

Inside looking out or outside looking in? An evaluation of visualisation modalities to support the creation of a Substitutional Virtual Environment

Jose F. Garcia
University of Portsmouth
jose.garcia@port.ac.uk

Adalberto L. Simeone
KU Leuven
adalberto.simeone@kuleuven.be

Matthew Higgins
University of Portsmouth
matthew.higgings@port.ac.uk

Wendy Powell
University of Portsmouth
wendy.powell@port.ac.uk

Vaughan Powell
University of Portsmouth
vaughan.powell@port.ac.uk

ABSTRACT

Current Virtual Reality systems only allow users to draw a rectangular perimeter to mark the room-scale area they intend to use. Domestic environments can include furniture and other obstacles that hinder the ease with which users can naturally walk. By leveraging the benefits of passive haptics, users can match physical objects with virtual counterparts, to create substitutional environments. In this paper we explore two visualisation modalities to aid in the creation of a coarse virtual representation of the physical environment, by marking out the volumes of space where physical obstacles are located, to support the substitution process. Our study investigates whether this process is better supported by an inside-looking-out 3D User Interface (that is, viewing the outside world while immersed in Virtual Reality) or from an outside-looking-in one (while viewing the Virtual Environment through an external device, such as a tablet). Results show that the immersive option resulted in better accuracy and was the one with the highest overall preference ratings.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; **Interaction techniques**;

KEYWORDS

Virtual Reality, Virtual Environments, Substitutional Reality

1 INTRODUCTION

Virtual Reality (VR) can now be considered a consumer grade technology. The HTC Vive, a Head-Mounted Display (HMD) released in 2016, allows users to experience natural-walking in a room in VR. The challenge current research is facing lies in the mismatch between Virtual Environments (VEs) portrayed in the chosen VR experience and the Physical Environment (PE) where the user is located [16].

In a domestic PE, such as the user’s own living room, the presence of furniture poses limits on the ease and safety of natural walking in the VE [15].

In this context, current VR systems do not provide users with methods to customise the layout of their room-scale area. The *de-facto* standard technique consists in displaying a bounding grid which becomes visible when the user is

in proximity of the boundaries of the tracking space. This system (known as *Chaperone* in the HTC Vive or *Guardian* in the Oculus Rift) uses measures taken by the user during a set-up phase. Users have the choice of drawing a rectangular perimeter that can either include any large furniture present in their desired room-scale area, or fit only the part of it which is completely empty. For example, including a large sofa within the perimeter of the room-scale area will increase the space available. However, users must be aware of the presence of the obstacle when they move. In this work, we explore two visualisation modalities that with available technology allow end-users to define the layout of the physical environment in which they are located, and support them in the creation of a *substitutional* environment

Defining an *intermediate* environment, consisting in a coarse representation of the volumes of the furniture and other objects in the user’s PE, can help users maximise the area available for the VR experience. To some extents, a similar process already happens in VR games such as *Job Simulator* and *Rick and Morty: Virtual Rick-ality*, which select the design of the VE that best fits within the size of the user’s tracking space, among different versions. However, these games perform their customisation of the VE based only on the dimensions of the room-scale area. In this work we focus on the scenario in which the design of the Virtual Environment where the VR experience takes place is informed by the intermediate environment previously defined by the user. We envision a near-future where, by leveraging the potential of passive haptics, the believability of the experience can be improved by using physical objects as proxies for virtual counterparts [10, 16]. In this scenario, the user creates the substitutional environment by assigning the assets provided by the designers of the VR experience, to the identified volumes of the intermediate environment.

Related research has shown that substitutional environments can be automatically generated by processing depth camera input [14, 20]. In this research we focused on exploring how best to support users in this alternative approach, which delegates the customisation of the VE to the end-user as a way of addressing the potential of repetitiveness of procedural content generation [18].

We developed a system allowing users to block out volumes of space occupied by objects or other obstacles. Users draw either a 4 or 8-point bounding volume. Successively, users decide how to replace this intermediate environment with the actual assets provided by the designers of the VR experience. This process poses the challenge of executing the task in the real PE, but receiving feedback only in the VE. We compared two different visualisation modalities: 1) *inside looking out*, that is, performing the task while wearing an HMD where a video feed of the PE is visible in the VR view; 2) *outside looking in*, that is, performing the task without an HMD, and while viewing feedback from the virtual world on a hand-held tablet.

The study we conducted provides insights on how these two different visualisation modalities can support users in creating substitutional environments. Results indicate that the HMD-based visualisation was more accurate in allowing users to mark volumes that fitted to the actual objects, while being slower than the tablet in terms of task completion times. The finding that users preferred adapting the VE to their physical surroundings while immersed in VR, instead of in the real world, highlights the need for further research into how best to support this activity.

