feet after completing the teleport. The cross on top of the ray can be repositioned (i.e., shifted further away or pulled closer in) using a controller's touchpad or analogue stick. This target selection diverges from other P&T implementations and current teleportation-based VR games (e.g. *RoboRecall*⁴) in which the targeted position always snaps to the ground of the geometry. Due to the very nature of a 3D point cloud the generation of a corresponding ground mesh would require an additional preprocessing step. We avoid such a preprocessing step by allowing users to modify both, position and altitude above the perceived ground, as long as those modifications do not result in a position outside of the 3D point cloud's bounding box. The teleport is initiated by a single press of the trigger button. The forward direction is not changed, as the ability to do so was determined as a possible cause for disorientation by Bozgeyikli et al. (2016).

5.5 Dash Locomotion

The dash locomotion is a variation of P&T featuring the same ui elements and limitations (i.e., regarding the bounding box). Pressing the trigger button initiates the transport process, locking locomotion controls and moving the user in a fast and sudden gliding motion to the new position. On arrival the controls are unlocked once more for further movement. As recommended by the Oculus best practices guide, the acceleration is instantaneous and not gradually. It is claimed that short bursts of movement with a high, constant velocity are less likely to cause motion sickness, as the vestibular system can only detect acceleration and not speed.

6. EVALUATION

The presented rendering system was implemented based on C++, OpenGL, GLSL, as well as OpenVR⁵. Performance tests were run on a test system featuring an Intel Core i7-5820K CPU, 16 GB main memory (DDR4, 1200 MHz), and a GeForce GTX 980 with

³<https://static.oculus.com/documentation/pdfs/intro-vr/latest/bp.pdf>

⁴<https://www.epicgames.com/roborecall/en-US/home>

⁵<https://github.com/ValveSoftware/openvr>

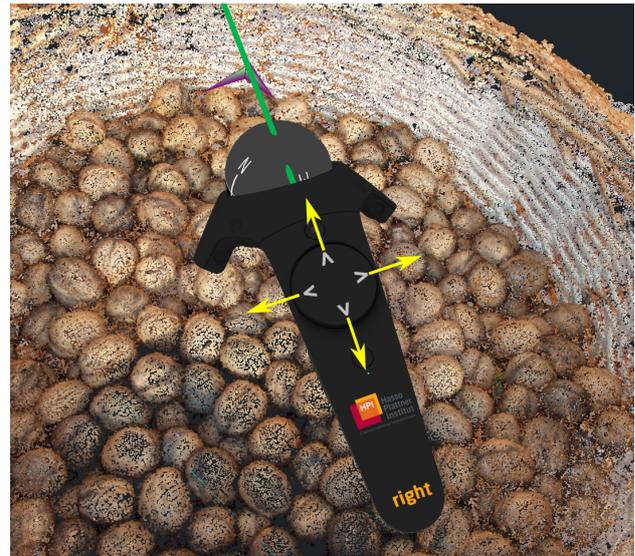


Figure 5. The touchpad allows movement in the selected direction relative to the forward direction symbolized by the green ray.

4096 MByte device memory (GDDR5, driver version 390.77). An HTC Vive was used as the primary output device, measurements on an Oculus Rift lead to comparable, slightly better results due to a tighter view frustum.

6.1 Rendering Performance

As test data sets, we used a mobile mapping scan of an urban area containing 2.6 billion points and a terrestrial indoor scan of an individual site consisting of 1.5 billion points. The corresponding kd-trees were generated within 145.9 minutes and 68.6 minutes, respectively, using a custom C++-based implementation. The rendering performance was measured for three different scenes: A pedestrian and a bird's-eye view of the mobile mapping scan as well as a close-up view of the indoor scan. To ensure the comparability of the performance test results (Table ??), we defined a static memory budget instead of using the dynamic one described in Section 3.1. Filtering hidden and occluded fragments through early fragment testing improves the rendering performance, although the effectiveness of those techniques varies, depending on the number of affected fragments. On the other hand, image optimization techniques, such as adaptive point sizes and image-based post-processing only have a moderate performance impact. While combining all post-processing techniques would amount to a significant performance drop, doing so will hardly be necessary as such a situation rarely presents itself.

