UNDERGRADUATE THESIS CPS 40 A/B PROJECT
FINAL REPORT

Department of Computer Science
Ryerson University

Designing an Architectural Model Explorer for Intuitive Interaction and Navigation in Virtual Reality

Submitted by Alessandro Profenna
Supervised by Dr. Tim McInerney
April 29, 2019

Table of Contents

Abstract

The availability of modern virtual reality (VR) technology provides increasingly immersive ways for the average person to observe and view building models. In this paper, prior research of multi-scale virtual environments, interaction and navigation methods, and existing commercial software were combined and extended to design a simple and intuitive architectural model explorer in VR. Specifically, this paper aimed at understanding how such a program should interface with a naïve user, that is, someone inexperienced with VR and architecture. A user study was conducted with 10 participants to observe the holistic effectiveness of the design decisions of the created program, such as the navigational benefits of the moveable small-scale model and its WIM-like interface.

The program was developed in Unity for the HTC Vive VR headset and was designed to include a full-scale model of a building and a small-scale 1:10 version of the same building within the same virtual space. The small-scale model can be moved wherever the user chooses or hidden completely. It can also be rotated and sectioned-off using a cutting plane. The user can teleport within the full-scale model by selecting a point on the ground around them or by choosing a point in the small-scale model. All interaction methods, including navigation, are initiated using a remote point and grab technique, combined with visual cues like colour changes, highlights, and affordances. Once something is grabbed, it can be manipulated using either hand motion or directional buttons on the controller's trackpad. With multiple options for teleportation and manipulation techniques, each user was given a navigational task and recorded and surveyed to understand their preferences and/or frustrations.

The results of the user study showed that the design of the program was mostly a success and that the small-scale building model used is a great multi-purpose tool for navigational aid. Being able to teleport both locally and using the small-scale model is advantageous. Pointing and grabbing for selection is useful for self-discovery and quick learning, and indication through magic/affordance is especially important. Inexperienced VR users instinctively used motion for movement and manipulation. It was therefore good enough for general usage, especially since users did not care to use the directional buttons for more precision.

Introduction

Interactive computer visualizations are incredibly useful in the development and communication of architectural designs. The architecture industry relies heavily on Building Information Modeling (BIM) software in a three-dimensional context, which is crucial in providing a virtual representation of a building's spatiality. The availability of modern VR technologies provides increasingly immersive ways for both architects and their clients to observe and view building models. "The integration of this technology in the design process, provides improved understanding and control on the overall multi-specialty architectural design, since the representation of the project in VR is progressively adjusted according to a specific process phase and task," [7]. The key advantage that VR provides for the field is the potential for more natural model interaction and navigation, especially in a "Multi-Scale Virtual Environment" (MSVE).

The goal of this project is to research, combine, and extend some of the various methods of interaction and navigation currently used in VR technology within the architecture industry and to observe their holistic effectiveness. The focus, specifically, will be on creating an interactive VR program for exploring an architectural model that is intuitive and simple enough for naïve users. Naïve, in this context, refers to those who are unfamiliar and/or inexperienced with architectural design, 3D modeling, and most importantly, virtual reality. This includes clients of architects and the average homebuyer interested in virtually experiencing the interior space of a potential new home.

Spatial awareness, orientation, and affordance all play vital roles in navigating and interacting within a virtual space. It is common for a person to become 'lost' within a model [9] resulting in an "unproductive and unpleasurable experience, even when trying to do the most basic three-dimensional navigation operations" [6]. For exploring a digital architectural model in VR, the key is communicating the design and usability to the user, while also allowing for manipulation. This includes model sectioning, scaling, and rotation, as well as teleportation and pre-set viewpoints. However, ensuring these methods are implemented in a way that minimizes naïve user disorientation and confusion in a multi-scale virtual environment is an ongoing problem.

I want explore further solutions to this problem by implementing:

1. The juxtaposition of the “dollhouse view” small-scale model and the full-scale 1:1 model with viewable interiors in the same virtual space
2. Intuitive selection, manipulation, and affordances for interaction and navigation

Through this, my research project will address the following questions:

1. What are the navigational benefits of supplying the user with both a small-scale and full-scale 3D model of an architectural building design?
2. How can researched methods of interaction/affordance be combined/extended to support this WIM-like interface for an improved naïve user intuition?

Background

The following is a summary of the research required for designing an architectural model viewer for VR. Design considerations for the experiment of this thesis (the VR program) are based on the conclusions and suggestions presented by the research below.

WIM and Transitioning Between Scales

Exocentric view aids can facilitate the acquisition of survey knowledge in a virtual building [16]. Spatial navigation in a building is always constrained within specific areas, such as on floors and staircases. This creates difficulty for new users to acquire the accurate survey knowledge of a multilevel building [16]. A 3D building map is an effective aid that better facilitates the acquisition of knowledge on the vertical dimension of a virtual multilevel building [16]. One extreme exocentric view is the three-dimensional (3D) model of the environment, such as ‘‘World In Miniature’’ (WIM) [16].

The WIM concept was originally created by Stoakley and Paush, who discussed object selection, object manipulation, user travel, visualization issues and props in the WIM [26]. WIM is like a miniature copy of the life-size virtual world and the objects in the model each correspond to a life-size object. Having the WIM along with the full-scale model gives the user another point of view from which to observe the scene. [23] The WIM therefore naturally offers two different scales to the user without requiring explicit modes or commands. [23] Additionally, if the WIM includes some representation of the user as an object in the scene, the user can simply reach into the WIM and “pick themself up” to change their location in the environment. [23]

Furthermore, scaling and camera movements are important factors when working with smaller versions of a virtual architectural model. For example, users find that rotating the camera with an immediate update of the full-scale world is very disorienting; discontinuities of the camera motion are to be avoided at all costs [23]. A solution to prevent disorientation in WIM is to maintain a reasonable scale [23]. In a “Scaled-Scrolling WIM” (SSWIM) study, it was found that users did not scale on a regular basis; they generally set the scale of the world as appropriate to the city and rarely adjusted the scale afterwards. [26] Furthermore, in a study observing how students view a 5-meter high temple in VR at different scales, the most preferred observation scale was (1:10) [24]. This scale resembles a virtual mockup in front of the user wearing the HTC Vive, which is very close to the scale of an architect working with mockups on a table in real world. Therefore, the closer the scale between a virtual avatar and his/her operation targets to the scale between a real human body and mockups on a table, the better the learning performance is. [24] This is especially useful for naïve users.

Similar to a building, the human body is another example of an object that reveals contextual information when observed at different scales. In a study investigating the anatomy of the human body in an MSVE, it was noted that the size of the user at any location must be compatible with the scale at that location. Apart from scaling the user appropriately, MSVEs must address other issues, such as how to tell the user which objects of the environment are at different "levels of scale" (LoS) and to how to make it easy for the user to travel between these levels while maintaining spatial orientation and understanding. [13] To give the user information about where they are currently located, a miniature model of the current LoS was shown as a visual cue to the user and contained a blinking dot that represented the user's position [13]. It was also found that that automatic scaling was clearly more efficient than manual scaling [13].

On the topic of scaling in VR, Glueck and Khan's paper explains the concept of intellection, defined in the paper as "the process by which a user reasons about the scene they are experiencing. A user, represented by a virtual camera, is required to first decipher their own position, orientation, and most difficultly, estimate their own size, within the 3-D environment," [12]. The prevalence of multi-scale environments, such as Google Earth, is increasing [12] and the difficulties of such virtual spaces are a problem of over-constraints, in that an optimal solution satisfying all conditions at different scales is essentially non-existent. Ideally, to get around this, the system should "dynamically extracts relevant features at different scales such that a minimum number of exocentric overviews are required to communicate position and orientation," [12]. Maintaining coherence increases in difficulty with a greater number of represented scales. For a user that wants to control the scale they are interacting with, discrete scales are better at communicating the scale change transition. Optimal cues such as depth of field and realistic rendering techniques also aid in expressing scale and size [12]. Note that a user's sense of their own size plays a vital role in understanding scale, however, grounding knowledge of their own exact virtual size is not usually available. Absolute size is still ambiguous, because a user inherently has no size [12]. Research on how to best support transitions between scales has not received enough attention or focus, particularly in ensuring user awareness with regard to their position/orientation, and communicating an implicit sense of scale [12].

Navigation and Locomotion

Navigation can be described as "the general process by which a user changes the position and orientation of the virtual camera used to render their point of view," [12]. It too, is a method of reasoning that helps the user, especially when the VR tools themselves recognize and harness this [12]. Locomotion refers to how the user actually moves around the space. There are four distinct VR locomotion types: [3]

- Motion-based: The VR locomotion techniques under this type utilize some kind of physical movement to enable interaction, while supporting continuous motion in open VR spaces. This VR locomotion type includes such techniques as walking-in-place, redirected walking, arm swinging, gesture-based locomotion and reorientation.

