INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION
https://doi.org/10.1080/10447318.2023.2238945

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Pointing It out! Comparing Manual Segmentation of 3D Point Clouds between Desktop, Tablet, and Virtual Reality

Carina Liebers, Marvin Prochazka, Niklas Pfützenreuter, Jonathan Liebers, Jonas Auda, Uwe Gruenefeld, and Stefan Schneegass

University of Duisburg-Essen, Essen, Germany

ABSTRACT

Scanning everyday objects with depth sensors is the state-of-the-art approach to generating point clouds for realistic 3D representations. However, the resulting point cloud data suffers from outliers and contains irrelevant data from neighboring objects. To obtain only the desired 3D representation, additional manual segmentation steps are required. In this paper, we compare three different technology classes as independent variables (desktop vs. tablet vs. virtual reality) in a within-subject user study (N = 18) to understand their effectiveness and efficiency for such segmentation tasks. We found that desktop and tablet still outperform virtual reality regarding task completion times, while we could not find a significant difference between them in the effectiveness of the segmentation. In the post hoc interviews, participants preferred the desktop due to its familiarity and temporal efficiency and virtual reality due to its given three-dimensional representation.

KEYWORDS

Desktop PC; manual segmentation; point clouds; tablet; virtual reality

1. Introduction

High-quality digital reconstructions of existing physical objects are essential for many applications. Robot learning, for example, benefits from reconstructed objects because they are crucial for domain randomization (Xie et al., 2020). Such, they allow constructing various training environments from a set of reconstructed objects by randomizing their properties such as location or orientation (Tobin et al., 2017). Moreover, training in augmented or virtual reality with realistic representations of physical objects enables acceleration of the training in a safe environment and enables generalization to multiple real-world environments (Tobin et al., 2018). Although 3D object databases exist, they currently lack most object categories. An alternative is to 3D scan real-world objects with (depth) cameras where a physical object is scanned with an optical sensor, transformed into a point cloud, and reconstructed into a digital object (Barnefske & Sternberg, 2022). Nevertheless, following this approach, the scanned point cloud often requires additional segmentation steps to extract only the points relevant to the scanned object (i.e., removing outliers and points that belong to neighboring objects).

In previous work, automated and manual segmentation approaches have been proposed. While state-of-the-art approaches can automatically segment entire scenes, understanding never-before-seen environments remains an open challenge (Liu et al., 2021). Since segmenting unknown objects from depth camera data still results in imprecise segmentation, manual segmentation of point clouds remains relevant (Liu et al., 2021). It is often performed by successfully enclosing the target object through volumetric selection in the form of bounding boxes (Li et al., 2010; Montano-Murillo et al., 2020; Wirth et al., 2019).

Although multiple types of devices exist, the *desktop* is the standard device for manually segmenting objects from a point cloud (Wirth et al., 2019). In comparison, mobile devices, such as *tablets* or smartphones, are known for their comfortable, natural, and efficient manipulation (Yee, 2004). Some solutions explored combining *desktop* applications with mobile devices to make use of their interaction possibilities (Montano-Murillo et al., 2020). Instead, *virtual reality (VR)* was found to have advantages when dealing with complex data, such as facilitating the understanding of data through its spatial representation (Pearl et al., 2019; Whitlock et al., 2020). However, it was outperformed by *desktop* regarding its effectiveness and efficiency on various tasks. Yet new *VR* applications were introduced for segmenting objects from point clouds, highlighting its advantages, in particular, for complex scenes (Stets et al., 2017). Since all introduced devices have different advantages (precision on desktop, natural manipulation on tablet, and spatial representation in VR), this raises the following question: To what extent are different devices (*desktop*, *tablet*, or *virtual reality*) suitable for the segmentation of *simple* and *complex* point clouds in terms of efficiency and effectiveness (RQ)?

CONTACT Carina Liebers carina.liebers@uni-due.de University of Duisburg-Essen, Schuetzenbahn 70, 45127 Essen, Germany.

In this work, we investigate the manual segmentation of point clouds to enable the representation and manipulation of real-life environments digitally. We compare three different devices for manual point cloud segmentation: a *desktop* PC, a *tablet*, and *virtual reality* headset. Furthermore, we consider the influence of complexity, considering *simple* and *complex* point clouds. To do so, we developed an application with the same interface and segmentation functionalities for all three devices. As our interaction option for selecting multiple points, we focus on one of the current basic functionalities – volumetric selection using a bounding box (Li et al., 2010; Wirth et al., 2019). We measure the efficiency and effectiveness of the segmentation in a user study (see in Figure 5) by comparing the participants' results to a ground truth point cloud obtained by detecting the correct points using a 3D object model. We counterbalanced both the order of the devices and the order of the task complexity in the groups using a Latin square design. Furthermore, we evaluate each participant's assessment of the segmentation tasks on the different devices.

Our work makes the following contributions:

1. We introduce a multi-platform point cloud segmentation application for *desktop*, *tablet*, and *VR*.
2. We investigate point cloud segmentation on each of the three devices assessing efficiency and effectiveness.

2. Related work

Point clouds became a quasi-standard for 3D object representation of real-world objects (Barnefske & Sternberg, 2022). They are widely used to represent buildings, trees, or indoor scenarios (Xie et al., 2020). One point of a point cloud contains a position, specifying a surface point in a 3D-Cartesian coordinate system. It can further have properties such as color or normal vectors (Barnefske & Sternberg, 2022; Hoang et al., 2019). They are used as raw data to extract objects and labels for standard datasets for algorithm development, evaluation, and comparison (Xie et al., 2020). During segmentation, the distinct points of a point cloud are grouped into non-overlapping regions, which receive semantic labels (Barnefske & Sternberg, 2022; Xie et al., 2020). However, since point clouds represent an environment with dense data points, segmentation tasks can be challenging and can influence discovery tasks, obtaining object, spatial or contextual information (Elmqvist & Tsigas, 2008). Furthermore, as object density increases, occlusion between objects will likely accumulate (Argelaguet & Andujar, 2013). Such occlusion reduces the user's selection performance (Stürzlinger et al., 2007).

Since we compare different devices for segmenting objects from point cloud data, these challenges directly concern our comparison. Hence, we present approaches considering such challenges for 2D and 3D devices, like multi- and single-selection techniques, strategies dealing with occluded scenarios, as well as their advantages and limitations.

2.1. Selection methods using 2D applications

To control 3D scenes using a 2D application, a mapping from the 2D input surface to the 3D data space is required (Isenberg, 2011). A basic interaction strategy of 2D devices with 3D objects is the image plane method, whereas users interact with 2D projections of 3D objects in a plane (Pierce et al., 1997). This method formed a basis for current methods such as cutting plane techniques using only a single surface (Klein et al., 2012).

Further, the selection of objects on a 2D display can be influenced by the shape of the objects or the selection tool. Objects are often irregularly shaped, which makes the selection with rectangular selection tools challenging. To tackle this, one can employ strategies that enable a selection of a subset of points by encircling them using either mouse or direct touch input and then estimate the border of the encircled object surface algorithmically (Yu et al., 2012). This input method was later enhanced to utilize users' gestural input to interfere with such a cluster (Yu et al., 2016). To deal with occlusion in 2D applications, visual feedback was found to have a critical role in aiding a selection. It supports the users' estimating of the closeness of points positioned behind each other (Vanacken et al., 2009).

Like occlusion, hand and tracker jitter is a common problem when selecting objects in a 3D environment, negatively affecting user performance. An introduced strategy to deal with such effects is progressively refining the set of selectable objects with a sphere-casting (SQUAD) method, thus successively narrowing down the area of the object further until it is precisely defined. It was found to be more accurate and faster with small targets (Bacim et al., 2013).

Since working with 3D data only using a mouse and keyboard can be challenging, some approaches introduced hybrid techniques, like a desktop computer and a tangible and tangible input control using a tablet. It enables users to cut planes and select objects using a tactile ray-cast (Besancon et al., 2017).

2.2. Selection methods in virtual reality

Rendering point cloud environments in *VR* can help humans explore distant places without the information loss resulting from modeling (Bruder et al., 2014). Furthermore, presenting a human avatar in third or fist person view through a point cloud can help in scenarios where the visibility of a user's body is needed or enhance social *VR* experiences (Ridha-Mahfoudhi & Dang, 2019).

However, users can not only perceive point cloud scenes in *VR* but edit and interact with them (Virtanen et al., 2020). Selecting objects in a dense environment in *VR* is often done by using *volumetric methods* to specify a 3D region where the target object is contained. For example, Wirth et al. (2019) use a transparent rectangle between the left and right controller to annotate objects in a 3D point cloud. Therefore, all points belonging to the target object need to be inside the rectangle, and all points not belonging to the object have to be outside. Similarly, objects can be selected by defining a region of interest between virtual hands (Jackson et al., 2018) and Zhang et al. (2022) introduced a method for the arbitrary selection of regions of interest.

To select objects at a distance, a virtual ray or cone originating from the user's hand or viewpoint can be used. Their orientation can be defined through the hand position and orientation (Argelaguet & Andujar, 2013). It enables interaction with all objects within the field of view; however, similar to jitter in 2D applications, the precision is limited to the user's hand angular accuracy and stability (Argelaguet & Andujar, 2013). Volumetric tools, such as cones, might indicate more than one object on selection (Stürzlinger et al., 2007). Thus, there are some mechanisms to disambiguate such selection (Argelaguet & Andujar, 2013). Examples are Grossman and Balakrishnan (2006), who enabled the selection from a list of intersected objects or Bacim et al. (2013), who progressively refine the selection by performing selections until a single element is left.

To increase the precision of a selection, hybrid solutions were proposed. For example, Montano-Murillo et al. (2020) introduced a selection technique that allows selecting multiple objects in dense virtual environments (VEs) (e.g., point clouds). The technique allows creating a slicing volume in the VE. *VR* users could select target objects by placing a selection volume. It is projected onto a tablet for fine-grained adjustments of the selected objects. They found that a physical tablet improved selection accuracy compared to a pure mid-air approach.