6.2 Usability

To evaluate the usability of the implemented interaction and locomotion techniques, we conducted a pilot study featuring a small number of participants. We plan to report results of a full-fledged user study with more participants in subsequent work and want to use the tendencies discovered in the pilot study as a base for future hypotheses investigated by the future study.

6.2.1 Experimental Session and Measures Participants began with a training session introducing and explaining each technique by setting simple tasks like following a straight line, scaling up and down and measuring a rooftop while in virtual reality.

The first actual experiment was set within a historic building. Participants were asked to follow a predefined path while using

Scene	Default		Early Fragment Filtering		Image Optimization		
	Rendered Pts.		Hidden	Occluded	Adaptive Pt. Size	Filling	Depth Cues
Mobile Mapping (near)	19.8M	15.93ms	15.59ms	12.95ms	13.88ms		
Mobile Mapping (far)	6.9M	9.23ms	9.19ms	9.27ms	9.48ms	+ 1.07ms	+ 0.32ms
Terrestrial	11.6M	12.15ms	11.87ms	11.11ms	12.46ms		

Table 1. Average rendering performance in ms/frame. Dynamic memory budget was disabled to ensure comparability of measured values. Filtering of hidden and occluded fragments was enabled for all image optimization techniques.

the gamepad locomotion. The path was 80 meters long including various sharp turns and narrow doorways. After completion the participants were asked to fill a questionnaire about the locomotion technique used. They could rate the technique on a 5-point scale regarding intuitivity, ease of use, and comfort. They were also asked about signs of discomfort and positive or negative remarks in text. Trial and questionnaire were repeated for the remaining three locomotion techniques (joystick flying, P&T locomotion, dash locomotion). Afterwards, the participants had to complete a last pass, but this time were allowed to freely choose a technique and were asked to state the reasons for that choice on the questionnaire. The second experiment was set outside in an historic city center with wider spaces and a longer path of 945 meters with long straights. The procedure of the first experiment was commenced in this environment using an identical questionnaire.

For the third experiment, all locomotion techniques beside real-walking were disabled. Participants were placed in front of a church and tasked to measure the length of its roof using the virtual tape measure. To complete this task they had to use the gesture-based transformation tool to scale the scenery down beforehand. They were asked to fill a questionnaire afterwards, rating the tape measure on a 5-point scale regarding intuitivity, ease of use and precision. This procedure was repeated with the precise measurement tool as well. Afterwards they were asked to measure the size of the church tower with a technique of their liking and to give reasons for their choice. A last questionnaire was given to grade the gesture-based transformation tool on a 5-point scale regarding intuitivity, ease of use, and usefulness as well as to comment on the technique's overall comfort.

6.2.2 Participants The participants where 8 students aged 18 to 21, six male and two female. All had a background in computer science, but only four had used VR devices before. Of these, only one had sporadic contact with VR technology while the remaining three had experienced it solely through other user studies.

6.2.3 Results and Discussion

Gamepad Locomotion Gamepad locomotion had the second highest score in almost all categories in the first and the second experiment. Only in the indoor scene, it came third in the *comfort* category, surpassed by joystick flying and dash locomotion. Two participants reported nausea and dizziness. Participants made positive remarks about the familiarity with the controller mapping known from video games and the ability to turn in the virtual environment without having to turn around physically. They further praised the possibility for a fluid and fast motion enabling them to turn without having to stop. On the downside, they reported problems with the height control, due to it being mapped on the same stick as turning and, thus, being easy to trigger accidentally. One participant stated a feeling of disorientation when the body was moved simultaneously with gamepad input.