- Room scale-based: This VR locomotion type utilizes physical movement to enable interaction, and it supports continuous motion (as with the motion-based type); however, the interaction takes place in VR environments whose size is limited by the real environment's size

- Controller-based: For this VR locomotion type, controllers are utilized to move the user artificially in the VR environment. The VR interaction space is open, and the motion is continuous. This type includes such techniques as joystick-based, human joystick, chair-based and head-directed locomotion.

- Teleportation-based: The VR locomotion techniques under this type utilize artificial interactions in open VR spaces with non-continuous movement, as the user's virtual viewpoint is instantaneously teleported to a predefined position by utilizing visual "jumps". Point and teleport is a VR locomotion technique that falls under this type.

For traveling near and medium distances within indoor virtual environments, such as an office building or real estate, Joy Stick (JS), Teleportation (TP), and Redirected Walking (RDW) are suitable techniques. Through a user study of locomotion in room-scale VR, it was found that TP and RDW have different benefits and drawbacks and should be preferred over JS since they were superior in most of the tested criteria [15]. These two techniques were also preferred by most of the participants, as JS lead to a significant increase in motion sickness. RDW allows users to gain better special knowledge, while TP is faster [15]. However, there are some time delays with TP due to several teleports required for users to reach their next target, along with users needing to reorient themselves after each teleport [15]. Also, the study performed by Sun et al revealed that fishing mode (Teleportation via parabolic raycasts) is preferred rather than flying mode in HTC Vive navigation [24], and the results of Zheng and Minsheng's study of qualitative data analysis suggest that teleportation is easier to learn and use than physical walking and zoom [29].

Regarding teleportation specifically, it has been hypothesized that for some tasks, pre-orientation of an avatar and preview of the post-teleport view will help reduce overall task completion time [8]. Elvezio et al demonstrated an interaction technique that allows a user to point at a world-in-miniature representation of a city-scale virtual environment and perform efficient and precise teleportation by pre- orienting an avatar. This is accompanied by a preview of the post-teleportation view of the full-scale environment, which updates as the user adjusts the position, yaw, and pitch of the avatar's head [8]. Additionally, a very basic visually simulated reference frames to the virtual scenes can significantly enhance user performance. That is, adding visually simulated reference frames consisting of only a simple wireframe rectangular box is enough to help VR users complete a navigational task in shorter times, with less revisits, and with shorter trajectories [18].

There is a need for more user-centric, empirical research approaches in VR, potentially under comparative settings [3]. Investigating the conditions under which spatial orientation is improved will not only deepen our understanding of human spatial cognition, but can also guide the design of more effective VR simulations [18]. The field of research on VR locomotion, in its new era, is still uncharted [3].

Selection, Manipulation, and Affordance

Using a VR "Head-Mounted Display" (HMD) encourages users to perceive their control method as more natural, which in turn determines how present in the virtual world they feel. Players respond well to controls they perceive as natural, becoming more present in the game world as a result [21]. In Bowman's "3D User Interface Design" [4] the following four principles for 3D UI design are outlined:

1. Consider "magic" interfaces in place of "natural" ones when tasks require productivity and efficiency.
2. Choose interaction techniques based on the requirements of the application – the same set of techniques will not work well in every situation.
3. Limit the required degrees of freedom for input whenever possible, and provide physical or virtual constraints to help guide user input.
4. Take advantage of the increased flexibility afforded by whole-body input, multimodal I/O, and novel input devices, but also give the user structure and support in managing the increased complexity.

A more complex interaction system, which includes grabbing elements of the world or having the elements react to user input, would certainly increase the immersion of users and could leave a more noticeable impact on them [17]. Note, however, that grabbing and manipulation must be considered separately for overall usability. Users in a study by Bowman et al found it easier to grab an object using ray-casting than with any of the arm-extension techniques, but no users preferred ray-casting techniques for object manipulation [5]. The HOMER (Hand-centered Object Manipulation Extending Ray-casting) technique is a hybrid of the two that offers many advantages over arm- extension techniques: object grabbing is easier, objects at any distance may be selected with the same amount of physical effort, object manipulation requires less physical effort, and object distance is easier to control [5].

To further immerse users in VR, it is also beneficial to avoid mismatches between what is felt in reality versus what is felt in a virtual world. These mismatches can disturb the coherence and meaningfulness of the set of stimuli given in VR: object mismatch (expected feedback in the physical world differs), time mismatch (latency, causing motion sickness), and spatial mismatch (virtual positions/scales of objects don't match with real ones). This can cause the sense of presence to decrease [27]. It is critical, however, for a specialized system to only display the most relevant information for the current task instead of overloading too much information onto the user's attention [25].

Immersion as affordance can be further discussed in three aspects. [22]

1. Immersion is a multifaceted concept broadly encompassing media, users, and contexts.
2. The importance of user cognition shows how users play an increasingly active role in forming immersion itself and in turn how immersion influences user-learning experiences.
3. Immersion can be a fluid and reflective concept rather than a fixed and isolated factor.

It is important to remember that designers, users, and clients all have different conceptual models and reference frames. Therefore, the correlation of perceptions of the different parties involved must be considered when guiding the design of a building through affordances [14]. "At a basic level the affordances perceived by all these parties are common and derive from everyday use of the built environment. Distortion comes from differences in priorities and related semantic and cultural constraints," [14]. VR learning and interaction are positively associated with affordances, and therefore the perceived affordance held by users is positively associated with usability and learnability [22]. Any such cue plays an important role as they trigger latent beliefs, including interactivity, flow, immersion, presence, and perceived quality. Eliciting these beliefs can facilitate the adoption of VR services greatly in learning contexts [22]. Highlighting the relationships between an architectural design and its structural system is vital to immersion. More investigations and researches should be also directed to this area [1].

It should also be noted that an affordance on its own does not necessarily communicate its action potential; variables such as its orientation and distance from the user play a role. This means that "potential for immediate action may bias attention toward an affording object," [11]. Our attention draws toward clearly graspable shapes of between two and eight centimeters (about 1 to 3 inches), and it's possible that the absence of graspable objects in 20th century architecture causes stress and anxiety [20]. Neurological and hormonal signals prepare us mentally to grasp objects in our close environment that we perceive to fit our hand [20]. A positive prehensile reaction enhances a person's wellbeing and performance, whereas a negative affordance set off by structural details that isolate us will influence all other actions negatively through superimposed anxiety and fatigue [20]. A minimalist approach to design frustrates the object affordance mechanism, diminishing or eliminating any prehensile connection with our surroundings [20]. The Five architectural characteristics of object affordance are Size, Shape, Material, Texture, and Distance [20]. Note that the structural range of both affordances and objects is actually small; what creates the diverse experiential flavour of gameplay is the design and application of mediating variables to this small set. [19].

Feedback is especially significant as well when selecting things in an immersive environment. Feedback, such the tactile or audible feedback from pressing buttons on a telephone, is useful in communicating opportunities to the user. Buildings cannot provide the same sort of level in their spatial feedback, where many design flaws, such as limited leg room, may only become apparent over time [14]. Yu et al conducted a study where RayCasting with visual feedback was deemed the easiest to learn and more comfortable compared to other selection methods [28]. They also learned some valuable lessons and made recommendations, including the following: Lesson 1) Visual feedback seems to be the most natural and can lead to better performance and lower error rates. Lesson 2) In cases where users have a very short time to familiarize themselves with the techniques, simple RayCasting with visual feedback could work best as it is aligned with how standard cursor movement works. Recommendation 1) A simple technique, like RayCasting with direct visual feedback, can work well for complex environments where many target distractors are clustered together. Recommendation 2) When considering feedback, visual feedback seems to elicit a quick response [28].

Context-Driven Interaction

S. Frees' paper explains that "interaction context represents the current state of the user, including position, interaction history, and intentions or objectives while performing a task," and that "context in fact represents all knowledge the system has about the user's objectives – the sum of many small bits of information," [10]. The question then becomes, how can context be used to support user interaction in an effective manner?

Designing a program with various different features requires the developer to decide on how/when the user can interact with such features. In a virtual world, there are competing contexts with which a user may interact with an object: one requiring free-flowing manipulation and another requiring constraints to support precision [10]. Now, the developer must decide if both contexts should be supported by one monolithic interaction technique, or different techniques chosen by the user. Both options have issues; the first may not support either context very well while the second may interrupt natural interaction through explicit mode switching [10].

In order for developers to select the best interaction technique for the user, it helps to model contextual information. Through an in-depth literature review, the most commonly used high-level characterizations of context were identified, which are referred to as context components in the paper. They are Level of Control, Workspace, Frame of Reference, Object Groupings, and Constraints [10]. Note that these cover most, but not all, common scenarios in a typical application. This leads to the idea of task/context pairs, which is the combining of an interaction task (e.g., "translate object") and a context component (e.g. high or low Level of Control). Doing so helps in the selection of an individual interaction technique (e.g. PRISM or Go-Go) [10].