Since occlusion is a challenge when dealing with point clouds, some strategies were proposed to deal with it. Most commonly, semi-transparency is used (Stürzlinger et al., 2007). For example, when using virtual rays, the opacity of objects in the line of sight can be changed, letting occluded objects appear (Elmqvist & Tsigas, 2008). Using slicing planes, segmentation tasks in partly or fully occluded environments can be accomplished by applying a cut upon a user's input to draw a defined region, as shown in Large Scale Cut Plane (Mossel & Koessler, 2016).

Although there are a variety of advanced techniques for selecting 3D objects with *desktop* and *tablet* and *VR* devices, in this work, we restrict ourselves to the fundamental rectangular volume selection. We choose this selection form as it allows us to compare the devices as fairly as possible.

2.3. Selection precision: The mouse and its superiority

Multiple studies compare mouse and keyboard input with other input methods regarding their performance level. In most cases, the mouse input was found to have a significantly higher performance, led to higher usability ratings, and increased productivity (Balakrishnan et al., 1997; Bérard et al., 2009; Jones et al., 2020; Teather & Stuerzlinger, 2008).

In 3D placement tasks, the mouse input outperformed a 2D tracker input, with and without a supporting surface and a three degrees of freedom (DoF) tracker regarding its movement time (Teather & Stuerzlinger, 2008). This result was later reaffirmed in an experiment evaluating the user performance and biosignals on a 3D placement task. The mouse input was not only found to be more efficient than the 3 DoF devices; it also induced more stress than using desktop device (Bérard et al., 2009).

However, for selecting objects in a 3D environment, the mouse's superiority does not remain unchallenged. Although mouse-based pointing was found to be fastest for targets positioned in the users' front view direction, targets placed behind a user were quicker selected using a ray-cast laser pointing technique (Petford et al., 2018). These results contradict the finding that the mouse input had the lowest movement times when selecting objects in a head-mounted VR game compared to the Razer Hydra game controller and a 3D tracker (Farmani & Teather, 2017). In a scenario where persons should select mid-air objects projected on a stereoscopic table, the results showed that using real hands was found to have the highest error rate but were the most effective technique at the same time. The virtual offset cursor and hand did not improve the overall performance (Bruder et al., 2013a). However, indications are suggesting that 3D pointing performance degrades for 3D but not for two-dimensional techniques when targets are displayed above a stereoscopic screen (Teather & Stuerzlinger, 2011). In a later Fitt's Law experiment investigating varying stereoscopic parallax, the results showed that 2D techniques are more efficient close to the screen, while 3D selection outperforms it for targets placed further away from the screen (in mid-air) (Bruder et al., 2013b). Regarding their accuracy and completion time, tangible mid-air input devices were found to support faster docking performance. Bare-handed interactions in mid-air achieved similar time performance and accuracy compared to constrained device (Vuibert et al., 2015). Contrastingly, in comparing a Leap Motion device that enables hand tracking and a mouse for target selection, the Leap Motion device led to lower user productivity, fatigue, and lower preference and usability ratings than mouse input (Jones et al., 2020). Koutsabasis and Vogiatzidakis (2019) systematically review mid-air interactions and their applications.

Although these works have compared the efficiency and effectiveness of various devices, we could not find any investigated *desktop*, *tablet*, and *VR* for segmentation tasks similar to point clouds, also considering the complexity of these tasks.

3. General approach

In this work, we want to answer our research question: To what extent are different devices (*desktop*, *tablet*, or *VR*) suitable for the segmentation of *simple* and *complex* point clouds in terms of efficiency and effectiveness (RQ)? Based on the related work, we assumed the following hypotheses:

H1 Segmenting 3D data on a *desktop* PC has the lowest task completion time (TCT).

H2 The physical demand and effort in *VR* is higher than on *desktop* and *tablet*.

H3 *VR* enables precise processing of the data, leading to higher correctness of the segmentation.

To answer these hypotheses, we conducted a user study. In the following, we introduce the segmentation procedure

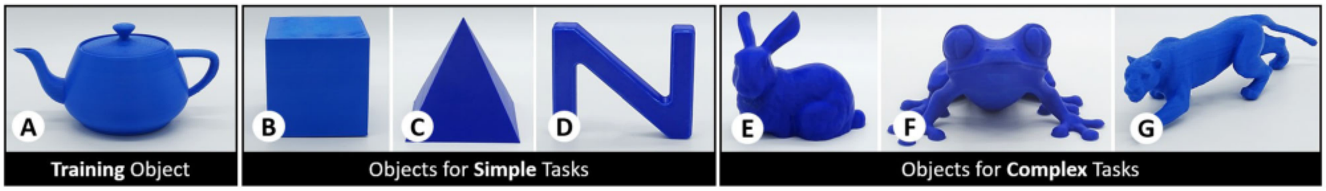

Figure 1. Printed 3D objects used for taking the point cloud images and as reference objects during the user study. They were split into three groups depending on their shape complexity: A) The Utah teapot as training object, objects for the simple segmentation tasks: B) a cube, C) a pyramid, D) N, and objects used for the complex segmentation tasks: E) the Stanford Bunny, F) a treefrog, G) a panther.

and corresponding functionality for each of the used devices. We further outline the point cloud creation and underlying ground truth data for our evaluation.

3.1. Generating point clouds and ground truth depth images

To enable the segmentation of an object for the comparison of the three devices (*desktop*, *tablet*, and *VR*), we created point clouds. They should represent different levels of the segmentation task complexity in real-life recorded point clouds, *simple* and *complex*. Additionally, we needed a base to evaluate which segmented points were correct, as ground truth images, to assess the procedure's effectiveness.

3.1.1. Designing simple and complex tasks

For the segmentation scene complexity, we considered multiple aspects. First, objects having more arithmetic contained *shapes* (Globa et al., 2016). These can include spatial features, intersections between model layers, faults, and unconformities or fractal dimensions (Pellerin et al., 2015; Reichert et al., 2017). Objects with multiple arithmetic shapes are more *complex* than others consisting of less and simpler features. Hence, we choose in the *simple* scenes target objects with clearly defined shapes, a cube, a pyramid, and an N, while the *complex* scenes included objects with finer details, the Stanford Bunny, a treefrog and a panther (see Figure 1). Although objects are usually not separated from others in their environment, a cumulative occurrence of *occlusion* increases the segmentation complexity (Montano-Murillo et al., 2020). Therefore, the selection of non-occluded objects located on a flat surface remains less challenging, as one can separate an object by selecting all points lying above the surface in most cases. In our images, we considered this by placing objects uncovered on a table in the *simple* scenes (see Figure 5) while they were covered and placed near other objects in the more *complex* scenes (see Figure 2A). Furthermore, sometimes capturing a scene from every angle is not possible. Missing recording angles can lead to incompletely represented objects in the picture (Dou et al., 2016). This increases the segmentation difficulty as an object's recognition might be difficult. Hence, we placed the objects for the *complex* scenes inside a cabinet, leading to missing information in the recorded images.

3.1.2. Enabling and facilitating ground truth images

Since we wanted to measure the effectiveness and efficiency of our participants' segmentations, we needed to determine which points originally belonged to the object. To avoid an intrinsic error due to our own segmentation, we resorted to already modeled 3D objects from Thingiverse[1].

First, we printed the *simple* and *complex* objects (see Figure 1) such that they had a similar height (12 cm). Since the participants should be able to familiarize themselves with the interface, we further printed the Utah Teapot as a training object. Its round form requires multiple segmentation angles, which provides enough material to test the interaction. All objects were colored blue to increase their identification.

We used the printed real-life models as the motive for taking the point cloud images. They were placed in the described *simple* and *complex* environments according to their shape complexity (see Figure 2A). We placed the teapot in the same environment as the *simple* objects to ensure that participants recognize it quickly and are not distracted by missing data during training.

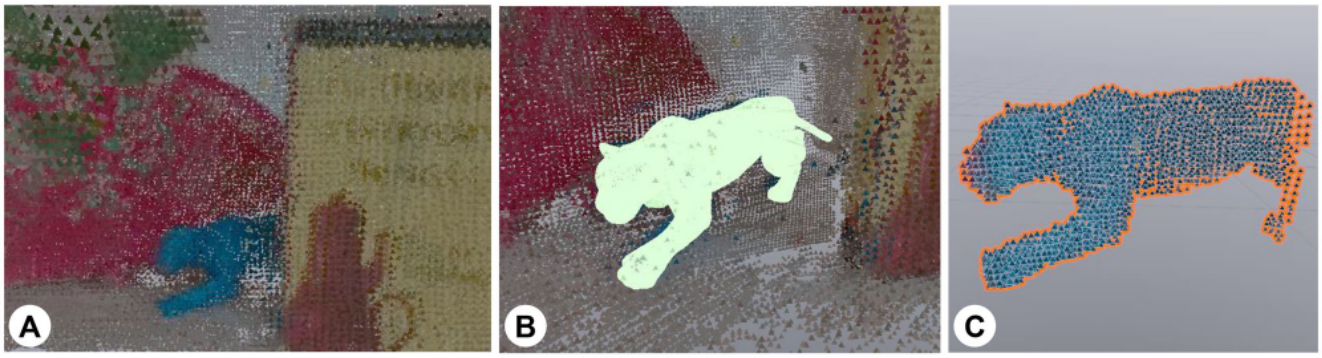

Figure 2. Workflow to obtain the point cloud images and ground truth data for the evaluation. We placed the printed 3D object in the scene and recorded it to receive a point cloud image (A). For analysis, we used the 3D model (B) to determine the points belonging to the object and saved them as the ground truth of the segmentation task (C).

We recorded the scenes with the Intel RealSense D435 RGB-D camera using a SLAM implementation in ROS. The recorded point clouds were post-processed by filtering outliers and removing points that did not belong to the area of interest, like walls and other objects in the room.

We used the original 3D model as a mask to determine the points belonging to the object. We decided to use this semi-manual procedure since recorded point clouds include artifacts. Automatic procedures, like extracting objects using color mapping, would in or exclude points recorded with false colors, for example, due to reflection. By overlaying the object in the point cloud with its mesh, we could calculate the belonging of the points (see Figure 2B). All points not contained in the mesh were deleted from the image, leaving only the points of the object (see Figure 2C). This segmentation was used as ground truth for comparison with participants' study results.