Joystick Flying Joystick flying had the highest score in all categories in the first and the second experiment. One participant

reported nausea, another experienced a short loss of balance in a fast turn that forced him to do a small side step to keep his balance. The participants made positive remarks about being able to change the speed with the trigger pull though some criticised it as too sensible. The control over the direction by pointing, allowing immediate corrections and course changes, was mentioned as positive as well. The necessity to turn the body to face the current movement direction faced a mixed reception: Some participants experienced it as natural while others found it cumbersome. No participant made extensive use of the touchpad for movement.

P&T Locomotion P&T locomotion was ranked third regarding *intuitivity* and fourth regarding *comfort* in both, the first and second experiment. Its *ease of use* was ranked third (outdoor) and lowest (indoor), respectively. Participants commented that the technique was a fast way to travel long distances. One participant reported nausea, whereas another one stated that it felt less unnatural and that he therefore felt less dizzy while using it. However, the necessity to place the cross on floor height to keep the current altitude and the repeated need for corrections was perceived as very negative. While some participants just experienced it as cumbersome or difficult, others reported that they repeatedly teleported to other locations than expected. The mandatory pause to adjust the position of the cross was perceived as disrupting.

Dash Locomotion The *intuitivity* of the dash locomotion was ranked fourth (indoor) and third (outdoor), respectively. Its *ease of use* was ranked third (indoor) and fourth (outdoor), its *comfort* second (indoor) and third (outdoor). One participant reported nausea as well as a light headache in the second experiment. The participants perceived the dashing motion as positive, especially in comparison with P&T. An important aspect for the participants was that they could see where they were going and did not have to regain their orientation afterwards. It also encouraged them to move by many smaller increments instead of trying to cover large distances by pushing the target far out. However, the main critique points of the P&T locomotion (i.e., the red cross and its necessary adjustment) were also present with the dash locomotion.

Virtual Tape Measure Compared with the precise measurement tool, the virtual tape measure ranked first regarding *intuitivity* and *ease of use*, but received a lower ranking regarding its *precision*. The participants had no problems using the technique but a few participants remarked that they would like to see both ends of the rendered tape for a more precise measurement.

Precise Measurement Tool The scores of the precise measurement tool were lower than that of the virtual tape measure regarding *intuitivity* and *ease of use*, but significantly higher regarding *precision*. Overall, the participants had no major issues with the technique, but noted that the required pressure on the trigger sometimes resulted in them missing the point they intended to select.

Gesture-Based Transformation Tool On the 5-point scales for *intuitivity*, *ease of use* and *usefulness*, the technique scored averaged values between 3 and 4 points. Participants found it easy to

change the scale of the scenery but commonly struggled with the grip button's location and with moving the scaled model around. Some participants experienced an accidental translation or rotation of the model due to not keeping the controllers in a straight line, crossing them, or moving them up or down while scaling.

Preferences: Locomotion When left with a choice, most participants chose the joystick flying, both in the indoor (5 participants) and the outdoor scene (6 participants). Reasons given were that they would not have to stop for adjustments, could see where they were going, and had at every moment control over speed and course. Fun was also a factor, as was not only noted quite often on the questionnaire, but could also be observed in the study: Participants commented that the application was missing a "superhero soundtrack" or made "hui!"- or "woosh!"-noises while moving. One participant seemed outright disappointed when the end of the path was reached and another compared it with a fast ride on a motorbike. However, four times gamepad locomotion was chosen by the participants. Two times for the indoor environment and two times for the outdoor environment. Reasons given were the ability to turn without body involvement via the analogue stick, and the control scheme similar to video games. While P&T locomotion was not chosen, one participant opted to use dash locomotion in the indoor environment for its precision and fluid movement.

Preferences: Measurement Six participants preferred the precise measurement tool while only two opted for the virtual tape measure. Reasons given were superior precision, the possibility to move between selections, and the fact that most measurements could be done without the necessity to scale the scenery. The two participants who selected the virtual tape measure stated that it gave them more flexibility in the measurement, as they found it difficult to select precisely the point they intended to select.