A developer must also implement methods for the application to figure out the current context, that is, to have a CRM (Context-Recognition Mechanism) [10]. A CRM may be implicit (context is deduced from observed behavior) or explicit (such as a menu system). For each task/context pair, the development and selection of appropriate CRMs requires significant thought, development time, and evaluation using extensive domain knowledge [10]. Implicit CRMs specifically are more challenging to develop and require user studies to avoid creating frustrating experiences for the user [10].

The paper suggests a general strategy towards the development of context sensitive user interfaces [10]:

1) Select relevant Task/Context pairs for the particular application
2) Select a Context Recognition Mechanism (CRM) to calculate each Task/Context pair
3) Select interaction techniques to support each range associated with each Task/Context pair

The selection of which technique to use for each context should be informed by literature and usability studies [10].

Existing Software

The following is a list of some of the VR architecture explorer software that is commercially available in the industry as of the time of this writing, along with notes on their notable features. Some use fully immersive VR headsets while others are used with mobile devices.

Table 1

Existing Software

Software	Notes
Enscape	• Revit Plugin • free roam • ability to walk and look around • change floor you're standing on with one click • real-time changes of object materials and placements, and time of day in the revitalization editor
IrisVR	• ability to walk and look around • Scale Model Mode • teleport to specific rooms or areas using a point and click menu • place your first-person view by pointing at scale model view • Model sectioning
ImmerseCreator	• collaborative building and 3D design
VRtisan	• teleport and look around • turn lights on and off • pick up objects, change properties like colour
REinVR	• for real estate • 360 degree views • animated movement • photo-realistic
InsiteVR	• Similar to Iris VR with "dollhouse view" and teleportation • ability to walk and look around
Symmetry	• Similar to IrisVR
ArqVR	• Allows you to remove all furniture in one click or move individual pieces of furniture
TruVision	• Walk around, change colours and materials of building walls

Methods

There are a few things to consider to develop a sophisticated VR experience: 1) knowing the user, the problem, outline the expectation, and the final goal, 2) VR platform (desktop or mobile) and identify the content of creation in reference to VR experience. 3) Integration method and developing time. [2].

With this in mind, the goal was to combine and extend the results of past VR research in the domain of architectural model interaction, and to thus create an application tailored to naïve users, i.e., those unfamiliar with VR and architecture. The program that was created contains a full-scale model of a building that can be navigated and explored, along with a small-scale version (1:10) of that same model, which can be interacted with in various ways. The following features that are common to architectural VR software have been chosen for this experiment:

1) Rotating a scale-model
2) Slicing a scale-model horizontal using a cutting plane
3) Choosing a location within the small-scale model to teleport to in the full-scale model
4) Teleporting within the full-scale scale model

Since this application is being developed for naïve users, the features do not need to be robust or numerous, but instead must be intuitive, quick to learn, and with a reduced potential for inducing motion sickness, disorientation, or discomfort. Design decisions regarding the chosen selection, manipulation, and navigation techniques, along with visual cues, and affordances, were all informed by the research above. A user study was conducted to measure the holistic effectiveness of these decisions to help us better understand what an architectural model viewer for naïve users should be like.

Technical Resources and Assets

To create this program, a lab station was set up with a powerful PC in the George Vari Engineering Building at Ryerson University. It was developed using the Unity 2018.1 game engine to work with the HTC Vive virtual reality headset and controllers. The SteamVR 2.2 plugin was downloaded to interface between Unity and the VR hardware. The HTC Vive was chosen because it is a full VR HMD (Head-Mounted Display) with controllers, and can easily have applications developed for it within Unity. It is among one of the newest and most immersive commercially available means of experiencing virtual reality simulation. Other types of VR, such as those implemented with smartphones, would not allow for the interaction fidelity required by the design of this simulation.

Also used in this program are the following third-party assets from the Unity Asset Store:

1) "Brick Apartment Seven With Exterior Textured"

Represents a three-storey apartment building model. It represents a good balance between a regular-sized home and a larger building, and each floor is identical. It is used as the definitive model that will be explored by the user.

Link to the asset:

https://www.turbosquid.com/3d-models/brick-apartment-building-interior-exterior-3d-obj/958368

2) "Cross Section"

This includes a set of shaders used for sectioning model geometry. In Architecture, it is a common use-case to section off chunks of building models in order to view interiors and see how rooms/floors are related/connected to each other. This asset is used in the following way: One of the shaders from the asset package is attached to a translucent quad, which will act as the cutting plane. Another shader from the package is attached to the model of the building, which will be sectioned by the cutting plane.

Link to the asset:

https://assetstore.unity.com/packages/vfx/shaders/cross-section-66300

Simulation Design and Implementation

Scene Description:

This program consists of only one scene, as shown in Figure 1. There is a skybox with a realistic landscape for increased immersion. There is a 1:1 (small-scale) model and 1:10 (full-scale) model of the Brick Building prefab sitting atop of a circular ground-plane. The small-scale model is located next to the full-scale model, but can be moved to wherever the player chooses. It is an exact copy of the full-scale model, with the same level of detail in the interiors and exteriors. Note that there is no furniture within the models. When the program begins, the user is virtually standing in front of the two models, and is at a 1:1 scale. The player's scale never changes.

Figure 1
Scene view, with the full-scale building model and small-scale building model

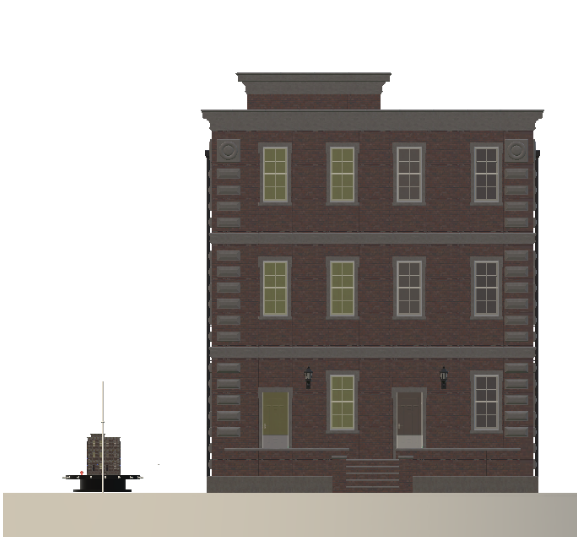


Figure 1(a)
Orthographic back view

Figure 1(b)
Orthographic top view with the floor plan exposed

Figure 1(c)
Perspective view of the entire scene

Small-Scale Model Description:

The small-scale model in Figure 2 exists as a navigational aid and companion for spatial awareness. It sits atop of a black, round table. The table can be rotated, which is indicated by the white spoke handles all around it. There is also a cross-section cutting plane that can be moved up and down to slice the small-scale model horizontally. Any part of the model above the cutting plane will be ghosted (very translucent) so that the user can view the interiors. The cutting plane is represented by a translucent blue square, attached to a white pole apparatus at a joint. This apparatus exists to indicate to the user that the cutting plane can move along the vertical axis. Within the small-scale model is a red, humanoid, small-scale avatar that dynamically moves to represent the location of the user in the full-scale model. The entire small-scale model, with the table and cutting plane, can be moved to whichever point on the ground/floor the user chooses. If the small-scale model is within the full-scale model, then the full-scale model's walls will animate away to avoid the two models from mesh clipping.

Figure 2

Small-scale building model, which includes a rotatable table, a vertically moving cutting plane, and a red avatar representing the user's location within the full-scale model

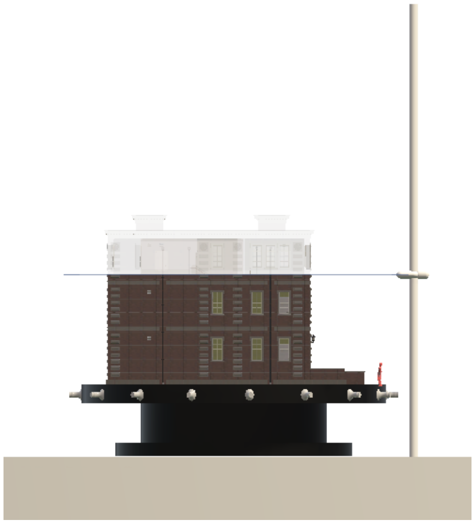


Figure 2(a)