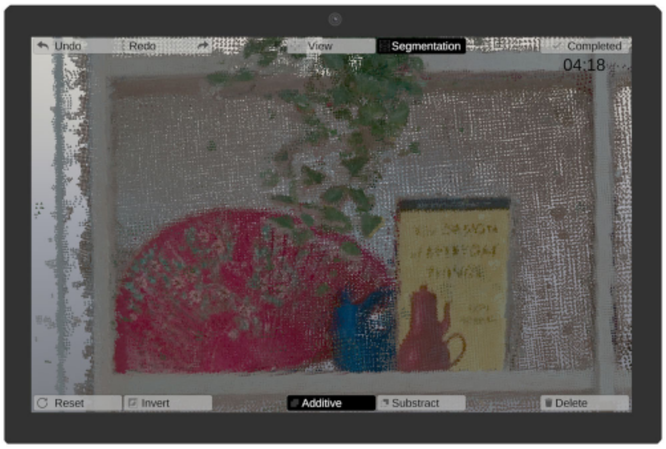

Figure 3. Interface of the *tablet* application. The buttons are grouped based on their semantic similarity, Undo ↶ and Redo ↷, View ✥ and Segmentation ▦, Revert ↻ and Invert ▨, Additive ■ and Subtractive ⬚, Delete 🗑, and Completed ✓. General functionalities are positioned at the top, and segmentation functionalities are at the bottom.

3.2. Segmentation application

To compare the different devices (*desktop*, *tablet* and *VR*), we developed the segmentation applications as similar as possible using *Unity3D* with one consistent graphical user interface (GUI) for all three devices (see Figure 3). The application used the same icons on each device to increase recognition of the functionalities. We obtained the icons from Blender[2] and ICONS8[3]. The application rendered the prerecorded point clouds of our objects within an empty room with white walls free of distracting details or limiting obstacles. We published both the source code and study applications online.[4]

Our application offers two modes – one for adjusting the view on the point cloud and one for the segmentation of objects. The *view* mode enables translating, rotating, and zooming of the point cloud image The *segmentation* mode enabled processing of the point cloud image.

3.2.1. View mode

Our application starts in the *view* mode to allow users immediate adjustment of their view on the point cloud.

3.2.1.1. Adjusting the view in 2D. The translation of the point cloud was implemented similarly for *desktop* and *tablet*. While the point cloud follows the mouse's movements on pressing the left mouse button in the *desktop view* mode, it follows the user's finger in the *tablet* application when one touch interaction is detected. All three rotations, pitch, yaw, and roll, were enabled for the 2D applications. For the *desktop*, moving the right-clicked mouse horizontally triggers the yaw rotation. Moving it vertically steered the pitch rotation. The roll rotation was enabled by pressing the gear wheel down and moving the mouse to the left or right. On the *tablet*, the rotations were steered using two-finger touch events, using the same directions as in the *desktop* application: Moving the two fingers horizontally caused a yaw rotation. A vertical movement with two fingers led to a pitch rotation. The roll rotation was triggered by moving the fingers clock or counterclockwise. To quickly switch to the initial axes, a small representation of the coordinate system was shown on the right-hand side of the screen. It enabled users to rotate the point cloud by clicking or touching the corresponding axes on the *desktop* or *tablet*, respectively. A zoom interaction could be performed by moving the mouse wheel in the *desktop* application or using a pinch gesture in the *tablet* application.

3.2.1.2. Adjusting the view in virtual reality. Users could translate and rotate the point cloud using controllers in *VR*. By pressing the index trigger, they could link the point cloud movement to the corresponding controller. It then

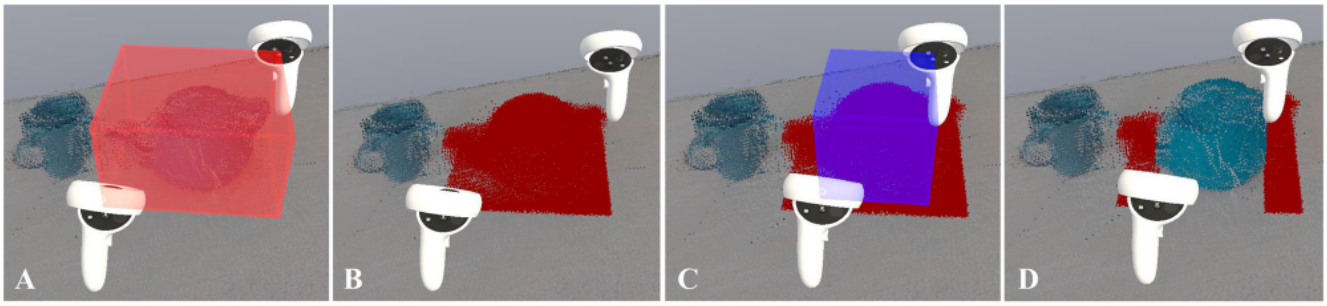

Figure 4. Selecting points using the cuboid selection tool in *VR*: Enclosing points with the red cuboid (A) led to the selection of these points, highlighted in red (B). The same functionality could be used to deselect points – by placing points inside the blue cuboid (C), they were deselected (D). Switching between the modes is done via the buttons in the interface.

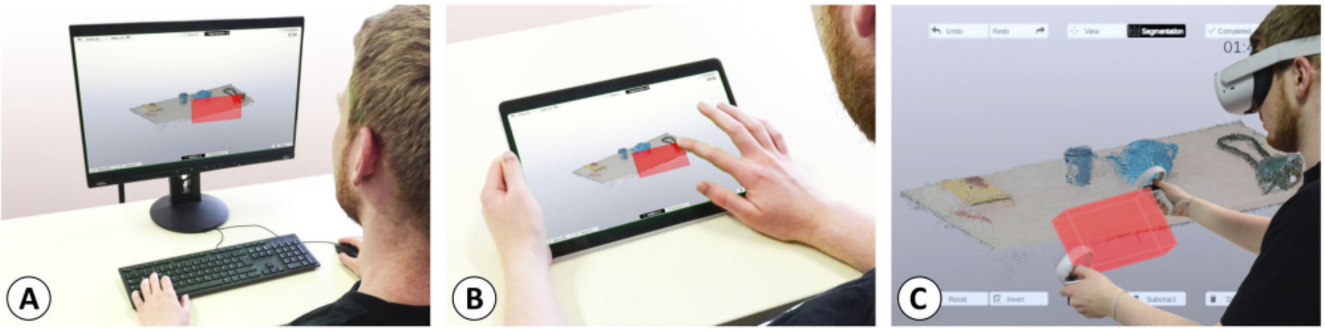

Figure 5. Study conditions to compare point cloud segmentation on a desktop PC (A), a tablet (B), and virtual reality (C) regarding their suitability for segmenting simple and complex point clouds. All pictures display the training scene of the study containing the *Utah teapot* as the target object.

followed the controller's movement and thereby was translated into space. By pressing the index triggers of both controllers simultaneously and increasing or decreasing their distance, users could in- or decrease the point cloud's size.

3.2.2. Segmentation mode

Users could switch from the *view* to the *segmentation* mode by pressing the corresponding button in the UI (see Figure 3). We applied dark background color to the active mode button to indicate the active mode.

In the *segmentation* mode, users could select parts of the point clouds using a cuboid volumetric selection. Such a selection tool is commonly used as basic selection for segmentation tasks to select multiple points simultaneously by spanning a bounding box across the desired points (Li et al., 2010; Montano-Murillo et al., 2020; Wirth et al., 2019). A user could place the cuboid over the point cloud to select points. All contained points were highlighted in red after selection (see Figure 4A, B). The selected points could afterward be deleted using the *Delete* 🗑 button. The users could place the cuboid over already selected points to deselected points (see Figure 4C). Upon releasing the cuboid, the enclosed points were deselected and, hence their highlighting disappeared (see Figure 4D). A user could determine the functionality of the cuboid (select or deselect) by either activating the *additive* mode by clicking the *Additive* ■ button or choosing the *subtractive* mode by activating the respective button (*Subtractive* ⬚) in the GUI. Since the *additive* and *subtractive* functionality exclude the other, the currently active button was highlighted with a dark background. They were placed next to each other in the middle at the bottom of the UI, as we hypothesized them being used often. We further set the color of the selection tool dependent on its active functionality: red when in *additive* mode, and blue in the *subtractive* mode (see Figure 4A, C). Furthermore, a user could remove all selections by pressing the *Reset* ↻ button. We also enabled inverting the current selection by pressing the *Invert* ◩ button: all selected points were thus deselected all prior deselected selected. A user could further undo (*Undo* ↶) or redo (*Redo* ↷) undone actions through the interface (see Figure 3).

3.2.2.1. Segmentation in 2D. Since the *tablet* and *desktop* applications only offer a 2D presentation, the volumetric selection tool is displayed as a rectangle. On the *desktop*, it is drawn by pressing the left mouse button, which sets the first corner of the rectangle and dragging it. Likewise, on the *tablet*, users set the first corner with an initial touch start event, moving the finger across the display spans a rectangle. Since such interaction only defines a planar region, we implemented the selection tool such that the missing dimension has an infinite length, marking all points positioned behind the selected area as well. We ensured that the selection was not influenced by perspective distortions by using an orthographic camera setting.

3.2.2.2. Segmentation in VR. In *VR*, where the users can perceive the scene in 3D, the selection tool is a cuboid. Users can place a selection cuboid into the scene by pressing the index trigger on one of their controllers, setting the initial cuboid's corner, and then moving the controller to span a cuboid between the initial corner and the position of the controller (see Figure 5C). When the index trigger is released, all points within the cuboid are selected. Moreover, simultaneously pressing both index triggers allows users to create a cuboid that spans between both controllers (see Figure 4A). Moving both controllers can be used to adjust the size and position of the cuboid. When the user releases the index triggers, the points within the cuboid are marked as selected (see Figure 4B).