Discussion The low ranking of our implementation of P&T locomotion is noteworthy, as P&T and joystick flying were also compared in the study by Bozgeyikli et al. (2016). In their study the difference in score between the two techniques was much smaller and as most of the critique is centered around our modification of the described technique (i.e., selecting the target position via the red cross), we think it likely that this modification could be the reason for this anomaly. For the future study, the current implementation of P&T locomotion could be modified with an additional rendering pass similar to the precise measurement system. Thus, the teleport target would always stick to the perceived ground preventing unwanted altitude changes, but it would also restrict movement as users could only navigate from point to point. Vertical movement would be very limited and 3D point clouds with a low density could be hard to navigate. The dash locomotion technique should be altered similarly, as the scores of this pilot study indicate it being a more comfortable, less disorienting version of P&T, as long as the issues with the target selection can be solved. Another noteworthy tendency is the popularity of the gamepad locomotion as studies, as the one done by Sarupuri et al. (2017), indicate that users prefer locomotion techniques based on 6DOF input devices. We suspect that this might be a side effect of the computer science background and the relatively young age of the participants as they are likely to have grown up with video games and are very familiar with gamepads as input devices. This should be examined closely in the later study and therefore participants from various age groups should be selected. Joystick flying proved to be the preferred locomotion technique. The virtual tape measure, the precise measurement tool and the gesture-based transformation tool performed as expected though the measurement of

areas and a more elaborate comparison between the measurement techniques is planned for the later study.

7. CONCLUSIONS AND FUTURE WORK

3D point clouds represent an essential category of geodata used in a variety of geoinformation applications and systems. Commonly, the interactive visualization of 3D point clouds is just as important as their efficient processing or analysis, for example, to effectively communicate analysis results (Richter et al., 2015) or to present digital twins of real-world sites to a larger public (Martinez-Rubi et al., 2016; R  ther et al., 2012). Our rendering system enables users to interactively explore arbitrary large 3D point clouds on consumer-level VR devices, providing immersion-preserving visual quality and frame rates that avoid motion sickness at all times. To this end, we provide various rendering, interaction, and locomotion techniques that can be freely combined and configured at runtime, allowing for a task-specific graphics and application design. We deem our approach to be especially beneficial for the presentation of endangered or hardly accessible natural and cultural heritage to the general public, due to the created perception of being physically present within a captured site. Performance tests on several massive data sets show the feasibility and scalability of our rendering approach. Results of a pilot study indicate the usability of the provided interaction and locomotion techniques.

As future work, we plan to conduct a full-fledged user study to evaluate those initial results and further inspect some aspects of them. Results of this study will be presented in subsequent work. Furthermore, we consider distributing the stereo rendering across two separate GPUs as a means to further improve the rendering performance. To adapt our approach to support mobile devices, we plan to integrate web-based rendering concepts for thin clients as described by Discher et al. (2018b). By using a central server to render and distribute stereoscopic panoramas of the given 3D scene, client-side hardware requirements could be reduced drastically. Additional future work could focus on combining eye tracking technology with VR technology: Manufacturers of eye tracking solutions such as Tobii⁶ and Pupil labs⁷ are already producing integration kits for the HTC Vive, while the Oculus Half Dome prototype⁸ comes with eye tracking technology already installed. Such solutions would enable an implementation advanced redirected walking algorithms (Sun et al., 2018), allowing users to navigate a 3D point cloud in the same way they would explore a real-world scenery. Redirected walking in this context is especially important as the discrepancy between the available physical space and the area covered by the scan tends to be significant.

ACKNOWLEDGEMENTS

We thank Leon Masopust and Sebastian Schulz for their contributions to the implementation and Dr. Sharon Nemeth for copy-editing. Data sets have been provided by Illustrated Architecture, SHH sp.z.o.o, and virtualcitySYSTEMS.