Orthographic side view

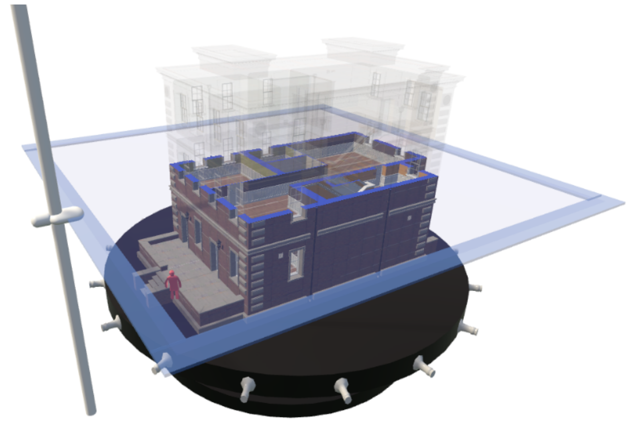


Figure 2(b)

Perspective view

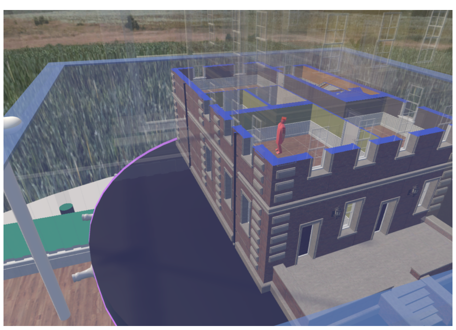


Figure 2(c)

The small, red avatar shows that the user is currently in a room on the second floor, facing the window

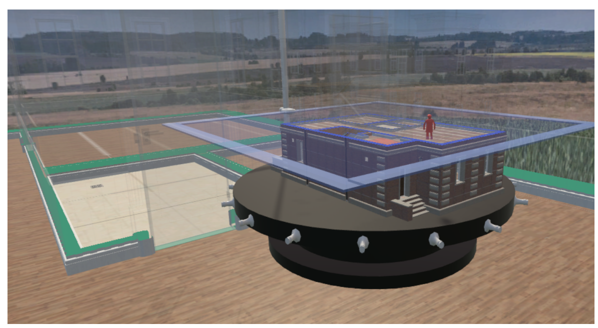


Figure 2(d)

The walls are down and invisible in the full-scale model because the small-scale model is present

Navigation:

The user can navigate the full-scale space through teleportation, and can do so using two different methods. The first is local teleportation, which is done by pointing at a spot on the floor or ground around the user in the full-scale model and choosing to teleport there. The second would be through small-scale model teleportation, which would have the user pointing at spot on one of the floors in the small-scale model and teleporting there within the full-scale model. The valid navigational surfaces are the ground plane, all three floors of the building, and the staircases within the building. A ghosted avatar is used to represent the teleportation point of the user, and can be pre-oriented before teleportation to help with user-orientation. If the user, for example, teleports to a corner and is facing a wall, they'd have to physically turn around to re-orient themselves; however, pre-orienting the avatar can solve this issue beforehand.

Interaction:

The user only requires one Vive controller and two buttons: the trackpad and the trigger. All interaction is done using the same method. Interaction is performed using a style of the HOMER technique, with selection being achieved via ray casting. A straight, raycasted line extends from the end of the controller; if it intersects something that can be interacted with, it will change colour and the object will highlight with that colour, depending on what the object is (red for a point on the ground/floor, orange for the cutting plane, purple for the small-scale model table). A ghosted hand will appear at the selected object to indicate that it can be grabbed. This essentially "selects" the object. Pressing the trigger "grabs" the object remotely, and a curved, tethered line appears to indicate a connection between the controller and the hand at the grab point. Manipulation of an object is only possible if it is grabbed; once something is grabbed, it can be performed using two optional methods: directional buttons that appear on the virtual controller's trackpad, or hand motion. The directional buttons allow for more precise, discrete, and un-constrained level of control, while hand motion allows for a quicker, natural, and intuitive level of control with pseudo physics.

There are four things that can be interacted with in this program, as outlined in Table 1.

Table 2
Interactive actions within the program

Selected	Grabbing	Manipulating
Small-scale model table (See Figure 3)	The user's controller is tethered to the closest spoke handle on the table, and manipulation is set up in a rotation context. Left and right directional arrow buttons appear on the	Rotation of the table proceeds around the model's y-axis: Hand motion: Move hand left to rotate clockwise, right to rotate counter-clockwise. The change in rotation is based on the distance of the left/right

	controller's trackpad. Let go to release.	hand movement. A quickly dampened rotational force is applied upon release. Directional buttons: Click left arrow button to rotate clockwise, right arrow button to rotate counter-clockwise. Can be held.
Small-scale model cutting plane apparatus (See Figure 4)	The user's controller is tethered to the cutting plane joint, and manipulation is set up in a translation context to vertically move the cutting plane. Up and down directional arrow buttons appear on the controller's trackpad. Let go to release.	Translation of the cutting plane proceeds along the model's y-axis: Hand motion: Move hand up to raise, and down to lower. The change in elevation is based on the distance of the up/down hand movement. Directional buttons: Click the up arrow button to raise, and down arrow button to lower. Can be held.
Point on full-scale ground/floors (See Figure 5) OR Point on small-scale model floors (See Figure 6)	The user's controller is tethered to the selected point and sets up a navigation context within the full-scale model. A ghosted, humanoid avatar is shown at the point to represent the player post-teleportation, and is set up in a rotational context to pre-orient the user. Left and right arrow buttons appear on the trackpad. Releasing the trigger teleports the user to this point in the full-scale model, facing the direction of the avatar.	Rotation of the avatar proceeds around the avatar's y-axis: Hand motion: Move hand left to rotate counter-clockwise, right to rotate clockwise. The change in rotation is based on the distance of the left/right hand movement. Directional buttons: Click left arrow button to rotate counter-clockwise, right arrow button to rotate clockwise. Can be held.

Note: The small-scale model can be moved by pointing at the ground/floor in the full-scale model and pressing the trackpad to instantly move it to the selected point. The model can be hidden by pointing at it and pressing the trackpad.

Figure 3

Small-scale model rotation

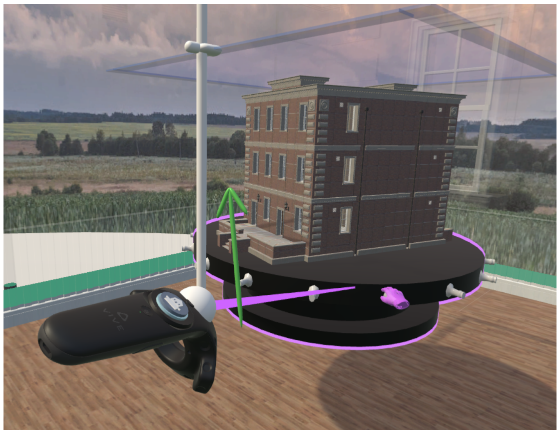


Figure 3(a)

Pointing and selecting the small-scale model table

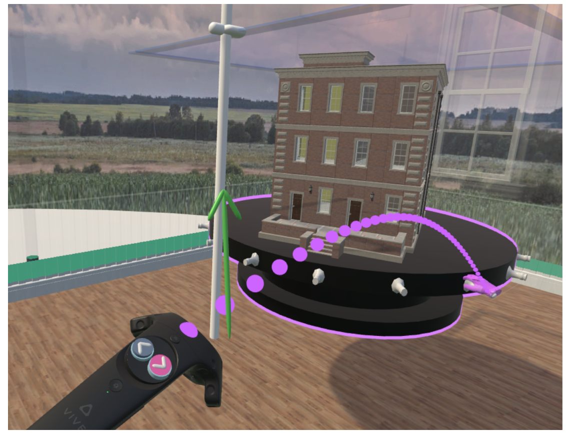


Figure 3(b)

Grabbing the small-scale model table and rotating it counter-clockwise

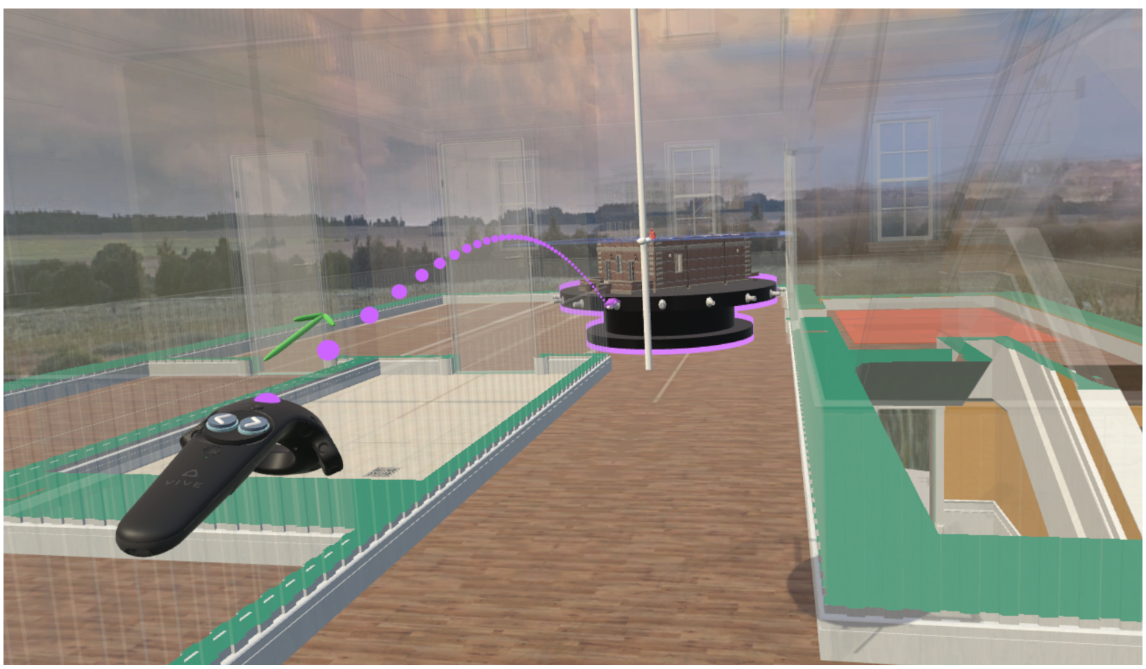


Figure 3(c)