2 RELATED WORK

Our research investigates the role of users as *co-creators* of the VE [2]. This work focuses on the growing area of VR “room-scale” applications where users move within a 1:1 tracked area, in which it is not uncommon to encounter furniture and other obstacles typical of a domestic environment. To overcome these issues, past research on “Substitutional Reality” (SR) investigated the idea of matching physical objects with virtual ones under the assumption that an explicit mismatch exists in the pairing of objects [16] or surfaces [15]. Other research has explored the automatic reconstruction of the physical environment [20] using a set of rules based on the principles of SR. *Reality Skins* by Shapira and Freedman [14] uses a blueprint, containing the set of objects and materials to use, provided by the designers to assist the substitution process. The system processes an environment scanned with a depth sensor to calculate the transformations necessary to find the best fits for each object in the scene. Garcia and Simeone proposed the use of a recommender system to facilitate the selection of virtual objects to substitute those present in their PE [4]. Valkov and Flagge studied different techniques for transitioning between a virtual replica of the user’s surroundings, to a substitutional environment [23].

In broader terms, the field of 3D object detection is an area of extensive research. Although progress continues to be made in this research field, with recognition rates achieving or surpassing 95% on some datasets, it requires specialist hardware and significant amounts of time to attain the best results [5]. Research in 3D object detection does not focus on how to use this information as the basis to build something else, which is the focus of this research.

Delegating the design of the VE to procedural content generation techniques is often necessary to create vast amounts of content in the video game industry [7, 17]. However one of the main critiques to this approach is the potential for repetitiveness [18]. Indeed, the games “Job Simulator” and “Rick and Morty: Virtual Rick-ality” match the VE design which best fits within the detected available space, between a few pre-defined ones.

The idea of substituting or using elements of the real world in a virtual experience has been also adopted in other contexts. For example, an Augmented Reality system by Hettiarachchi and Wigdor [8] overlays virtual objects over the detected area of the physical objects. *Shifty* by Zenner and Krüger [26] investigated the concept of shifting a weight in order to better support the illusion of manipulating objects of different forms or weights. *CarVR* by Hock et al. [9] uses the forces acting on a moving vehicle to corroborate the experience of playing a mobile cockpit-base VR game.

Creating a VE from scratch was the topic of the 2013 edition of the annual contest held at the *IEEE Conference on 3D User Interfaces*. The proceedings contain several examples of immersive level editing systems [1, 6, 11, 22, 24]. Successively, Wang and Lindeman have studied the transition between different interface modalities (e.g., non-occlusive HMD and tablet) in a level-editing task [25]. Mine et al. describe the development of a hand-held device allowing immersive modeling in various visualization modalities, such as a CAVE, a large display, and both stereo and non-stereo displays, but not a HMD [12]. The *VR Editor* in *Unreal Engine 4* provides an immersive 3D User Interface allowing users to author 3D environments for both VR and non-VR applications [3].

The idea of creating a window with which to view the “other” world builds on the concept of the *Magic Lens*, virtual [21] or tangible surfaces [19] that are typically used to provide additional details or alternative visualisations of the data normally displayed. In our system, we implemented a virtual lens in the HMD visualisation, which shows the outside world through the embedded video camera, and a tangible one, which uses a Microsoft Surface Pro 4 to provide a visualisation of the virtual world.

3 SYSTEM

We created an interaction technique that enables users to define the volumes of virtual space occupied by physical objects, and in a second phase, replace and customise these volumes with contextually appropriate virtual objects (see Figure 1). We designed the system around the idea of marking volumes as the size of an object to be the best predictor of whether the substitutive object would match with the physical proxy. Objects of mismatching sizes negatively affect the experience [16].

This study investigates whether this activity is better performed *inside looking out*, while immersed in VR with a view on the outside world, or conversely, *outside looking in*, in the real world, with a window on the virtual world. Thus, we implemented two visualisations that share the same interaction

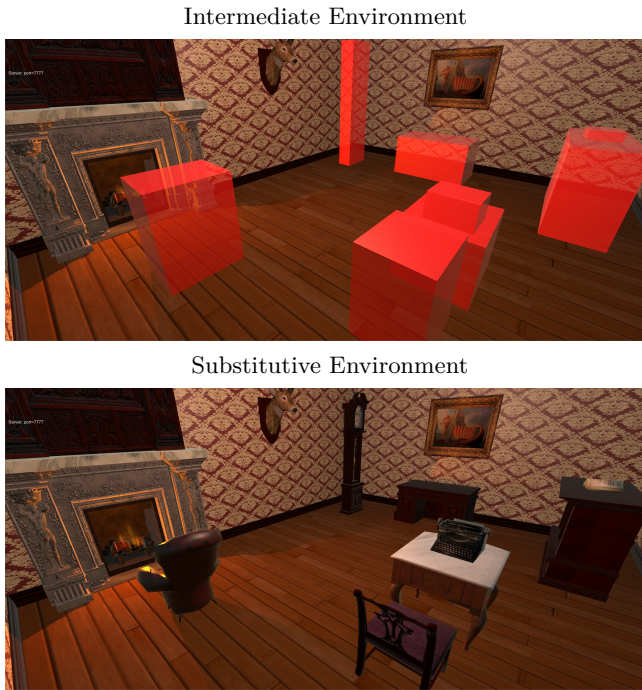


Figure 1: The figure shows the intermediate environment resulting from drawing the volumes (above) and the result of substituting these with the assets provided (below).

technique. The first uses a HTC Vive to immerse the user in the VE they will adapt. Using the Vive’s onboard camera, a view of the outside world is shown in the VE, and provides the user with outside spatial awareness. However, due to the Vive having only a single optically non-aligned camera (located in the lower half of the Vive), the view provided is monoscopic. The second visualisation is the opposite: without wearing a headset, users perform the same interactions while viewing a monoscopic rendering of the virtual world on a tablet.