3.2.2.3. Shortcuts. The *desktop* and *VR* applications enable the activation of all buttons in the interface (see Figure 3) through shortcuts. In the *desktop* application, the following keys could be used instead of the button listed in parentheses: 1 (*View* ✥), 2 (*Segmentation* ▨), + (*Additive* ■), − (*Subtractive* ⬚), del (*Delete* 🗑), R (*Reset* ↻), I (*Invert* ◩), ctrl + z (*Undo* ↶) and ctrl + y (*Redo* ↷). In *VR*, the shortcuts were linked to the controllers' joystick. All general functionalities were bound to the right controller: up (*View* ✥), down (*Segmentation* ▨), left (*Undo* ↶), right (*Redo* ↷). The joystick on the left controller enabled steering the specialized segmentation functionalities: up (*Reset* ↻), down (*Invert* ◩), left (*Additive* ■), right (*Subtractive* ⬚) and pressing it (*Delete* 🗑).

4. Evaluation

We conducted a user study to investigate the best device for segmenting objects from *simple* and *complex* point cloud scenes regarding efficiency and effectiveness.

4.1. Study design

We conducted a within-subjects controlled laboratory study to compare the different devices. Our independent variables were the type of device (*desktop*, *tablet*, and *VR*) and the level of segmentation task *complexity*. On each device, our participants had to segment two objects. The two segmentation scenes included one *simple* segmentation task and one more *complex* segmentation task in each trial. We grouped together the following objects according to their shapes and scene complexities: the *Stanford bunny* and the *cube*, the *treefrog* and the *pyramid*, and the *panther* and the *N* (see Figure 1). Furthermore, we counterbalanced both the devices' and task complexity in the groups using a Latin square, which resulted in eighteen configurations. As our dependent variables, we measured task completion time (TCT), segmentation correctness, and usability with the System Usability Scale (SUS) questionnaire from Brooke (1996), task load with the NASA Raw-TLX questionnaire proposed by Hart (1986); Sandra G. Hart (2006), individual Likert items, and technology ratings. The NASA-TLX questionnaire is frequently used for interactive segmentation tasks Ramkumar et al. (2017). Moreover, we conducted semi-structured interviews at the end of the study to gather qualitative feedback.

4.2. Apparatus

For the *desktop* device, we used a monitor with full high definition (HD) resolution (1920 × 1080 pixels). The application could be controlled via a connected laser mouse and keyboard. As the device for the *tablet* application, we use the Samsung Galaxy TAB S7. Its 12.4-inch touch-enabled display offers a screen resolution of 2560 × 1600 pixels. We used the *Meta Quest 2* head-mounted display (HMD) as a device for *VR* because stand-alone operation is possible. The application is displayed with 1832 × 1920 pixels per eye using a refresh rate of 72 *Hz* and is also suitable for persons who wear glasses.

The study was conducted in a room that contained a desktop workplace and a $4.07m \times 4.05m$ free space for using the *VR* application. Additionally, the safety guard was set in *VR* beforehand, and the experimenter paid attention to ensure that the participants did not leave the designated area during the study. On each device, we recorded the user session data (e.g., interaction times and resulting point cloud segmentation) for later analysis (Agarwal et al., 2020).

4.3. Procedure

At the beginning of the study, we introduced our participants to the study procedure and goal. We addressed all open questions and emphasized that we measure the task completion time and precision of the resulting segmentation. The participants were informed that they could stop their participation without any drawbacks at any time. The study began after the participants gave their written consent.

We split the study into three blocks– one for each device. First, the participants entered a training scene where they could freely explore the application's features. The scene included the *Utah teapot* standing on a table with a mug, a book, and a camera (see Figure 5). As in the following tasks, the participants were asked to segment the teapot. Participants were informed that the scenes might include artifacts, so points may have different colors, causing color assignments to result in possible errors. The experimenter placed the printed teapot next to the participant to enable verification of its properties. In *VR*, the object was placed on a table behind the safety guard so that participants could view it using the see-through functionality introduced before the training started. During training, which lasted for a maximum of fifteen minutes, the experimenter answered all questions regarding the usage of the application. Furthermore, the experimenter ensured that all functionalities were applied at least once. If a participant did not use a functionality, the experimenter suggested it. In *VR*, the experimenter gave verbal instructions to ensure the use of all features before allowing the participant to explore the application on their own. After familiarizing themselves with the application, the participants could begin the segmentation tasks. As in the training scene, the experimenter placed the object to be segmented next to the participant. The participants had five minutes to remove all pixels that did not belong to the target object, after which the scene automatically ended. If the participant finished before the time expired, they could end the scene themselves by clicking the *Complete* ✓ button (see Figure 3). After all segmentation tasks on one device were finished, the participants were asked to complete the NASA TLX Index from Hart (1986) and the SUS questionnaire proposed by Brooke (1996). They then answered custom Likert items and questions regarding their assessment of segmentation on the different devices. After the participants finished all tasks on all devices, we conducted a semi-structured interview. Each participant took approximately 1 hour and 15 minutes for the entire study.

4.4. Participants

Eighteen volunteers (twelve male, six female, and zero nonbinary) participated in our user study. The median age of the participants was 31 years ($M = 32.83$, $SD = 8.54$, $Min = 25$, $Max = 62$). Regarding their expertise with manual point cloud selection, 13 said they had never segmented three-dimensional objects before, while five said they had done manual segmentation a few times. While all participants reported using a *desktop* computer every day, the usage of *tablet* and *VR* devices varies. For *tablet* devices, three participants said they use one every day, and seven said they use one frequently. Two said they use a *tablet* sometimes, five said they had used it a few times, and one participant said they had never used a *tablet* before. Although none of our participants use *VR* daily, five use it frequently, two sometimes use it, eight used it a few times, and three participants never used it.

4.5. Ethics

To ensure the participants' privacy, we pseudonymized the data at the beginning of the study. After finishing the study, we deleted the assignment from the participant's personal data. The study was approved by our ethics committee.

5. Results

In the following, we present the results of our evaluation. We recorded the Task Completion Time (TCT) of each segmentation and the final segmentation results of our participants. Furthermore, we present subjective results gathered from post-study questionnaires.

5.1. Quantitative analysis

In the following, we introduce the quantitative results of our evaluation. For the nonparametric data, we applied the Aligned Rank Transformation (ART) using the ARTool toolkit and conducted a paired-sample t-test with Tukey correction as a post hoc analysis, as was suggested by Wobbrock et al. (2011).

5.1.1. Task completion time (TCT)

We measured the TCT for each performed segmentation task. The TCT per task had an upper bound of five minutes (maximum time). We list all mean values and the interquartile range (IQR) of all measured times in Table 1.

As the normality assumption of TCT was violated (p = 0.048), we performed a two-way repeated measures analysis of variance (RM-ANOVA) after transforming our nonparametric data using ART (Wobbrock et al., 2011). We found a significant effect of *device* ($F_{2,85} = 20.71, p < 0.001$) on TCT. In the post hoc analysis, we found a significant difference between *desktop* vs. *VR* ($t(85) = -3.940, p < 0.001$), but no difference between *desktop* vs. *tablet* ($t(85) = -1.507, \mathrm{p} = 0.407$) or between *tablet* vs. *VR* ($t(85) = -2.433, \mathrm{p} = 0.051$). Here, we can conclude that *desktop* is significantly faster than *VR*. We also found a significant effect of the task complexity ($F_{1,85} = 20.71, p < 0.001$) on TCT and can conclude that *simple* tasks are significantly faster than *complex*.

Moreover, we found a significant interaction effect for *device* × *complexity* ($F_{2,85} = 3.36, p < 0.001$). In the post hoc analysis, we found a significant difference between some conditions (see Table 2). From these findings, we can conclude that *desktop* and *tablet* are impacted by *complexity*, but we did not find a difference for *VR*.

Table 1. Task Completion Time (TCT) between *desktop*, *tablet*, and *virtual reality (VR)* for simple and complex segmentation tasks.

	Overall		Simple		Complex	
	Mean	IQR	Mean	IQR	Mean	IQR
Desktop	248.08	98.46	210.90	144.70	295.64	47.45
Tablet	270.16	82.12	265.07	144.18	300.01	28.15
VR	290.73	32.95	299.91	63.58	300.00	3.10

We report the mean and interquartile range (IQR) values in seconds.

Table 2. Significant interaction effects between *device* × *complexity* on the task completion time (TCT) using the Aligned Rank Transformation (ART) with Tukey correction.

Contrast	estimate	SE	df	t.ratio	p-value	significance
Desktop (C) – Desktop (S)	23.444	7.155	85	3.277	<0.05	*
Tablet (C) – Desktop (S)	36.722	7.155	85	5.133	<0.0001	****
Tablet (C) – Tablet (S)	30.667	7.155	85	4.286	<0.001	***
VR (C) – Desktop (S)	38.889	7.155	85	5.435	<0.0001	****
VR (C) – Tablet (S)	32.833	7.155	85	4.589	<0.001	***
Desktop (S) – VR (S)	−23.222	7.155	85	−3.246	<0.05	*

5.1.2. Segmentation correctness

To determine the segmentation correctness of our participants, we recorded the resulting segmentation of each trial (i.e., the points that our participants have left over from the point cloud). We compared the final segmentation to the ground truth of our objects (see Section 3.1).

Following, we report the F1 score as our participants' segmentation correctness. We choose this score as it is the harmonic mean of precision and recall: Recall determines the proportion of correctly shown points (true positive (TP)) of the participants' segmentation from those that should be displayed based on the ground-truth data (TP + false negative (FN)), while precision indicates the proportion of correctly shown points (TP) from the overall result of the participant (TP + false positive (FP)). We chose this metric since comparing correctly deleted points (true negative (TN)) in the segmentation is disproportionately high, so measured differences are difficult to report. It results from point clouds containing several hundred thousand to millions of points (see Figure 2A), whereas the amount of points belonging to one object is considerably low (see Figure 2C). Only considering the recall might distort a comparison since it would automatically be perfect when the participants did not segment the image. Therefore, reporting the F1 score was more meaningful in this study. The F1 score, the harmonic mean of precision and recall, is calculated by dividing the multiplication of both through their sum, as in the following formula: $\frac{Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN}$. This value indicates the overall correctness of the participant's segmentation (see Figure 6, right). Thus, a higher value signifies higher segmentation correctness.