Grabbing the small-scale model from a distance

Figure 4

Vertical translation of the small-scale model's cutting plane

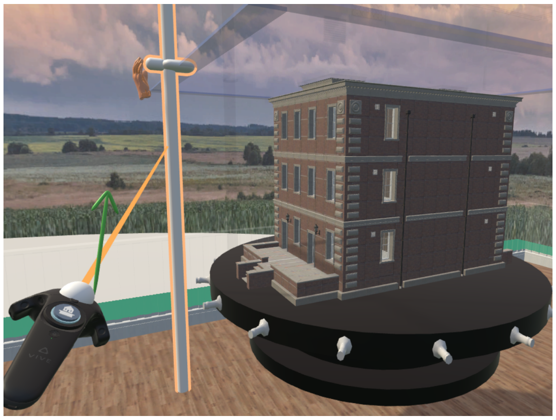


Figure 4(a)

Pointing at the pole apparatus to select the cutting plane

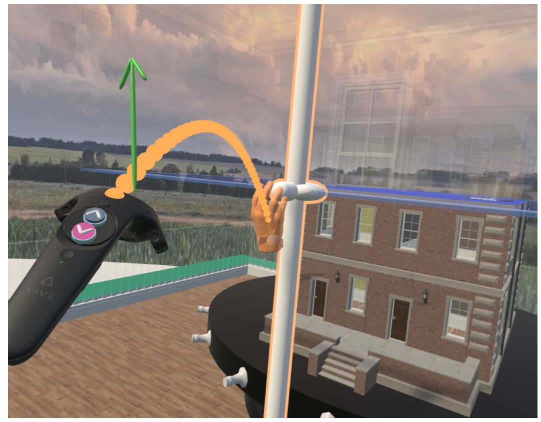


Figure 4(b)

Grabbing and translating the cutting plane downward

Figure 5

Local teleportation

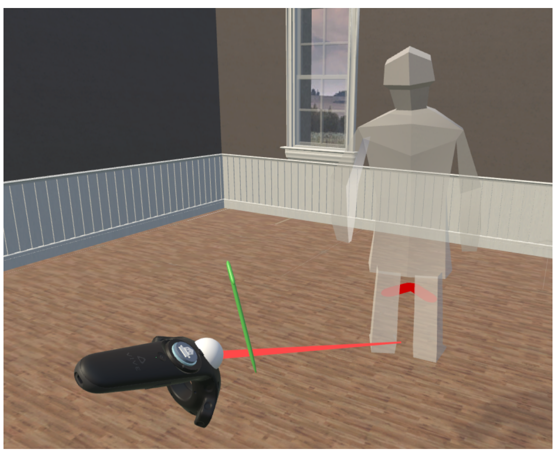


Figure 5(a)

Pointing and selecting a spot on the floor to teleport to in the full-scale model

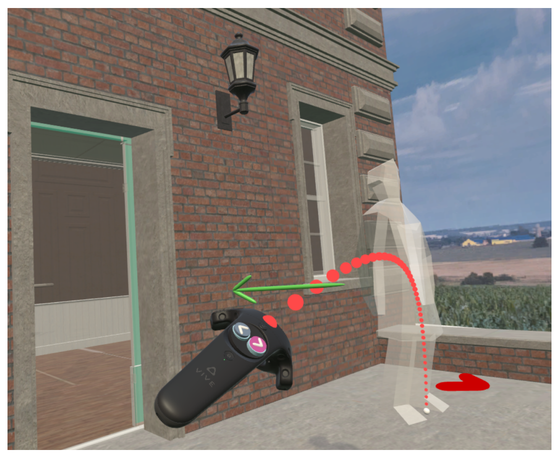


Figure 5(b)

Grabbing the spot on the ground and rotating the avatar to pre-orient the user to face away from the building post-teleportation

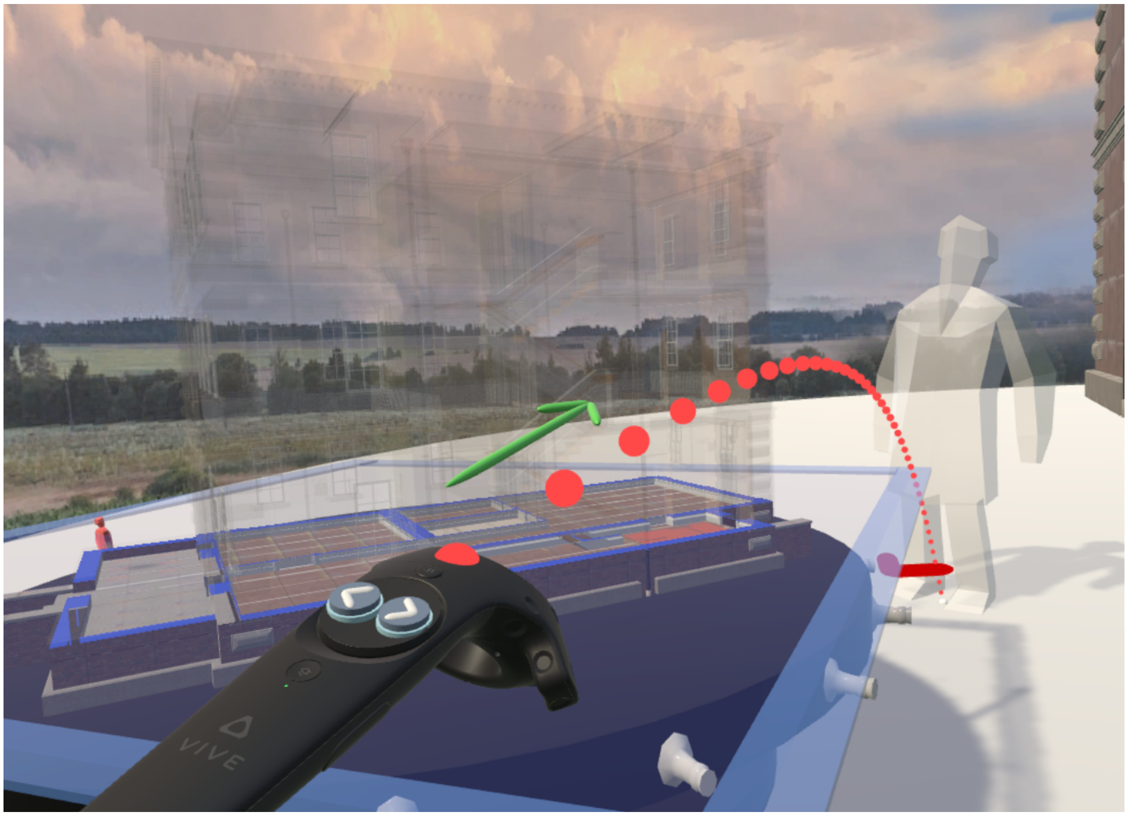


Figure 5(c)

Choosing to teleport to the other side of the small-scale model to get another view of it

Figure 6

Small-scale model teleportation

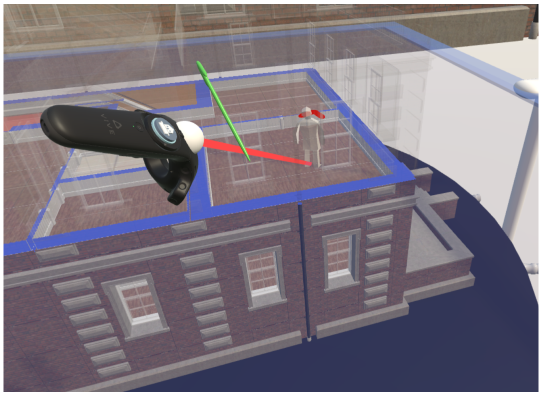


Figure 6(a)

Pointing and selecting a spot on the floor to teleport to in the full-scale model

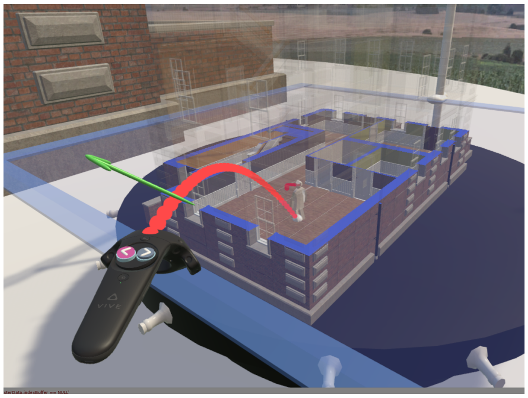


Figure 6(b)

Grabbing the spot on the floor and rotating the avatar to pre-orient the user to face the hallway post-teleportation

User Study

The effectiveness of the program's design was tested through a user study, and each participant was tested in the same way. They were prefaced with some information before starting, then the headset was placed on their head and the test would begin.