The interaction technique is structured into four modalities, and requires a single HTC Vive controller. A simple state machine allows users to cycle through these modalities by pressing the controller’s trackpad button. These are: 1) Volume drawing; 2) Object spawning; 3) Manipulation; 4) Scaling. Figure 2 shows how the actions are mapped to the Vive Controller.

3.1 Tablet instrumentation

Our system is aimed at end-users. Therefore, in order to have an equal comparison between the HMD and the Tablet system, we only used commercially available devices for both. For ease of development, we used the Microsoft Surface Pro 4, as it allows to run standard Windows applications. However,

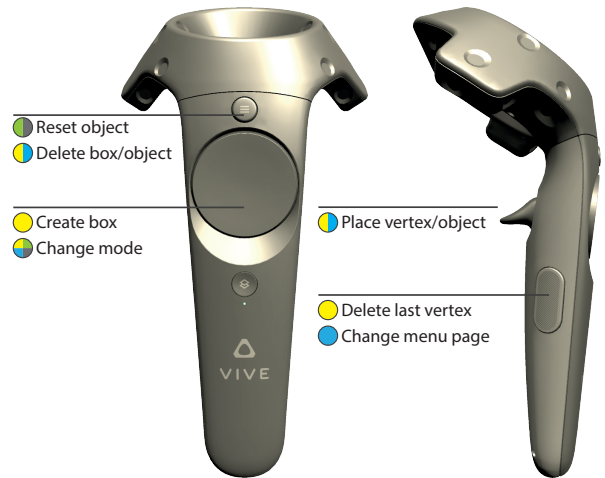


Figure 2: The figure shows the control mappings used in our system. The colours correspond to the four modalities: *Volume drawing* (Yellow); *Object spawning* (Cyan); *Manipulation* (Green); *Scaling* (Grey). A subdivided circle indicates that the action is contextually dependent on the modality.

the major 3D game engines can target other devices such as iOS and Android tablets.

The main issue we faced is that, with present-day technology, tablets cannot be tracked with the same ease of the HTC Vive’s headset and controllers. Due to the constraint of only using commercial devices, we felt that instrumenting the tablet with IR markers was not feasible, as the equipment necessary to track objects in this way is unlikely to be found in an average home. Another possibility was to attach the Vive controller and the tablet together, in order to track the tablet using the controller via a custom-designed mount. Aside from access to a 3D printer, this presents two other problems. First, the tablet risks occluding the controller, causing loss of tracking. Second, depending on the form factor of the tablet, if the two are attached together, the risk of the tablet colliding with physical objects increases, as the system requires users to draw volumes around these objects. Users might try to steer clear of potential collisions, thus decreasing the potential accuracy of the tablet.

We thus decided to keep the tablet and the controller separated. Users would hold the tablet with their non-dominant hand, and the controller in the dominant hand. Therefore, the position and orientation of the camera used to display the view of the VE in the tablet references the values of the controller the user is holding. However, moving the tablet does not change the view, as it is the controller that is responsible of the view displayed. This allows the user to be as accurate as possible and since no interaction happens on the tablet itself (as it is used for reference only) we decided this to be a good compromise between potential accuracy and ease of instrumentation. It is conceivable that the future release of

standalone trackers or improvements in inside-out tracking for tablets will improve the usability of this configuration.

3.2 Volume drawing

In our system, the process of adapting a VE to the physical surroundings requires that the user first surveys the layout of the room and marks the volumes of space occupied by physical objects accordingly (see Figure 3).

This phase is necessary as the virtual “output” world will be initially empty, and this intermediate environment provides a frame of reference for the subsequent substitution phase. To support this task we designed an interaction technique that allows users to draw vertices of an oriented bounding box, by pressing the Vive controller’s trigger. A transparent sphere is shown in proximity of the controller to indicate where the next vertex will be placed after pressing the trigger (see Figure 4).

The technique allows users to create a bounding box by placing four or eight points. Although it is possible to draw a box with less points (i.e. only the two opposing corners), we felt that drawing the perimeter was more intuitive and could mitigate the impact of incorrect placements. These bounding boxes are instantiated by placing the appropriate number of points and then pressing the trackpad button (which in this case will not cause the system to cycle modalities). Since users will not be able to place the vertices in such a way that they would form a perfect rectangle with orthogonal sides, the algorithm works by identifying the sides that form the perimeter of a regular convex polygon. By iterating over the set of vertices, it identifies those pairs whose sides would form an angle close to 90°. Once the four sides are found, the direction of the two parallel pairs are averaged together to form the opposing sides of the new box. The height is obtained by averaging the y -coordinates of the vertices.

Since the largest objects in a domestic environment are placed on the floor, the 4-point technique requires drawing only the four vertices of the perimeter of the topmost face of the box. The appropriate height such that it reaches the floor of the VE will then be calculated, using the averaged height of the four points as its maximum. The 8-point technique is aimed at smaller objects. It requires the user to draw the perimeters of both the top and the bottom face of the box. However, in this case, the bottom vertices are used only to determine the height.