References

Berg, L. P. and Vance, J. M., 2017. Industry use of virtual reality in product design and manufacturing: a survey. *Virtual Reality* 21(1), pp. 1–17.

⁶<https://www.tobii.com/product-listing/vr-integration/>

⁷<https://pupil-labs.com/blog/2018-04/htc-vive-pro-eye-tracking-addon/>

⁸<https://www.roadtovr.com/facebook-oculus-half-dome-prototype-vr-headset-140-degree-varifocal-f8/>

- Boucheny, C., 2009. Interactive Scientific Visualization of Large Datasets: Towards a Perceptive-Based Approach. PhD thesis, Université Joseph Fourier, Grenoble.
- Bozgeyikli, E., Raij, A., Katkooi, S. and Dubey, R., 2016. Point & teleport locomotion technique for virtual reality. In: *Proceedings of CHI PLAY*, pp. 205–216.
- Butler, H., Finnegan, D. C., Gadowski, P. J. and Verma, U. K., 2014. plas.io: Open source, browser-based webgl point cloud visualization. In: *AGU Fall Meeting Abstracts*.
- Chen, H., Chen, S. and Rosenberg, E. S., 2018. Redirected walking in irregularly shaped physical environments with dynamic obstacles. In: *Proceedings of IEEE Virtual Reality*.
- Discher, S., Masopust, L., Schulz, S., Richter, R. and Döllner, J., 2018a. A point-based and image-based multi-pass rendering technique for visualizing massive 3d point clouds in vr environments. In: *Proceedings of WSCG*.
- Discher, S., Richter, R. and Döllner, J., 2018b. A scalable webgl-based approach for visualizing massive 3d point clouds using semantics-dependent rendering techniques. In: *Proceedings of Web3D*.
- Discher, S., Richter, R., Trapp, M. and Döllner, J., 2019. Service-oriented processing and analysis of massive point clouds in geoinformation management. In: *Service-Oriented Mapping*, Springer, pp. 43–61.
- Dobrev, P., Rosenthal, P. and Linsen, L., 2010. An image-space approach to interactive point cloud rendering including shadows and transparency. *Computer Graphics and Geometry* 12(3), pp. 2–25.
- Eitel, J. U., Höfle, B., Vierling, L. A., Abellán, A., Asner, G. P., Deems, J. S., Glennie, C. L., Joerg, P. C., LeWinter, A. L., Magney, T. S. et al., 2016. Beyond 3-d: The new spectrum of lidar applications for earth and ecological sciences. *Remote Sensing of Environment* 186, pp. 372–392.
- Elseberg, J., Borrmann, D. and Nüchter, A., 2013. One billion points in the cloud—an octree for efficient processing of 3d laser scans. *ISPRS P&RS* 76, pp. 76–88.
- Goswami, P., Erol, F., Mukhi, R., Pajarola, R. and Gobbetti, E., 2013. An efficient multi-resolution framework for high quality interactive rendering of massive point clouds using multi-way kd-trees. *The Visual Computer* 29(1), pp. 69–83.
- Green, C., 2007. Improved alpha-tested magnification for vector textures and special effects. In: *Proceedings of SIGGRAPH 2007 Courses*, pp. 9–18.
- Gross, M. and Pfister, H., 2011. *Point-based graphics*. Morgan Kaufmann.
- Liu, Y. and Snoeyink, J., 2005. A comparison of five implementations of 3d delaunay tessellation. *Combinatorial and Computational Geometry* 52, pp. 439–458.
- Martinez-Rubi, O., de Kleijn, M., Verhoeven, S., Drost, N., Attema, J., van Meersbergen, M., van Nieuwpoort, R., de Hond, R., Dias, E. and Svetachov, P., 2016. Using modular 3d digital earth applications based on point clouds for the study of complex sites. *International Journal of Digital Earth* 9(12), pp. 1135–1152.