We list all mean values of the F1 scores per device and task complexity and their IQR in Table 3. Figure 6 provides an overview of the overall values. As the normality assumption of the F1 score was violated between the conditions ($p < 0.001$), we performed a nonparametric two-way repeated measures analysis of variance (RM-ANOVA) using ART (Wobbrock et al., 2011). We determined whether *device* × *complexity* significantly influences the F1 score. We found a significant effect of *complexity* ($F_{1,85} = 84.85$,

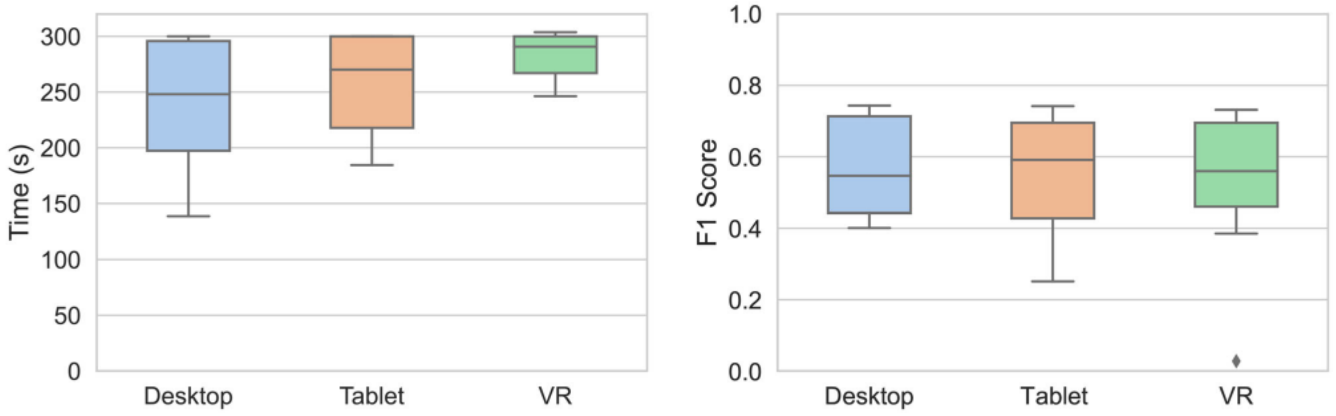

Figure 6. Left, measured Task Completion Times (TCTs) when using *desktop*, *tablet* or *virtual reality* right. Right, the corresponding F1 scores per device (higher values indicate a higher segmentation correctness).

$p < 0.001$) on the F1 score and can conclude that *simple* tasks are significantly more correct than the *complex* segmentation tasks. However, we did not find a significant interaction effect for *device* $(F_{2,85} = 0.22, p = 0.806)$ or *device* × *complexity* $(F_{2,85} = 0.35, p = 0.706)$.

5.1.3. Subjective measures

In the following, we present the subjective feedback of our participants. For ordinal data, such as Likert items, we applied a Friedman test to analyze if a difference between the devices exists and performed the exact Wilcoxon test with Bonferroni correction as post hoc analysis.

5.1.3.1. System usability scale (SUS). We evaluated the participant's responses to the individual SUS questionnaires from Brooke (1996) items for each device. The median SUS score (interquartile range) across all participants for each *device* are: $desktop = 78.75$ (IQR = 10.0), $tablet = 71.25$ (IQR = 20.63), and $VR = 80.00$ (IQR = 12.5). Since all these values are above 70, all applications were rated as acceptable (Bangor et al., 2009). A Friedman test did not reveal a significant effect of the devices on the scores of the SUS $(\chi(2) = 2.901, p = 0.234, N = 18)$.

5.1.3.2. Task load index. The participants' overall NASA Raw-TLX scores for the different devices are as follows, listed as median (interquartile range) in ascending order: $desktop = 37.50$ (IQR = 12.5), $tablet = 39.50$ (IQR = 12.75), and $VR = 42.5$ (IQR = 23.25). A Friedman test $(\chi(2) = 3.3913, p = 0.1835, N = 18)$ did not reveal a significant effect of the devices on the TLX scores. Since the TLX is ill-formed, as stated by Hart (2006), we further evaluated the individual TLX scores and report the ones where we found a significant difference after performing a Friedman test.

The values for *Physical Demand* are listed in the following as median (interquartile range) in ascending order: $desktop = 3.00$ (IQR = 2.75), $tablet = 3.50$ (IQR = 5.00), and $VR = 7.50$ (IQR = 8.75). A Friedman test $(\chi(2) = 11.4, p = 0.0033, N = 18)$ indicated significant differences between the devices' scores for physical demand. A post hoc test using the Exact Wilcoxon test with Bonferroni correction showed significant differences between *desktop* and *VR* $(p < 0.05, r = -0.699, p = 0.007)$.

Table 3. Segmentation correctness between *desktop*, *tablet*, and *virtual reality* for *simple* and *complex* segmentation tasks.

	Overall		Simple		Complex	
	F1	IQR	F1	IQR	F1	IQR
Desktop	0.55	0.27	0.71	0.19	0.44	0.16
Tablet	0.59	0.27	0.70	0.13	0.43	0.17
VR	0.56	0.24	0.70	0.17	0.46	0.18

We report the F1 score and the interquartile range (IQR) per condition.

5.1.3.3. Additional feedback. We wanted to understand how our participants perceived the adjustment and segmentation of the point clouds on the different devices. Therefore, we gathered subjective feedback on specific aspects of the segmentation procedure using a seven-point Likert-Scale (see Figure 7).

Our participants rated the statement that it was very easy to navigate the point cloud scene and adjust the view as follows, listed as median (IQR): $desktop = 5$ (IQR = 3), $tablet = 5$ (IQR = 2.75), and $VR = 6.5$ (IQR = 1). A Friedman test $(\chi(2) = 11.04, p = 0.004, N = 18)$ indicated significant differences between ratings of the devices regarding the point cloud adjustment. Applying exact Wilcoxon tests with Bonferroni correction for the three device groups revealed significant differences between *desktop* and *VR* $(p < 0.05, r = -0.653, p = 0.016)$ as well as between *tablet* and *VR* $(p < 0.05, r = -0.806, p = 0.001)$. Although our participants agreed to this statement for all devices, we conclude from our findings that *VR* was rated as significantly better than *desktop* and *tablet*.

In response to the statement of having no difficulties extracting the object, our participants responded as follows, listed as median (interquartile range): $desktop = 5$ (IQR = 2), $tablet = 3.5$ (IQR = 1), and $VR = 4$ (IQR = 2). We could not find a significant difference between the device technologies by applying a Friedman test $(\chi(2) = 3.5, p = 0.174, N = 18)$.

5.1.3.4. Technology ratings. We asked the participants to rate which device they favored the most and which the least for segmenting point clouds. Since we asked them about

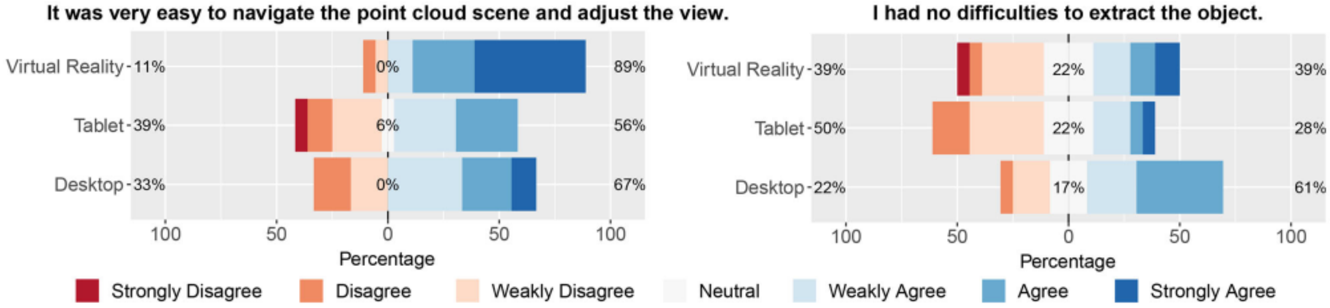

Figure 7. Boxplots of the values given to the individual Likert-Scale items regarding the navigation and segmentation of the point cloud.

their first and last preference, we could establish a ranking. On the question of which device they favored the most for such segmentation tasks, eight participants answered *desktop* and *VR* while *tablet* was named twice. As the second most favored device, *desktop* and *VR* were each mentioned 7 times, whereas *tablet* was mentioned only 4 times. *tablet* was rated as the least favorable device for segmentation (12 times), while *desktop* and *VR* was only rated as worst 3 times. We performed a Friedman test to evaluate if these device preferences significantly vary. It indicates significant differences on all three levels: The rating which one was favored for a segmentation task ($\chi(2) = 16, p = 0.0003, N = 18$), which one was the second most favored for a segmentation task ($\chi(2) = 14$, $\mathrm{p} = 0.0009, \mathrm{N} = 18$), and ($\chi(2) = 24, p < 0.0001, N = 18$) which one was least favored for a segmentation task. Overall, we observed that our participants preferred both the *desktop* and *VR* device over *tablet*.

5.2. Qualitative analysis

At the end of each study, we conducted semi-structured interviews. To analyze the data, we combined all the answers from the interview sessions and conducted a thematic analysis after (Braun & Clarke, 2006). Three researchers coded the data throughout the process and discussed potential codes and themes. In the following, we present the identified themes in detail.

5.2.1. Movement and navigation

When comparing, participants often referred to the navigation and perception of the point cloud on the different devices. The most positive comments on the navigation by far were made regarding *VR* (11 out of 18 participants), while *tablet* received the most negative comments (9).

Referring to the *desktop*, four participants commented negatively, highlighting a need for further controls like using the keyboard to move the view or emphasizing that switching between *view* and *segmentation* mode was difficult *"because I had to change frequently"* (P9).

We only received negative comments regarding the navigation on the *tablet* (9). The participants often had difficulties using the touch interaction to translate or rotate the point cloud. They expressed not being used to navigate through point cloud data and particularly stated to have *"problems to zoom with the pinch, because it rotated as well"* (P2) or to manipulate the view: *"I found it more difficult to get [the point cloud] into the position I wanted it to be in"* (P16).