3.3 Object Spawning

Defining a volume tells the software the volume in the PE that a physical object, such as a chair or table, is occupying. Newly created volumes will be rendered as red transparent boxes (see Figure 1). Users can then substitute them with the actual assets provided by the system. Each asset was re-scaled from its original size and stored as a unit cube model (in the Unity reference system, where 1 unit equals to 1 m), in such a way that the model is completely contained within. This allows the system to easily fit the chosen asset inside the bounds of the user-created box.

In the system, we have included three sets of Victorian-themed assets representing objects appropriate to the environment we used in the study, a Victorian living room. The system provides a set of six larger furniture items (a bookcase, a writing desk, two tables, a sofa, and a piano), a set of six different types of chairs and armchairs, and a set of six smaller objects (a phonograph, a typewriter, a book, a vase, a candelabrum, and a mirror). Each set is browsable through a radial menu which displays six objects at a time. Users select assets through the Vive Controller’s trackpad, which is mapped to the different sections of the radial menu. Users can switch sets of objects by pressing the grip button. The trigger button allows users to place the currently highlighted object by pointing at a box (causing it to become highlighted). When users do so, the box is hidden and the object appears in its place. The last action can be undone by pressing the menu button, which will restore the box associated to it. Pressing it again will remove the box entirely.

3.4 Manipulation

Once an object is placed in the VE, users can manipulate and scale it, in their respective modalities. Users can manipulate an object (after being placed) by holding the trigger button, causing the reference box to reappear. Objects can be rotated through the trackpad. Pressing it in correspondence of the four cardinal points causes the asset to alternate between different combinations of orientation and scale. Since every asset is stored with a pre-determined front-facing side (if appropriate), this feature allows users to choose the one most suitable to the environment. For example, in the case of a bookcase, pressing the four locations on the trackpad will cause the bookcase to resize and reorient itself in order to have the books facing one of these four directions. The orientation can be reset by pressing the menu button.

Users can scale objects in case of a mismatch between the size of the box and the size of the asset. For example, the open piano asset is completely inscribed in the unit cube it is stored in. If a user creates a box matching the size of a desk, the piano will be scaled accordingly and the upper part of the box will be a tangential plane for the model. In this case, users may wish to match the height of the piano keys to the height of the box created around a physical desk, thus making the virtual object more accurate in terms of height (but rendering the lid immaterial). Scaling is performed by holding the trigger and moving the controller. The location of the 3D controller when the trigger is pressed is taken as the reference point. Moving it horizontally or vertically affects its width or height relative to the user’s viewpoint, respectively. All models have their pivot point placed in the center of their bottom plane. Pressing the menu button allows users to reset the model to its original size (fitting inside the user-created box).

3.5 Differences between the two visualisations

In the VR visualisation, users view a 3D representation of the controller they are holding. After pilot testing we decided



Figure 3: Above, the sequence of actions necessary to identify a physical object and substitute it with a mismatching virtual object, as shown from the HMD-based interface. In (a), the white box is chosen for substitution; successively the user draws eight vertices around the physical corners of the box through the tracked Vive controller (b); a box is fitted within these eight vertices (c); finally, the chosen object is fitted within its bounds (d).

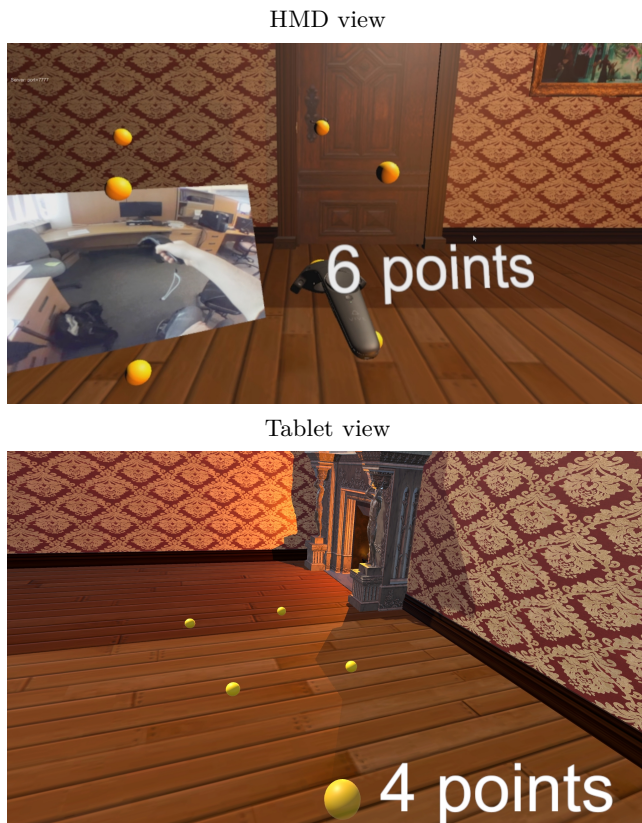


Figure 4: The figure shows the difference between the two views. Above, the 3D model of the Controller is visible. A window displaying the Vive camera feed is shown next to the controller. The point indicator floats above the trackpad. Below, in the tablet view, the camera is placed above the Vive controller in order not to obstruct the view and the point indicator is moved to the lower right corner.

to place the 2D window of the camera feed to the left of the controller, rather than in a fixed 2D overlay. In this way, the monoscopic window follows the controller at all times and is

placed at the same depth that the controller is currently at. The window constantly reorients itself in order to face the viewer perpendicularly (see Figure 4).