They were given enough information to get started, but details were purposely left out. I wanted to observe which interaction methods they learned on their own, and which they preferred. Therefore, they were given contextual information about the environment, controls, and the fact that they could move around in multiple ways and interact by pointing and grabbing. They were left to bridge the gap between what they wanted to do, and the method in which to do it.

Their task was to physically reach out and touch a manually placed orb floating in space. How they got to it was their choice, and this choice is what was observed. Each time they touch an orb, it moves to a new, preset location, and they'd have to find and touch it again, 10 times in total. The only hint of the orb's locations is a green arrow pointing to it, floating above the controller. If the participant did not figure out all interaction/navigation methods by the 5^{th} orb, they would be told everything through a quick, verbal tutorial.

Doing this enabled the analysis of self-learning, while also observing how they chose to operate once they knew of everything they could do within the program. For example, the 2^{nd} orb is on the first floor of the building while the 3^{rd} orb was on the third floor of the building, directly above. It was interesting to see if they chose to teleport along the floor, and up the stairs to the third floor, or instantly teleported there using the small-scale model.

Figure 7

Touching an orb in the user study

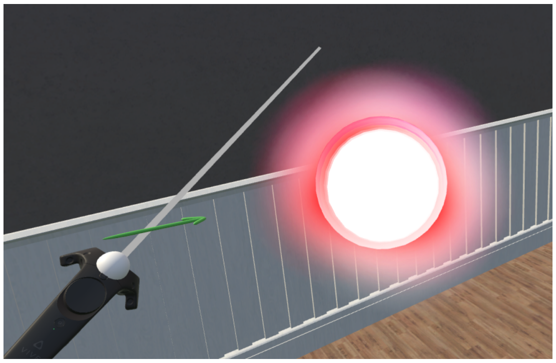


Figure 7(a)

Before touching the first orb, notice how the green arrow is pointing at it

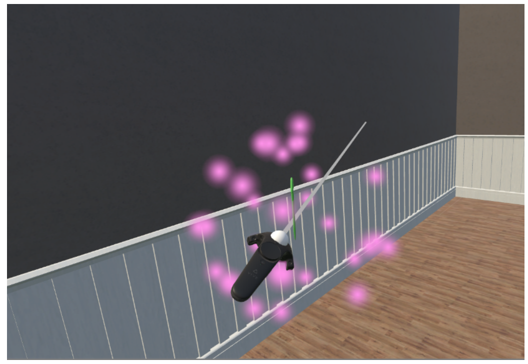


Figure 7(b)

Just after touching the first orb

This is what users were told beforehand:

Exploring Architectural models is important for architects, clients, or anyone who simply wants to understand and experience the design of a building before it is constructed. This is made easier with Virtual Reality.

This program will place you in a world that full-size building next to a small-scale model version of the exact same building.

In this world, you can:

1) Interact with the small-scale model, which sits atop a table
2) Navigate, or move yourself, around the outside/inside of the full-scale building through teleportation (which basically means you can instantly move to the point of your choosing)

Controls:

1) Use a single HTC Vive controller to interact with the world. The only buttons you need are the trackpad and the trigger. Point at something and then "grab" it using the trigger.

2) Keep an eye on your trackpad, as it will display different buttons that can be pressed, depending on what you're pointing at/ grabbing.

Goal: There is a floating, glowing orb that you must navigate to and touch with your hand controller. It will then move to a new position, and you will have to go find it and touch it again. The orb must be found and touched 10 times. The locations of the orb will mostly be inside the building. A small arrow above the controller will always be pointing in the direction of the orb's position, to give you a hint as to where it is.
Take your time exploring the features of this program; you are not being tested for completion time of this task.

Stats were recorded for each type of interaction performed by the user during the study. They were also timed. Each action will be analyzed for how many times it was performed, with a timestamp of each occurrence.

- Rotating the small-scale model using hand movement
- Rotating the small-scale model using directional buttons

- Raising/lowering the cutting plane using hand movement
- Raising/lowering the cutting plane using directional buttons

- Rotating the avatar using hand movement
- Rotating the avatar using directional buttons

- Teleporting using the small-scale model
- Teleporting in the local area

- Moving/hiding the small-scale model
- Touching the orb

Some specific questions that will be tested and answered by the user study:

1. Is having the option for both clicking and grabbing useful or just confusing for users? Should only one be available instead?
2. Do users prefer one manipulation method to the other? Why?
3. Is the design intuitive enough for quick self-learning, or will the input methods and program features need to be described and explained to users?
4. What features or interaction methods will users assume exist or try to use, or desire once they're finished?
5. How often do users pre-orient their avatar when teleporting? Is pre-orientation helpful?
6. Do users prefer to navigate primarily using scale-model or full-scale teleportation? Why and when?

Results

The user study was conducted through 10 participants, with 70% identifying as male and 30% identifying as female. The median age is 22, with the youngest participant being 22 and eldest being 57. Half of the participants have used a tethered VR Headset (HTC Vive, Oculus Rift, PS VR, etc.) in the past, with an average total usage of less than 4 hours. Half of the participants also have experience with smartphone-based VR (Google Cardboard, Samsung Gear VR, etc.), with an average total usage of less than 7 hours. Therefore, this group represents those who are rather inexperienced with VR, either never using it or using it only for a few hours. Also, half of the participants play video games regularly (at least a few hours a week). 80% reported never using 3D modeling or architectural design software, and 90% having never studied or worked in Architecture or a related field.

The following is the processed data from the quantitative measurements taken during each study, as well as the qualitative survey filled out by each participant after the study.

Interaction

Figure 8

Interaction Survey – Based on the Likert Scale (0 strongly disagree, 7 strongly agree), here are the average choices from the users of their opinions on interaction in the program.