In contrast, only one participant resented the navigation in *VR* noting *"it also needs a lot of [physical] space"* (P9). However, most participants (11) expressed particularly positive thoughts about *VR*. They found rotating the *view* with the controllers comfortable and strongly emphasized how moving in the scene enhanced their way of working with the point cloud: *"that you could walk around the object and change the perspective; that's very cool in VR"* (P11). Walking inside the point cloud to reach the target object was a common strategy: *"in VR, you could go into the point cloud and not just look at it from the outside"* (P12). Further, participants highlighted the depth perspective as intuitive and natural, mentioning no longer needing the view mode for adjustments.

5.2.2. Segmentation control

In terms of controlling the segmentation process, *tablet* received the most negative comments from 14 subjects, while *VR* was viewed positively by the vast majority (16), as was *desktop* (14).

On the *desktop*, using the keyboard was perceived as highly supportive. Participants felt to be more precise. Such, P2 particularly mentioned that *"I could switch fast using the shortcuts and be precise with the mouse"*, leading to a *"easier positioning of the object to cut it to size"* (P4). Although being assessed mostly positively, (2) participants reported rotating the objects felt cumbersome due to the need to move the mouse for repositioning.

Regarding the *tablet*, the majority of participants (14) criticized the segmentation controls. However, (2) found it most usable for coarse segmentation but reported the accuracy as problematic. An often mentioned disadvantage was missing shortcuts since *"if you want to work a lot and productively with it, shortcuts are important and time-saving"* (P14). The rotation of the point cloud was considered cumbersome, as participants reported having problems controlling the zoom and rotation functionality separately.

In *VR*, participants were having difficulties with the segmentation (i.e., the cuboid selection tool): six participants had difficulties *"estimating the depth in the selection tool"* (P3) resulting in being unsure *"which points I am selecting with the box"* (P10). They further mentioned that the physical demand was high (2), learning the shortcuts was demanding (2), and the selection did not feel precise (2). However, the interaction in *VR* was assessed well by the majority (16). They highlighted the handling of the cuboid selection (6), particularly the rotation (7), and the point cloud handling and zooming made a huge impression on many participants (6). Such P7 described the first moments in *VR*: *"the view handling, that was a wow moment; flipping, raising, especially with the dots when you zoom in, […] you could enlarge them and have that impression of being in a bigger world – this stood out the most."* In contrast to the negative comments, five participants felt more precise in *VR* than in the other devices. P17 stated having a *"little more control over the individual points"*. Further, they mentioned that the application felt natural and easy (3) and that the hotkeys were helpful (4) after getting used to them.

5.2.3. Familiarity

Eight participants stated that they were most familiar using *desktop*, leading to more intuitive navigation and the feeling of being more productive: *"[…] the computer is the more familiar tool, and I feel like I could be more productive with it"* (P11). Additionally, six participants commented on the unfamiliarity of *VR* and how they struggled to get used to the controls: *"I did not use VR a lot so far. Therefore I found the start way harder than working on the desktop"* (P6). For the *tablet* condition, only (4) participants reported familiarity, focusing on the intuitiveness of moving and rotating the objects with finger gestures similar to using a smartphone.

5.2.4. User satisfaction

The rating of the individual devices was often influenced by satisfaction exhibiting a large discrepancy between the device types. While *VR* received positive comments about satisfaction from 13 participants, and only six connoted this negatively, *desktop* received positive opinions from 7 participants, with only one commenting negatively. *Tablet* received the most negative feedback regarding satisfaction (10), while only 3 expressed their preference.

Regarding the *desktop*, the workload using the shortcuts and switching between modes were perceived as high effort and thus disliked by participants. They positively mentioned that the handling was efficient and easy to use, especially for longer sessions: *"it has the nice features of being lazy"* (P10).

The three participants commenting positively on the *tablet* mostly emphasized its ease of use. From their statements, it became clear that the others felt limited by the gesture control in their way of working, as it did not feel efficient to them. Such P11 said that *"VR required more effort in order to work with it, but it also has benefits – on the tablet, I don't have any benefits for work."* However, this participant saw an area of application for the tablet as a mobile device *"it's nice for travel, but no working device"*. Thinking about why he did not favor using the tablet, P8 noted: *"why should I use something that is neither efficient nor fun?"*

One-third of our participants (6) had a negative sentiment regarding *VR*. While participants reported problems like motion sickness (2) and eye strain(1), often the physical demand of *VR* was emphasized (*"it is also more strenuous and you had to focus more"* P12). However, a majority (13) positively assessed *VR*, often due to its immersive character (6): *"You have a play instinct; probably due to the immersion, you want to cut everything perfectly, with the others it felt more like work"* (P1). Also, the visual representation (2), like P7, who said that he *"first looked around a bit – that was really cool"*, and the feeling of the interaction (3) were commonly mentioned. P8 *"found the VR environment the coolest in terms of feeling; it was fun."* Overall most participants found *VR* pleasant to use.

5.2.5. Additional feedback

At the end of the interview, we asked participants about further improvement suggestions. Our participants wished for more segmentation functionalities. In particular, they desired tools for fine-grained segmentation, different shapes for selection (e.g., selection spheres or brushes), and a single-point selection functionally. Additionally, our participants suggested enhancing the arrangement of shortcuts when using *desktop* and ease switching between *view* and *segmentation* mode.

6. Discussion

The user study revealed various insights into point cloud segmentation between three different devices that might enable using applications more precisely through the device selection. In particular, we observed differences in user performance and segmentation correctness between the complexity of segmentation scenes. In the following, we discuss these findings in greater detail, outline limitations, and propose future research.

6.1. Efficiency between the devices

In our study, we observed that our participants' segmented objects faster using *desktop* and *tablet* than in *VR*. While this result that segmentation in *VR* is significantly slower than on *desktop* supports H_1, we could not find a significant difference to *tablet* in temporal comparison. We believe that there are two main reasons for this result. First, using a mouse and keyboard and moving fingers on a tablet requires users to move less. This was also reflected in the interview, where our participants mentioned that in contrast to *desktop* and *tablet*, *VR* demanded increased physical movement. The NASA TLX further supported the assessment by showing significant differences in physical demand. As it was significantly higher for *VR* than for *desktop* or *tablet*, we can accept H_2. We see another main factor in the difference in device familiarity of our participants. Since they stated to have more experience using *desktop* computers compared to *VR*, we suspect the familiar environment led them to focus directly on the segmentation task on *desktop*, while in *VR*, other factors might have influenced their time, like the general controls and immersion of the device although having performed a training. Although they were not as familiar with *tablet* as with *desktop* computers, there was still a huge difference to *VR*. Additionally, our participants mentioned in the interview that *VR* motivated them to put more effort into the segmentation, which might have led to longer editing times.

6.2. Scene complexity influences efficiency and effectivity

We further found that the complexity of the segmentation task influenced the TCT and segmentation correctness. More *complex* segmentation tasks led to higher TCTs than *simpler* ones. We further found that our participants were faster in completing the *simple* segmentation tasks on *desktop* computers compared to *VR*. For *complex* tasks, we did not observe significant differences between these devices. Regarding the correctness of our participants' segmentations, we found significant differences in the complexity of the segmentation task. The F1 score, indicating how many points were falsely added to the final segmentation of our participants, was lower for more *complex* segmentation tasks than *simpler* tasks. We believe that the complex segmentation tasks demanded higher effort from our participants, as intended when designing the scenes. For instance, our participants had to constantly change the view during segmentation due to occlusion. However, since we could not find a significant difference between the devices for the F1 score, we consequently have to reject H_3.

6.3. Users' device and context preferences

We assessed our participant's preferences regarding the different technology classes. A *desktop* computer, as well as *VR*, were preferred by our participants for the segmentation of point clouds over the *tablet*. During the interview, our participants mentioned that they appreciated the efficiency and familiarity of *desktop* computers. Regarding *VR*, they emphasized that immersing themselves into the *VR* environment allowed for a better view of the point cloud and enabled them to move around freely. Although our participants took longer compared to desktop computers or tablets, we received positive responses from most of our participants regarding their satisfaction when using *VR*. Our results showed that navigating the point cloud was significantly easier when using *VR* compared to *tablet* and *desktop*.

Although we could not find a significant difference in the segmentation correctness between the three devices, we see these findings as hints to different application scenarios for point cloud segmentation. *Desktop* solutions still enable a fast and familiar working style, potentially benefitting from a wide range of existing applications. Comparatively, VR might be superior for understanding scenes and displaying non-trivial occlusion environments due to its outstanding options to navigate the scene. Since we further found a high satisfaction using *VR*, we believe it might attract other user groups for such tasks. In the long perspective, we expect users to become more familiar with *VR* when such devices are used more often in the general population, which might influence the current advantage of *desktop* regarding temporal efficiency. Although our participants mostly declined the *tablet*, we found suggesting that it could be used as a mobile device when traveling.

Taken together, we can answer our research question: Both *desktop* and *tablet* were found to be more efficient for segmenting objects from point cloud images compared to *VR*, while we could not find a significant difference between them regarding their segmentation correctness. Further, we observed that the complexity does influence the effectiveness of a segmentation task. *desktop* and *tablet* temporally outperform *VR* in *simple* scenarios. However, *VR* offers ways to engage users during the segmentation process.

6.4. Limitations

We acknowledge the following limitations of our work.

6.4.1. Study objective and selection tool

The focus of our work is to compare *desktop, tablet, and VR* in terms of their suitability for the segmentation of *simple* and *complex* point clouds regarding efficiency and effectiveness (RQ). As such, our study does not involve comparing existing non-commercial and commercial applications, as it is out of the scope of this study.

We included one particular selection tool, a cuboid volumetric selection, as the base functionality for the segmentation task. Since the restricted form did not enable the direct selection of round shapes or single points, our comparison is also limited to this tool. Including further segmentation options, like a lasso or algorithmic segmentation, could influence the performance of the devices.

6.4.2. Object size

In our study, we used objects of similar size with a height of 12 centimeters. We anticipate that our findings would apply to objects of similar size. Significantly larger objects might influence the correctness of segmentation and task performance, as the size could affect the participants' usage of zoom and rotation functionalities. Participants may more easily detect larger objects, but removing all unwanted points from their potentially large surface could increase the effort required for an acceptable segmentation. Segmentation of much smaller objects deeply embedded within a point cloud could require an extensive search phase and negatively impact task completion time. However, these assumptions require further studies and empirical validation.