The behaviours previously described have been implemented in the exact same way on the tablet. The only difference is that, in the tablet, the viewpoint of the camera is placed on top of the controller, in order not to obstruct the view with its model in the foreground (see Figure 4). The textual indicator for the number of vertices is placed in the lower-right corner of the screen.

4 STUDY

The goal of the study was to evaluate whether the task of adapting a VE to the physical environment is better performed “inside” while immersed in VR, or from the “outside”, through a portable display. We were interested in evaluating whether these two visualisation modalities would provide different levels of performance, accuracy, and subjective preference. We thus designed a within-subjects study where participants performed this task while experiencing both visualisations in counter-balanced order.

4.1 Apparatus

As introduced, we used the HTC Vive for the VR condition, and a Microsoft Surface 4 (i5 with 8GB of RAM) tablet for the other condition. The test application was developed in Unity 4.5. The tablet visualisation was implemented by mirroring the Vive controller’s input from the server machine to the client application running on the tablet, by relying on Unity’s networking features while running on a dedicated router. The study took place in our VR laboratory, a 6.3 m by 4 m area.

4.2 Participants

Seventeen participants (10 male, 7 female) aged 19-29 (M 21.35, SD 2.64) took part in the study. Our participants self-reported having average experience with VR technologies (M 3.82, SD 1.55) and high experience with computer games in general (M 6.12, SD 0.99), in a scale where 1 represented “little to no experience” and 7 “very high experience”. Each participant was compensated by a £10

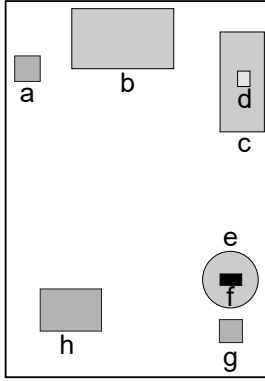


Figure 5: In clockwise order, the layout of the physical objects used in the experiment: a high stool (a); a desk (b); a standing desk (c); a book (d); a round table (e), a box (f), a chair (g); a couch (h). The outer outline represents the limits of the tracked space.

voucher for their time. Each study session took approximately 60 minutes.

4.3 Task

We furnished the lab with various items, as depicted in Figure 5. We placed a total of eight physical objects, six of which were larger furniture items (such as desks or chairs); the remaining two were smaller objects, a box and a book. The task itself was divided in two stages: 1) volume drawing; 2) object spawning. A floating cube in the room provided participants with a way to start the task and the data recording by intersecting it with their controller, causing it to disappear. For the first stage, we instructed participants to first create all the box volumes. Once the system registered a total of eight boxes, the floating cube reappeared in the room. Intersecting it a second time caused the system to begin the next one.

The second stage concerned the substitution of the boxes with the actual assets we provided. We told participants that they did not necessarily have to substitute objects with virtual counterparts having the same affordances, e.g. replacing the physical modern desk with a Victorian desk. There was no time limit and participants were free to take as long as they wished. Once participants placed all of the eight models, the floating cube appeared once more. Intersecting it one last time completed the task and stopped the collection of data.

4.4 Procedure

Participants received a written explanation of the study and provided informed consent. Before starting, they watched a training video showing how the interaction technique worked, using the Vive controller as a reference. After filling a demographics questionnaire, the experimenter asked the participant to perform all actions they had seen in the video, and explained the differences with the tablet version. The order in which the devices were presented was counterbalanced. After completing the task with each device, participants filled an

unweighed NASA-TLX questionnaire and another custom questionnaire. Successively, they performed the task with the remaining device. Finally we interviewed them following a semi-scripted set of questions based on our observations.

4.5 Analysis

In order to evaluate whether the two modalities lead to a difference in the way users interacted with them, we logged the sequence of actions they performed (e.g., change of modality, manipulation, object spawning, etc.), and the number of errors they made (vertices placed and then removed). To analyse the accuracy of the user-created volumes, we recorded the size of the volume, due to the links between the volume of the VR object and believability of the experience [16]. We used those to compare the difference in terms of volume to a set of baseline measurements obtained manually and as accurately as possible.

5 RESULTS

We performed a repeated-measures ANOVA with the type of *device* as independent variable, and the metrics we introduced in the previous section as the dependent variables. For each participant we recorded one set of data for each of the two devices. After finishing the two stages of the task with each device, we asked participants to fill the NASA-TLX questionnaire and another custom questionnaire on their interaction experience.

Overall, there was a significant effect of the *device* ($F_{1,186} 11.6, p < 0.01$) used when drawing the volumes in the first stage. The tablet was faster ($M 160.73s, SD 55.58s$) than the HMD ($M 276.73s, SD 139.90s$). No differences in terms of task completion times were found in the second stage ($p 0.9$, HMD: $M 260.05s, SD 135.96s$; Tablet: $M 254.37s, SD 136.53s$).