- Marx, A., 1994. Using metaphor effectively in user interface design. In: *Proceedings of CHI*, pp. 379–380.
- Mavriplis, D., 1995. An advancing front delaunay triangulation algorithm designed for robustness. *Journal of Computational Physics* 117(1), pp. 90 – 101.
- Nebiker, S., Bleisch, S. and Christen, M., 2010. Rich point clouds in virtual globes—a new paradigm in city modeling? *Computers, Environment and Urban Systems* 34(6), pp. 508–517.
- Preiner, R., Jeschke, S. and Wimmer, M., 2012. Auto splats: Dynamic point cloud visualization on the gpu. In: *Proceedings of EGPGV*, pp. 139–148.
- Razzaque, S., 2005. Redirected Walking. PhD thesis.
- Razzaque, S., Kohn, Z. and Whitton, M. C., 2001. Redirected walking. In: *Proceedings of EUROGRAPHICS*, pp. 105–106.
- Razzaque, S., Swapp, D., Slater, M., Whitton, M. C. and Steed, A., 2002. Redirected walking in place. In: *Proceedings of EGVE*, pp. 123–130.
- Richter, R., Discher, S. and Döllner, J., 2015. Out-of-core visualization of classified 3d point clouds. In: *3D Geoinformation Science*, Springer, pp. 227–242.
- Rosenthal, P. and Linsen, L., 2008. Image-space point cloud rendering. In: *Proceedings of Computer Graphics International*, pp. 136–143.
- Rusinkiewicz, S. and Levoy, M., 2000. Qsplat: A multiresolution point rendering system for large meshes. In: *Proceedings of SIGGRAPH*, pp. 343–352.
- Rüther, H., Held, C., Bhurtha, R., Schroeder, R. and Wessels, S., 2012. From point cloud to textured model, the zamani laser scanning pipeline in heritage documentation. *South African Journal of Geomatics* 1(1), pp. 44–59.
- Saito, T. and Takahashi, T., 1990. Comprehensible rendering of 3-d shapes. In: *ACM SIGGRAPH Computer Graphics*, Vol. 24 number 4, pp. 197–206.
- Sarupuri, B., Hoermann, S., Whitton, M. C. and Lindeman, R. W., 2017. Evaluating and Comparing Game-controller based Virtual Locomotion Techniques. In: *Proceedings of ICAT-EGVE*.
- Schütz, M., 2016. Massive time-lapse point cloud rendering in virtual reality. Presentation at SIGGRAPH.
- Schütz, M. and Wimmer, M., 2015. Rendering large point clouds in web browsers. *Proceedings of CESC* pp. 83–90.
- Slater, M., Steed, A. and Usuh, M., 1995a. The virtual treadmill: A naturalistic metaphor for navigation in immersive virtual environments. In: *Proceedings of Virtual Environments*, pp. 135–148.
- Slater, M., Usuh, M. and Steed, A., 1995b. Taking steps: The influence of a walking technique on presence in virtual reality. *ACM TOCHI* 2(3), pp. 201–219.
- Stojanovic, V., Trapp, M., Richter, R. and Döllner, J., 2018. A service-oriented approach for classifying 3d points clouds by example of office furniture classification. In: *Proceedings of Web3D*, pp. 2:1–2:9.
- Sun, Q., Patney, A., Wei, L.-Y., Shapira, O., Lun, J., Asente, P., Zhu, S., McGuire, M., Luebke, D. and Kaufman, A., 2018. Towards virtual reality infinite walking: Dynamic saccadic redirection. In: *Proceedings of SIGGRAPH*.
- Usuh, M., Arthur, K., Whitton, M. C., Bastos, R., Steed, A., Slater, M. and Brooks, Jr., F. P., 1999. Walking > Walking-in-place > Flying, in Virtual Environments. In: *Proceedings of SIGGRAPH*, pp. 359–364.
- Wloka, M. M. and Greenfield, E., 1995. The virtual tricorder: A uniform interface for virtual reality. In: *Proceedings of UIST*, pp. 39–40.