6.4.3. User interface

Since we developed *desktop* and *tablet* similar to existing applications, we limited the design of the *VR* application to the design of the user interface for comparability. Commonly used characteristics for menu navigation, like placing the menu static in the spatial environment, were thus not used. This design decision might have negatively influenced the participants' work in *VR*.

6.5. Future work

Since we found user preferences in *VR*, we believe that improving effectiveness in *VR* would benefit many wishing to use it for segmentation tasks. Since we found reason to believe that one major aspect of the *desktop's* high efficiency is due to its familiarity, we assume new perspectives using *VR* in the future. In future work, we plan to introduce additional tools for *VR*, including segmentation algorithms that can be controlled and adjusted by humans as human in the-loop applications. We see in it possible perspectives to combine the advantages of human recognition with the calculation speed of computers. Furthermore, we see perspectives in the spatial representation in *VR*, as was emphasized by our participants. Since missing data is a common problem with real-life recordings of point cloud data, we consider exploring the manual or semi-manually filling of such gaps in *VR*. Furthermore, it would be valuable to compare the devices, considering only participants with similar familiarity with each device. Involving participants who are equally experienced with desktop, tablet, and VR devices could provide more precise insights into each device's initial advantages and limitations in the context of point cloud segmentation.

Our findings show that the complexity of a scene impacts the efficiency and effectiveness of segmentation tasks. However, point cloud scenes can represent entire environments and present additional information through visual cues such as text annotations. Since whole environments, due to occlusion, are inherently more complex, we believe it's worth exploring how visual contextual cues can affect the perceived difficulty, considering variations in point cloud quality. Different user objectives within these environments might further influence the effectiveness of these visual cues in compensating for the complexity, which might be a further research direction to consider. Furthermore, considering the distinct immersive and perceptual characteristics of *desktop*, *tablet*, and *VR* devices, it would be valuable to investigate how visual cues should be designed for each device to compensate for any missing information caused by lower point cloud quality.

7. Conclusion

In this paper, we compared *desktop*, *tablet*, and a *VR*-HMD in a user study including 18 participants for segmenting objects from point cloud data. We examined whether the devices differ regarding various measures, including effectiveness and efficiency. Moreover, we investigated if the *complexity* of a segmentation task influences user performance. Our results show that *desktop* and *tablet* outperform *VR* in the task completion time (TCT), while we could not find significant differences between them for the segmentation correctness. While observing a significant difference for the TCT between the *simple* segmentation tasks, we could not measure a difference for *complex* scenes. However, we found that scene complexity influences segmentation correctness. We conclude that for segmenting objects from point cloud images, all devices, *desktop*, *tablet*, and *VR* are currently suitable for segmenting objects from point clouds. Subjective feedback indicates that *VR* engages users during the segmentation process and allows for a more natural view and adjustment of the point cloud, while *desktop* was often preferred due to its familiarity and temporal efficiency. Although *tablet* outperformed *VR* regarding the processing time, like *desktop*, it was rejected by our participants due to missing satisfaction but seen as an option when traveling.

Notes

1. Thingiverse, https://www.thingiverse.com/, last retrieved on July 20, 2023.
2. Blender, https://www.blender.org/, last retrieved on July 20, 2023.
3. Icons8, https://icons8.com/, last retrieved on July 20, 2023.
4. The source code and applications are available online (http://vrsegmentation.hcigroup.de).

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work is funded by the German Federal Ministry of Education and Research (01IS21068B).

ORCID

Carina Liebers http://orcid.org/0000-0002-5025-9649
Marvin Prochazka http://orcid.org/0000-0002-4415-0073
Niklas Pfützenreuter http://orcid.org/0000-0001-7290-3469
Jonathan Liebers http://orcid.org/0000-0002-6923-9066
Jonas Auda http://orcid.org/0000-0003-1326-1405
Uwe Gruenefeld http://orcid.org/0000-0002-5671-1640
Stefan Schneegass http://orcid.org/0000-0002-0132-4934

References

Agarwal, S., Auda, J., Schneegaß, S., & Beck, F. (2020). A design and application space for visualizing user sessions of virtual and mixed reality environments. In J. Krüger, M. Niessner, & J. Stückler (Eds.), *Vision, modeling, and visualization*. Eurographics Association. https://doi.org/10.2312/vmv.20201194

Argelaguet, F., & Andujar, C. (2013). A survey of 3D object selection techniques for virtual environments. *Computers & Graphics*, *37*(3), 121–136. https://doi.org/10.1016/j.cag.2012.12.003

Bacim, F., Kopper, R., & Bowman, D. A. (2013). Design and evaluation of 3D selection techniques based on progressive refinement. *International Journal of Human-Computer Studies*, *71*(7-8), 785–802. https://doi.org/10.1016/j.ijhcs.2013.03.003

Balakrishnan, R., Baudel, T., Kurtenbach, G., Fitzmaurice, G. (1997). The Rockin' mouse: Integral 3D manipulation on a plane. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (pp. 311–318).

Bangor, A., Kortum, P., & James, M. (2009). Determining what individual SUS determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, *4*(3), 114–123.

Barnefske, E., & Sternberg, H. (2022). Evaluating the quality of semantic segmented 3D point clouds. *Remote Sensing*, *14*(3), 446. https://doi.org/10.3390/rs14030446

Bérard, F., Ip, J., Benovoy, M., El-Shimy, D., Blum, J. R., & Cooperstock, J. R. (2009). Did "minority report" get it wrong?

Superiority of the mouse over 3D input devices in a 3D placement task. *Human-Computer Interaction – INTERACT 2009*, *5727*, 400–414. https://doi.org/10.1007/978-3-642-03658-3\textunderscore45

Besancon, L., Issartel, P., Ammi, M., & Isenberg, T. (2017). Hybrid tactile/tangible interaction for 3D data exploration. *IEEE Transactions on Visualization and Computer Graphics*, *23*(1), 881–890. https://doi.org/10.1109/TVCG.2016.2599217

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Brooke, J. (1996). SUS – A quick and dirty usability scale. *Usability Evaluation in Industry*, *189*(194), 4–7.

Bruder, G., Steinicke, F., & Nuchter, A. (2014). *Immersive point cloud virtual environments* [Paper presentation]. 2014 IEEE Symposium on 3D User Interfaces (3DUI) (pp. 161–162). IEEE. https://doi.org/10.1109/3DUI.2014.6798870

Bruder, G., Steinicke, F., & Sturzlinger, W. (2013a). *Effects of visual conflicts on 3D selection task performance in stereoscopic display environments* [Paper presentation]. 2013 IEEE Symposium on 3d User Interfaces (3DUI) (pp. 115–118). IEEE. https://doi.org/10.1109/3DUI.2013.6550207

Bruder, G., Steinicke, F., Sturzlinger, W. (2013b). To touch or not to touch? Comparing 2D touch and 3D mid-air interaction on stereoscopic tabletop surfaces. In *Proceedings of the 1st Symposium on Spatial User Interaction* (pp. 9–16). https://doi.org/10.1145/2491367.2491369

Dou, M., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S. R., Kowdle, A., Escolano, S. O., Rhemann, C., Kim, D., Taylor, J., Kohli, P., Tankovich, V., & Izadi, S. (2016). Fusion 4D: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics*, *35*(4), 1–13. https://doi.org/10.1145/2897824.2925969

Elmqvist, N., & Tsigas, P. (2008). A taxonomy of 3D occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics*, *14*(5), 1095–1109. https://doi.org/10.1109/TVCG.2008.59

Farmani, Y., & Teather, R. J. (2017). *Player performance with different input devices in virtual reality first-person shooter games* [Paper presentation]. SUI '17: Symposium on Spatial User Interaction (p. 165). https://doi.org/10.1145/3131277.3134361

Globa, A. A., Donn, M., & Ulchitskiy, O. A. (2016). Metrics for measuring complexity of geometric models. *Scientific Visualization*, *8*(5), 74–82.

Grossman, T., & Balakrishnan, R. (2006). *The design and evaluation of selection techniques for 3D volumetric displays* [Paper presentation]. Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (pp. 3–12). https://doi.org/10.1145/1166253.1166257

Hart, S. G. (1986). Task load index (NASA-TLX).

Hart, S. G. (2006). NASA-Task Load Index (NASA-TLX): 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *50*(9), 904–908. https://doi.org/10.1177/154193120605000909

Hoang, L., Lee, S.-H., Kwon, O.-H., & Kwon, K.-R. (2019). A deep learning method for 3D object classification using the wave kernel signature and a center point of the 3D-triangle mesh. *Electronics*, *8*(10), 1196. https://doi.org/10.3390/electronics8101196

Isenberg, T. (2011). Position paper: Touch interaction in scientific visualization. INRIA (pp. 24–27). https://hal.inria.fr/hal-00781512/

Jackson, B., Jelke, B., & Brown, G. (2018). *Yea big, yea high: A 3D user interface for surface selection by progressive refinement in virtual environments* [Paper presentation]. 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (pp. 320–326). IEEE. https://doi.org/10.1109/VR.2018.8447559

Jones, K. S., McIntyre, T. J., & Harris, D. J. (2020). Leap motion- and mouse-based target selection: Productivity, perceived comfort and fatigue, user preference, and perceived usability. *International Journal of Human–Computer Interaction*, *36*(7), 621–630. https://doi.org/10.1080/10447318.2019.1666511

Klein, T., Guéniat, F., Pastur, L., Vernier, F., & Isenberg, T. (2012). A design study of direct-touch interaction for exploratory 3D scientific visualization. *Computer Graphics Forum*, *31*(3pt3), 1225–1234. https://doi.org/10.1111/j.1467-8659.2012.03115.x

Koutsabasis, P., & Vogiatzidakis, P. (2019). Empirical research in mid-air interaction: A systematic review. *International Journal of Human–Computer Interaction*, *35*(18), 1747–1768. https://doi.org/10.1080/10447318.2019.1572352