5.1 Accuracy

We found a significant effect of the *device* ($F_{1,186} 6.34, p 0.02$) used to create the volumes of the larger furniture items in the first stage, when compared to their respective reference volumes. The volumes created with the HMD are closer to the actual size of the physical object ($M 99.47\%, SD 42.61\%$), whereas those created with the tablet are larger ($M 115.65\%, SD 49.83\%$). The slightly smaller volumes resulting from using the HMD can be explained as an artefact of averaging the y -coordinate of the four user-placed vertices to obtain the final height of the box. The analysis of the volume ratios of smaller objects created with the 8-point technique did not indicate any significant difference ($p 0.12$, HMD: $M 114.75\%, SD 113.15\%$, Tablet: $M 83.57\%, SD 54.19\%$).

5.2 Interaction Technique

We also analysed the usage of the 3D User Interface (3DUI) across the two devices. As introduced, the 3DUI has been implemented in the two devices to be as similar as possible. The device used did not significantly affect ($p 0.45$) the average

number of errors (i.e. placing a vertex and then removing it) made in the first stage: the HMD-based technique accounted for an average of 1.41 errors (SD 2.65) per session (out of an average of 48.41, SD 19.71 markers placed), while the tablet resulted in 0.65 errors (SD 1.00) out of 42.71 markers placed.

We analysed the proportion of time spent in each of the different modalities during the second stage, but did not find any significant difference (p 0.90). The highest proportion of time was spent in the *Object Spawning* mode (51-58%), then in the *Manipulation* (20-21%) and *Scaling* modes (18-23%).

In the questionnaires, we also asked participants to rate how well they felt each device supported some of the fundamental tasks of 3D Interaction: *rotation*, and *scaling*. The HMD was found to provide significantly better support for the task of scaling ($F_{1,32}$ 7.67, $p < 0.01$; HMD: M 6.65, SD 0.70; Tablet: M 5.71, SD 1.21).

5.3 Questionnaires

Immediately after completing the two stages of the task with each device, we asked participants to rate the interaction technique they just used with an unweighed NASA-TLX questionnaire (using a scale from 1 to 100, where lower is better) and a custom one. The *Overall* score was not affected by the device used (p 0.72, HMD: M 43.78, SD 15.51; Tablet: M 45.67, SD 13.74). However, analysing the individual scales paints a clearer picture. In terms of *performance* the HMD technique received a better score ($F_{1,28}$ 6.1, p 0.02, M 26.00, SD 19.84) than the tablet technique (M 46.00, SD 24.29). Our interviews confirmed that participants felt more effective while using the HMD technique.

In our custom questionnaire, we asked participants to rate on a scale 1 to 7 (with seven being the higher/better score) various measures such as: whether they felt more *present* in the virtual world, or *aware* of their physical surrounding; how *difficult*, *accurate*, *easy to learn* the technique felt; and finally, give each technique an overall *preference* rating. In the results (see Figure 6) the HMD-based technique was reported to be easier to learn ($F_{1,32}$ 8.42, $p < 0.01$): (M 5.88, SD 0.70) as opposed to (M 4.82, SD 1.33) for the tablet. Not surprisingly, the HMD provided significantly better presence in the VE ($F_{1,32}$ 24.34, $p < 0.01$, M 6.65, SD 0.61) than the tablet (M 4.53, SD 1.66), while the opposite is true in terms of awareness of the physical surroundings for the tablet (M 6.18, SD 1.38 to M 4.65, SD 1.73). However, participants rated the HMD-based technique significantly higher than the Tablet technique in terms of personal preference ($F_{1,32}$ 14.02, $p < 0.01$, M 6.35, SD 0.86 to M 4.76, SD 1.52).

6 DISCUSSION

In this paper we sought to evaluate whether the task of adapting a set of virtual assets to the layout of their physical surroundings is best performed *inside*, while immersed in VR, or *outside*, in the real world. The results arising from our

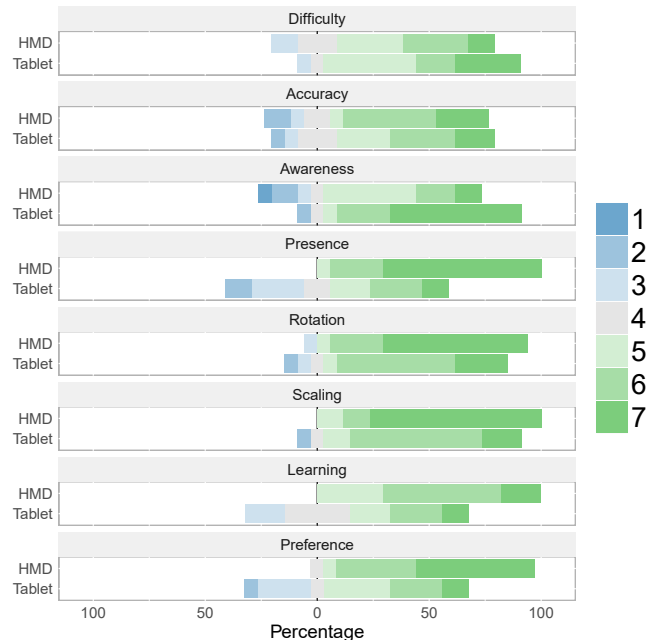


Figure 6: The scores in our custom questionnaires grouped by the device used, from 1 to 7, where the higher the better.

research can provide insights on how to improve the process of customising and editing a VE.