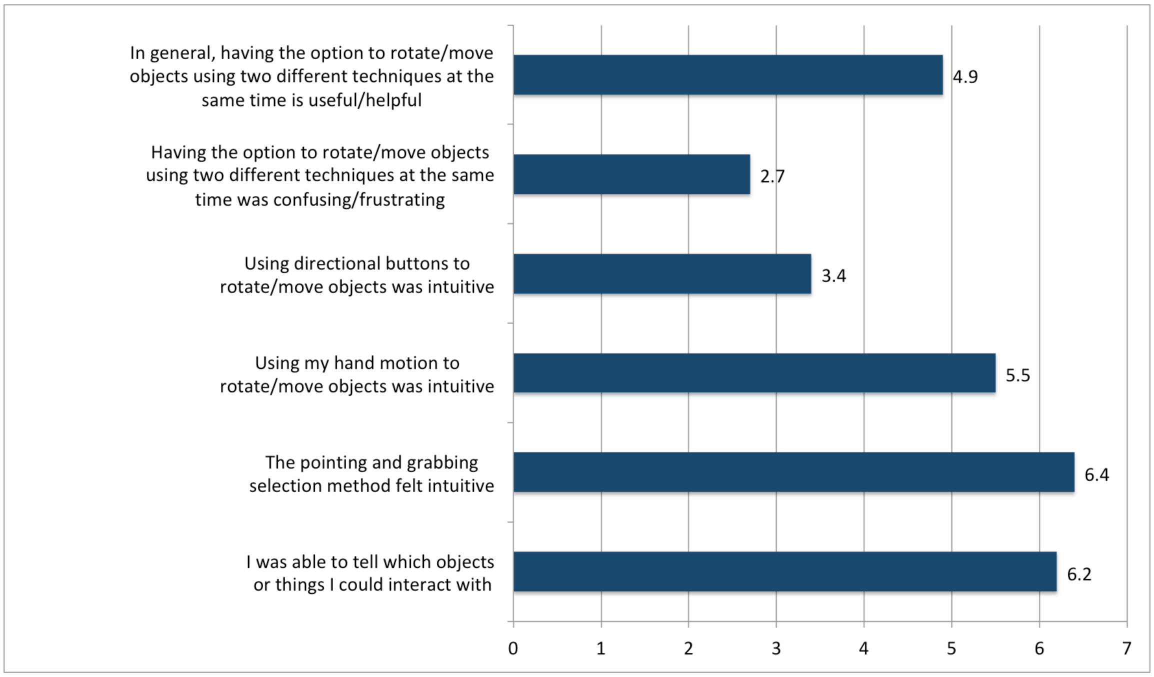
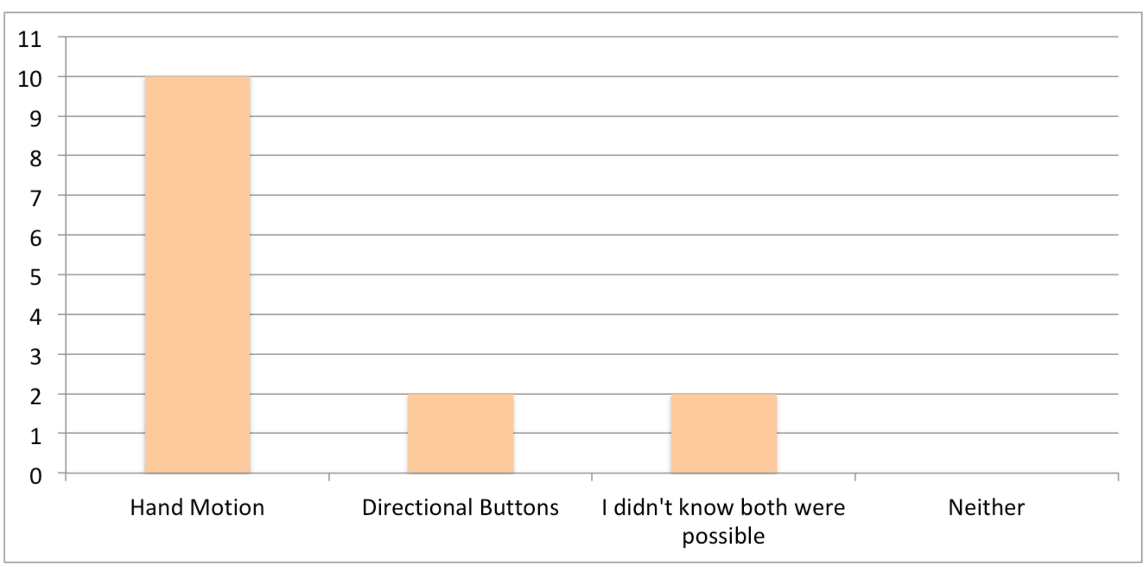


Figure 9

Manipulation Preference – The number of users who selected a manipulation method as their preferred choice. Some users selected both.



Navigation

Figure 10

Navigation Survey – Based on the Likert Scale (0 strongly disagree, 7 strongly agree), here are the average choices from the users of their opinions on navigation in the program.

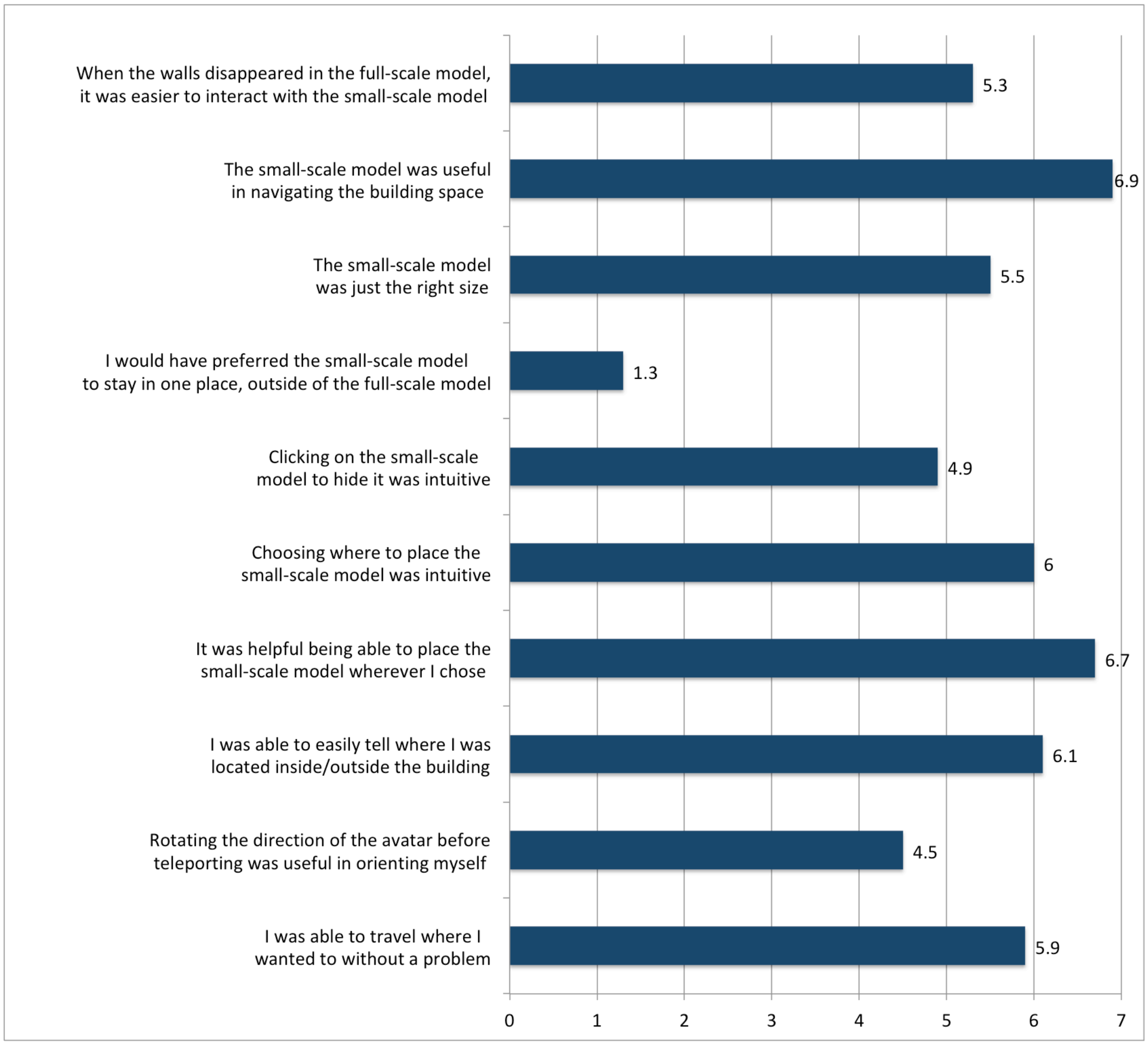


Figure 11

Teleportation Preference – The number of users who selected a navigation method as their preferred choice. Some users selected both.

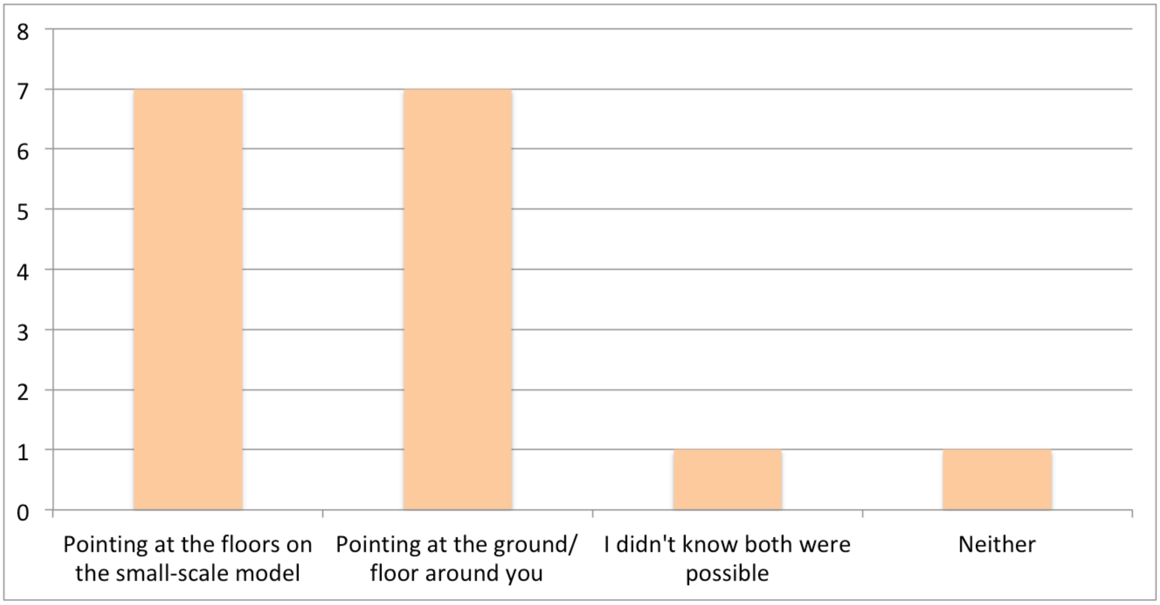


Table 3

User Action Legend – Describes the key used to represent each user action that was measured during each user study trial

Action	Description
model_hand	Rotating the small-scale model table using hand motion
model_buttons	Rotating the small-scale model table using directional buttons
model_move	Moving the small-scale model to another location in the full-scale model
model_hide	Hiding the small-scale model (temporarily removed from the scene)
cp_hand	Raising/lowering the cutting plane using hand motion
cp_buttons	Raising/lowering the cutting plane using directional buttons
avatar_hand	Rotating the avatar before teleportation using hand motion
avatar_buttons	Rotating the avatar before teleportation using directional buttons
teleport_model	Teleporting via the small-scale model
teleport_local	Teleporting via local teleportation

Table 4

User Action Count – The number of times each action was executed by the users. This table includes the count for each user (P1, P2, etc...) and the average.