Li, H., Zhang, X., Jaeger, M., Constant, T. (2010). *Segmentation of forest terrain laser scan data*. ACM. http://dl.acm.org/citation.cfm?id=1900179 https://doi.org/10.1145/1900179.1900188

Liu, Y., Zulfikar, I. E., Luiten, J., Dave, A., Ramanan, D., Leibe, B., … Leal-Taixé, L. (2021). Opening up open-world tracking. arXiv. https://arxiv.org/abs/2104.11221

Montano-Murillo, R. A., Nguyen, C., Kazi, R. H., Subramanian, S., DiVerdi, S., & Martinez-Plasencia, D. (2020). *Slicing-volume: Hybrid 3D/2D multi-target selection technique for dense virtual environments* [Paper presentation]. 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (pp. 53–62). IEEE. https://doi.org/10.1109/VR46266.2020.00023

Mossel, A., & Koessler, C. (2016). Large scale cut plane: An occlusion management technique for immersive dense 3D reconstructions. In D. Kranzlmüller & G. Klinker (Eds.), *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology* (pp. 201–210). ACM. https://doi.org/10.1145/2993369.2993384

Pearl, H., Swanson, H., & Horn, M. (2019). Coordi: A virtual reality application for reasoning about mathematics in three dimensions. In S. Brewster, G. Fitzpatrick, A. Cox, & V. Kostakos (Eds.), *Extended abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1–6). ACM. https://doi.org/10.1145/3290607.3312931

Pellerin, J., Caumon, G., Julio, C., Mejia-Herrera, P., & Botella, A. (2015). Elements for measuring the complexity of 3D structural models: Connectivity and geometry. *Computers & Geosciences*, *76*, 130–140. https://doi.org/10.1016/j.cageo.2015.01.002

Petford, J., Nacenta, M. A., & Gutwin, C. (2018). *Pointing all around you: Selection performance of mouse and ray-cast pointing in full-coverage displays* [Paper presentation]. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (pp. 1–14). https://doi.org/10.1145/3173574.3174107

Pierce, J. S., Forsberg, A. S., Conway, M. J., Hong, S., Zeleznik, R. C., & Mine, M. R. (1997). Image plane interaction techniques in 3D immersive environments. In A. van Dam (Ed.), *Proceedings of the 1997 Symposium on Interactive 3D Graphics – SI3D'97* (p. 39–ff). ACM Press. https://doi.org/10.1145/253284.253303

Ramkumar, A., Stappers, P. J., Niessen, W. J., Adebahr, S., Schimek-Jasch, T., Nestle, U., & Song, Y. (2017). Using GOMS and NASA-TLX to evaluate human–computer interaction process in interactive segmentation. *International Journal of Human–Computer Interaction*, *33*(2), 123–134. https://doi.org/10.1080/10447318.2016.1220729

Reichert, J., Backes, A. R., Schubert, P., & Wilke, T. (2017). The power of 3D fractal dimensions for comparative shape and structural complexity analyses of irregularly shaped organisms. *Methods in Ecology and Evolution*, *8*(12), 1650–1658. https://doi.org/10.1111/2041-210X.12829

Ridha-Mahfoudhi, H., & Dang, N.-T. (2019). Real time point cloud self-avatar with a single RGB-D camera. In T. Trescak (Eds.), *25th ACM Symposium on Virtual Reality Software and Technology* (pp. 1–2). ACM. https://doi.org/10.1145/3359996.3365041

Stets, J. D., Sun, Y., Corning, W., & Greenwald, S. W. (2017). Visualization and labeling of point clouds in virtual reality. In D. Gutierrez & H. Huang (Eds.), *Siggraph Asia 2017 posters* (pp. 1–2). ACM. https://doi.org/10.1145/3145690.3145729

Stürzlinger, W., Kitamura, Y., & Coquillart, S. (2007). *Exploring the effects of environment density and target visibility on object selection in 3D virtual environments* [Paper presentation]. 2007 IEEE Symposium on 3D User Interfaces. https://doi.org/10.1109/3DUI.2007.340783

Teather, R. J., & Stuerzlinger, W. (2008). *Assessing the effects of orientation and device on 3D positioning* [Paper presentation]. 2008 IEEE Virtual Reality Conference (pp. 293–294). https://doi.org/10.1109/VR.2008.4480807

Teather, R. J., & Stuerzlinger, W. (2011). *Pointing at 3D targets in a stereo head-tracked virtual environment* [Paper presentation]. 2011 IEEE Symposium on 3D User Interfaces (3DUI) (pp. 87–94). https://doi.org/10.1109/3DUI.2011.5759222

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). *Domain randomization for transferring deep neural networks from simulation to the real world* [Paper presentation]. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 23–30). https://doi.org/10.1109/IROS.2017.8202133

Tobin, J., Zaremba, W., & Abbeel, P. (2018). *Domain randomization and generative models for robotic grasping* [Paper presentation]. IEEE/RSJ International Conference on Intelligent Robots and Systems. http://ieeexplore.ieee.org/servlet/opac?punumber=8574473 https://doi.org/10.1109/IROS.2018.8593933

Vanacken, L., Grossman, T., & Coninx, K. (2009). Multimodal selection techniques for dense and occluded 3D virtual environments. *International Journal of Human-Computer Studies*, *67*(3), 237–255. https://doi.org/10.1016/j.ijhcs.2008.09.001

Virtanen, J.-P., Daniel, S., Turppa, T., Zhu, L., Julin, A., Hyyppä, H., & Hyyppä, J. (2020). Interactive dense point clouds in a game engine. *ISPRS Journal of Photogrammetry and Remote Sensing*, *163*, 375–389. https://doi.org/10.1016/j.isprsjprs.2020.03.007

Vuibert, V., Stuerzlinger, W., Cooperstock, J. R. (2015). *Evaluation of docking task performance using mid-air interaction techniques* [Paper presentation]. Proceedings of the 3rd ACM Symposium on Spatial User Interaction (pp. 44–52). https://doi.org/10.1145/2788940.2788950

Whitlock, M., Smart, S., & Szafir, D. A. (2020). *Graphical perception for immersive analytics* [Paper presentation]. 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (pp. 616–625). IEEE. https://doi.org/10.1109/VR46266.2020.00084

Wirth, F., Quchl, J., Ota, J., & Stiller, C. (2019). *Pointatme: Efficient 3D point cloud labeling in virtual reality* [Paper presentation]. 2019 IEEE Intelligent Vehicles Symposium (IV). https://ieeexplore.ieee.org/servlet/opac?punumber=8792328 https://doi.org/10.1109/IVS.2019.8814115

Wobbrock, J. O., Findlater, L., Gergle, D., & Higgins, J. J. (2011). *The aligned rank transform for nonparametric factorial analyses using only ANOVA procedures* [Paper presentation]. Proceedings of the Sigchi Conference on Human Factors in Computing Systems (pp. 143–146). Association for Computing Machinery. https://doi.org/10.1145/1978942.1978963

Xie, Y., Tian, J., & Zhu, X. X. (2020). Linking points with labels in 3D: A review of point cloud semantic segmentation. *IEEE Geoscience and Remote Sensing Magazine*, *8*(4), 38–59. https://doi.org/10.1109/MGRS.2019.2937630

Yee, K.-P. (2004). *Two-handed interaction on a tablet display* [Paper presentation]. Chi '04 Extended Abstracts on Human Factors in Computing Systems, Computing Machinery (pp. 1493–1496). Association for Computing Machinery. https://doi.org/10.1145/985921.986098

Yu, L., Efstathiou, K., Isenberg, P., & Isenberg, T. (2012). Efficient structure-aware selection techniques for 3D point cloud visualizations with 2D of input. *IEEE Transactions on Visualization and Computer Graphics*, *18*(12), 2245–2254. https://doi.org/10.1109/TVCG.2012.217

Yu, L., Efstathiou, K., Isenberg, P., & Isenberg, T. (2016). Cast: Effective and efficient user interaction for context-aware selection in 3D particle clouds. *IEEE Transactions on Visualization and Computer Graphics*, *22*(1), 886–895. https://doi.org/10.1109/TVCG.2015.2467202

Zhang, Q., Kim, C.-H., & Byun, H. W. (2022). Multi-finger-based arbitrary region-of-interest selection in virtual reality. *International Journal of Human–Computer Interaction*, *0*(0), 1–15. https://doi.org/10.1080/10447318.2022.2108666

About the authors

Carina Liebers is a PhD student in the Human-Computer Interaction Group at the University of Duisburg-Essen. Her focus of research lies in human-assisted support for reinforcement learning methods. She attained her master's degree in computer science from the University of Duisburg-Essen in Germany.

Marvin Prochazka obtained a bachelor's degree in computer science at the University of Duisburg-Essen. He is currently studying for his master's degree and is a research assistant at the Human-Computer Interaction Group. His research interest lies mainly within the field of usability and virtual reality.

Niklas Pfützenreuter obtained a bachelor's degree in computer science at the University of Duisburg-Essen. He is currently a master's student and a research assistant at the Human-Computer Interaction Group. His research interest lies mainly within the field of machine learning and virtual reality.

Jonathan Liebers obtained a bachelor's and master's degree in computer science at the University of Duisburg-Essen. He is currently a PhD student at the Human-Computer Interaction Group, University of Duisburg-Essen. His research interest lies mainly within the field of usable security and implicit identification.

Jonas Auda is a postdoctoral researcher in Human-Computer Interaction at the University of Duisburg-Essen. His research encompasses various areas, including interactions with virtual and augmented realities, brain-computer interfaces (BCIs), cross-reality systems, and human-drone interaction. His dissertation specifically focused on investigating novel interaction opportunities with virtual realities.

Uwe Gruenefeld is a postdoc researcher in human-computer interaction at the University of Duisburg-Essen, Germany. He is fascinated by a wide range of topics around AR and VR. His research has mainly focused on visualizing out-of-view objects, passive haptics, usable security, and cross-reality systems.

Stefan Schneegass is a professor of human-computer interaction at the University of Duisburg-Essen. His research interests include the crossroads of human-computer interaction and ubiquitous computing, particularly the development of implicit authentication mechanisms. Schneegass received a PhD in computer science from the University of Stuttgart, Germany.