In summary, participants preferred creating the intermediate VE (see Figure 1) while immersed in VR. Although they were slower while using the HMD, they were able to create a more accurate representation of their physical surroundings. The immersive VR interface was also the technique rated the highest by participants. We also analysed the proportion of time spent in the four interaction modalities across the two different interfaces, and the number of errors made (the number of vertex placed and then deleted). But, as described in the *Results* section, we did not find any statistically significant difference. Therefore, in the following, we will elaborate on the *accuracy* of the two 3DUIs and on user *preference*.

6.1 Accuracy

The accuracy of the volume drawn to its physical counterpart (and subsequently of the substituted object) can greatly impact the believability of the experience [16]. Objects that appear larger in VR than their physical proxy can cause users to visually collide with the virtual object without experiencing any passive haptic feedback. Thus, in systems where users are responsible for creating or adapting a VE, the accuracy of the interaction technique or the visualisation modality used plays an important role.

When asked about a subjective assessment of their performance, several participants stated that they felt they performed better while using the tablet (see Figure 6). This is supported by the analysis of the task completion times, but not by the analysis of the size mismatch, which revealed

the better accuracy of the HMD-based interface. In the VR condition we observed how participants moved more carefully, which explains the slower times. Similarly, their outside spatial awareness was limited to the 2D visualisation next to the Vive controller in the VR view. Thus, participants were generally more attentive in placing the vertices of the box they were drawing. For example, we observed participants complement their lack of stereo vision, due to the monoscopic real-world window, with the haptic feedback resulting from lightly touching the object with their controller to better identify the corners.

In the tablet modality, participants were unimpeded by cables. Indeed, as shown by the task completion times, they completed the tasks quicker. Being in the real world, they had a perfect awareness of their surroundings. We believe that the 16% larger volumes obtained with the tablet can be explained by the fact that users did not touch their controller to the object.

6.2 Preference

While the accuracy of the algorithm responsible for creating bounding boxes can be improved, our participants still significantly preferred performing the activity of adapting a VE through the immersive VR visualisation. Participant #1 stated that *“it was easier to see what I was doing.”* ; *“the HMD felt more intuitive and connected to the environment.”* (#2); *“with the HMD I felt more in control”* (#15). Conversely the main reason for the lower preference of the tablet-based interface are due to the disconnection between the movement of the tablet and the feedback from the virtual camera. As noted, the view from the tracked position and orientation of the controller was shown on the tablet. However, some participants stated that *“[they] focused more on the controller”* (#11) or that *[they] did not like to focus on two things simultaneously.* With the tablet *“[they] would not look at it and only glance at it sporadically, whereas the HMD was always in front.”* (#4). Although the release of standalone trackers or inside-out tracking (technologies that were not available at the time of our study) will mitigate the disconnection between the position and orientation of the tablet and those of the virtual camera, the potential of users not glancing at it as much, raises bigger concerns.

6.3 Future Research Directions

Future studies should update the results of this study when progress in inside-out or standalone trackers make the prospect of tracking a tablet without external equipment, a viable one. Other alternative visualisations or setups warrant further investigation. For instance, a visualisation based on first-person Augmented Reality capable of aligning the virtual volumes to the physical environment might provide new insights. Investigating whether just relying on the controller to define the volumes could evaluate the effect of supporting the task with or without a device providing a “window” on the virtual world.

The HMD-based visualisation could be improved by enhancing the view of the outside world while immersed in VR. Part of the reason for the longer times can be ascribed to the difficulties in moving in a physical environment based purely on the information provided by the non-stereo window attached to the controller. This might be achieved by immersing the user in a VE resulting from a coarse reconstruction of the PE (such as the one provided by the Kinect Fusion [13]), which might help users navigate their PE with more ease. Future research could investigate how much of the PE to bring in the VR view to support this task, or conversely, how much of the VE to bring in the PE.

A further research direction arising from systems that allow end-users to design or adapt VEs to their physical surroundings, is the potential loss of novelty that could be caused by this process. We envision these systems as providing a prerequisite step to other activities that would then take place in the designed VE. However, if users, as VE designers, also have a role in how the environment will appear, they might lose some of the enjoyability resulting from discovering an environment created by someone else. Future studies could look at this from a games research perspective to understand under which circumstances users designing a VE for their own use could result in detracting from their enjoyment or enhancing it.

7 CONCLUSION

In this paper we evaluated two visualisation modalities supporting the task of adapting a Virtual Environment to the physical layout of the domestic environment in which the user is in. Our research investigated whether this process is best supported by providing a view of the outside world while immersed in VR or, conversely, by providing a view of the Virtual Environment through an external device. Our results show that although users were slower, they rated the VR interface significantly higher, which also led to a more accurate reconstruction of the physical environment. The preference shown in favour of performing this activity while immersed in VR highlights the need to research better supporting interaction techniques and visualisations, and the challenge of how much of the outside world to bring in the virtual world or vice versa.

ACKNOWLEDGMENTS

This research was funded by the Research Development Fund of the University of Portsmouth, UK.

REFERENCES

- [1] Camille Barot, Kevin Carpentier, Marie Collet, Andrea Cuella-Martin, Vincent Lanquepin, Mathilde Muller, Esteban Pasquier, Loic Picavet, Arthur Van Ceulen, and Kevin Wagrez. 2013. The wonderland builder: Using storytelling to guide dream-like interaction. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*. IEEE, 201–202.
- [2] Federico Cabitza, Daniela Fogli, and Antonio Piccinno. 2014. Fostering participation and co-evolution in sentient multimedia systems. *Journal of Visual Languages & Computing* 25, 6 (2014), 684–694.

- [3] Epic Games. 2017. Unreal Engine VR Editor. (2017). <https://docs.unrealengine.com/latest/INT/Engine/Editor/VR/>.
- [4] Jose Garcia Estrada and Adalberto L. Simeone. 2017. Recommender System for Physical Object Substitution in VR. In *Poster at the 2017 IEEE Conference on Virtual Reality*. IEEE.
- [5] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, and Jianwei Wan. 2014. 3d object recognition in cluttered scenes with local surface features: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 11 (2014), 2270–2287.
- [6] Taejin Ha and Woontack Woo. 2013. ARWand for an augmented world builder. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*. IEEE, 207–208.
- [7] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9, 1 (2013), 1.
- [8] Anuruddha Hettiarachchi and Daniel Wigdor. 2016. Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1957–1967.
- [9] Philipp Hock, Sebastian Benedikter, Jan Gugenheimer, and Enrico Rukzio. 2017. CarVR: Enabling In-Car Virtual Reality Entertainment. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 4034–4044.
- [10] Brent Edward Insko. 2001. *Passive haptics significantly enhances virtual environments*. Ph.D. Dissertation. University of North Carolina at Chapel Hill.
- [11] Jason Jerald, Paul Mlyniec, Arun Yoganandan, Amir Rubin, Dan Paullus, and Simon Solotko. 2013. Makevr: A 3d world-building interface. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*. IEEE, 197–198.
- [12] Mark Mine, Arun Yoganandan, and Dane Coffey. 2015. Principles, interactions and devices for real-world immersive modeling. *Computers & Graphics* 48 (2015), 84–98.
- [13] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 127–136.
- [14] Lior Shapira and Daniel Freedman. 2016. Reality Skins: Creating Immersive and Tactile Virtual Environments. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 115–124.
- [15] Adalberto L. Simeone, Ifigeneia Mavridou, and Wendy Powell. 2017. Altering User Movement Behaviour in Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics* 23, 4 (April 2017), 1312–1321. <https://doi.org/10.1109/TVCG.2017.2657038>
- [16] Adalberto L. Simeone, Eduardo Velloso, and Hans Gellersen. 2015. Substitutional Reality: Using the Physical Environment to Design Virtual Reality Experiences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 3307–3316. <http://doi.acm.org/10.1145/2702123.2702389>
- [17] Ruben Smelik, Tim Tutenel, Klaas Jan de Kraker, and Rafael Bidarra. 2010. Integrating procedural generation and manual editing of virtual worlds. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. ACM, 2.
- [18] Gillian Smith, Mike Treanor, Jim Whitehead, and Michael Mateas. 2009. Rhythm-based level generation for 2D platformers. In *Proceedings of the 4th International Conference on Foundations of Digital Games*. ACM, 175–182.
- [19] Martin Spindler, Christian Tominski, Heidrun Schumann, and Raimund Dachselt. 2010. Tangible views for information visualization. In *ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 157–166.
- [20] Misha Sra, Sergio Garrido-Jurado, Chris Schmandt, and Pattie Maes. 2016. Procedurally generated virtual reality from 3D reconstructed physical space. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. ACM, 191–200.
- [21] Maureen C Stone, Ken Fishkin, and Eric A Bier. 1994. The movable filter as a user interface tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 306–312.
- [22] Tuukka M Takala, Meeri Mäkäräinen, and Perttu Hämäläinen. 2013. Immersive 3d modeling with blender and off-the-shelf hardware. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*. IEEE, 191–192.
- [23] Dimitar Valkov and Steffen Flagge. 2017. Smooth Immersion: The Benefits of Making the Transition to Virtual Environments a Continuous Process. In *Proceedings of the 5th Symposium on Spatial User Interaction (SUI '17)*. ACM, New York, NY, USA, 12–19. <https://doi.org/10.1145/3131277.3132183>
- [24] Jia Wang, Owen Leach, and Robert W Lindeman. 2013. DIY World Builder: an immersive level-editing system. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*. IEEE, 195–196.
- [25] Jia Wang and Robert Lindeman. 2015. Coordinated hybrid virtual environments: Seamless interaction contexts for effective virtual reality. *Computers & Graphics* 48 (2015), 71–83.
- [26] André Zenner and Antonio Krüger. 2017. Shifty: A Weight-Shifting Dynamic Passive Haptic Proxy to Enhance Object Perception in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 23, 4 (2017), 1285–1294.