Action	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Avg.
model_hand	23	15	4	2	4	19	6	8	22	19	12.2
model_buttons	8	2	0	0	0	2	0	0	5	1	1.8
model_move	37	44	19	25	19	17	24	26	20	17	24.8
model_hide	17	10	7	3	7	5	2	5	6	3	6.5
cp_hand	7	11	16	4	3	12	6	6	14	6	8.5
cp_buttons	1	0	1	0	0	2	0	0	0	0	0.4
avatar_hand	10	0	8	1	2	7	1	7	4	26	6.6
avatar_buttons	4	9	1	2	0	2	0	2	13	5	3.8
teleport_model	13	19	15	17	7	7	13	15	10	11	12.7
teleport_local	202	31	85	81	74	90	39	39	126	103	87

Figure 12

Average User Action Count – A bar graph visualization of the average number of times each action was executed by the users.

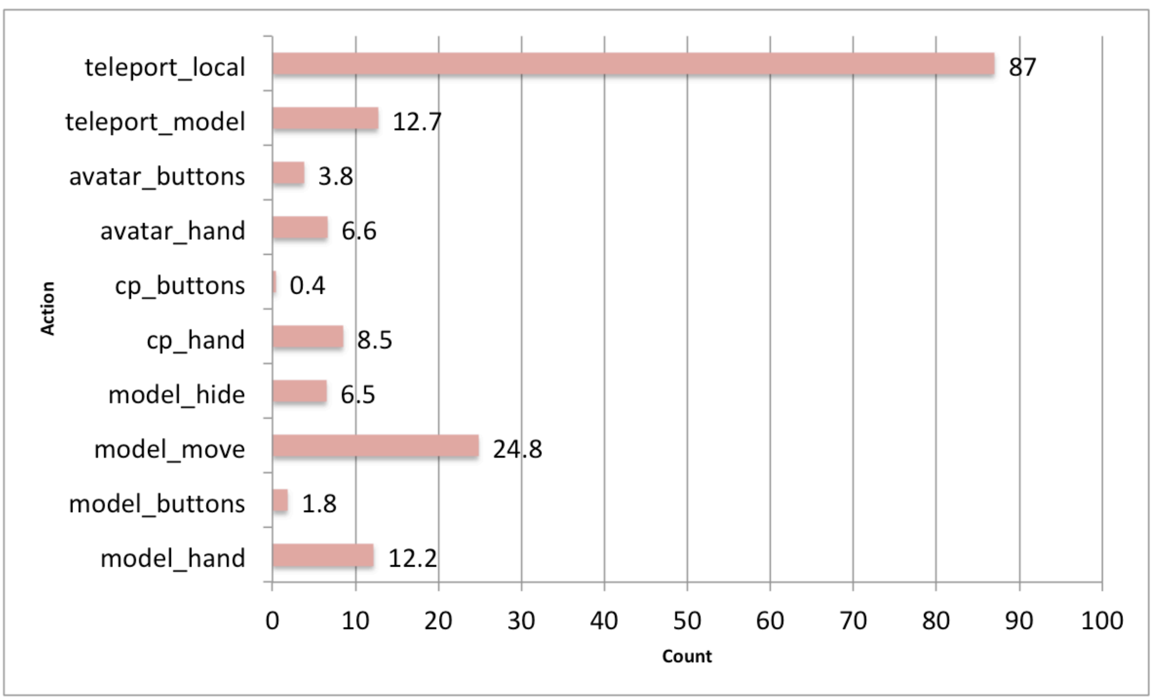


Table 5

User Action Discovery – The last touched orb before the user first executed a given action. The orbs were touched chronologically, from 1 to 10. If the orb is 0, then the user discovered an action before they even touched the first orb. This table includes orb discovery number for each user (P1, P2, etc..) and average for both the orb discovery number and average time it took to discover.

Action	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Avg. Orb Number	Avg. Discovery Time
model_hand	0	1	0	0	3	0	0	0	0	1	0.5	76.20
model_buttons	5	3	10	10	10	5	10	10	0	5	6.8	557.59
model_move	0	0	2	1	3	1	0	0	0	1	0.8	96.73
model_hide	0	1	2	1	3	1	0	0	0	5	1.3	163.7
cp_hand	5	1	2	0	3	1	0	5	5	1	2.3	314.49
cp_buttons	7	10	3	10	10	1	10	10	10	10	8.1	680.51
avatar_hand	0	10	0	7	8	5	7	5	5	0	4.7	290.79
avatar_buttons	5	3	5	7	10	5	10	5	0	5	5.5	464.07
teleport_model	0	0	0	2	3	0	0	0	5	1	1.1	125.86
teleport_local	0	0	0	0	0	0	0	0	0	0	0	44.08

Table 6

User Study Task Completion Time – The time, in seconds, that the users took to touch all 10 orbs

Action	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Avg.
Finish time (seconds)	2085.07	494.77	650.84	710.76	528.03	701.9	396.8	519.8	983.04	945.47	801.65

Figure 13

Average Orb Number for Action Discovery – A bar graph visualization of the last orb number, on average, to be touched before an action was first executed (or discovered) by the users.

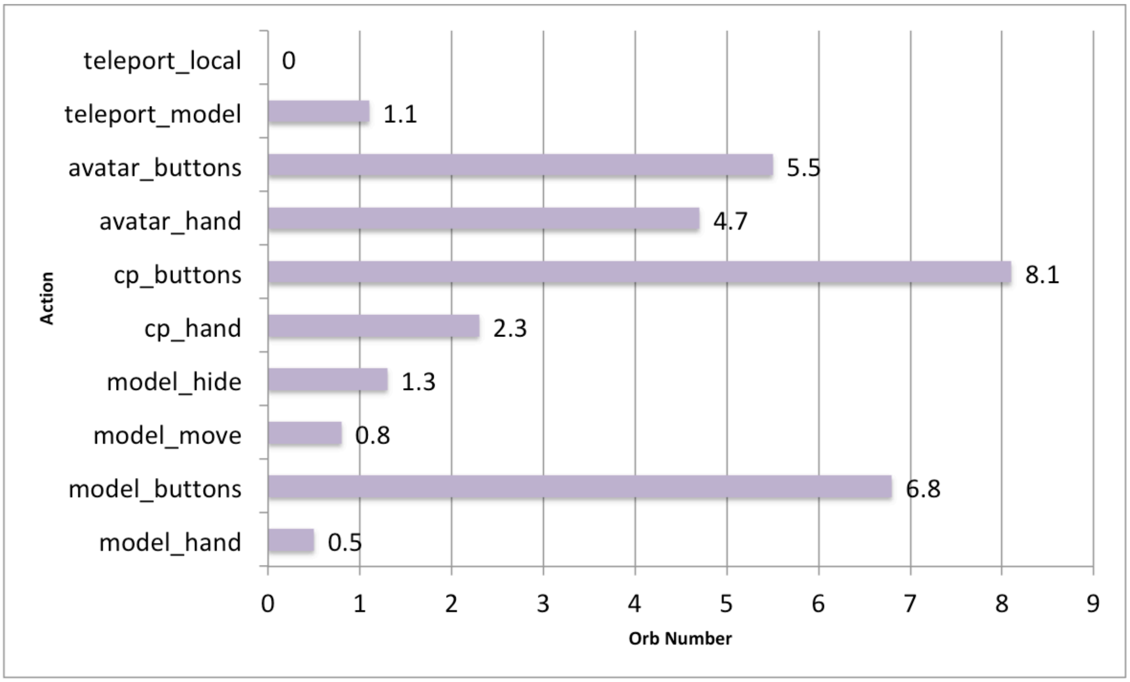
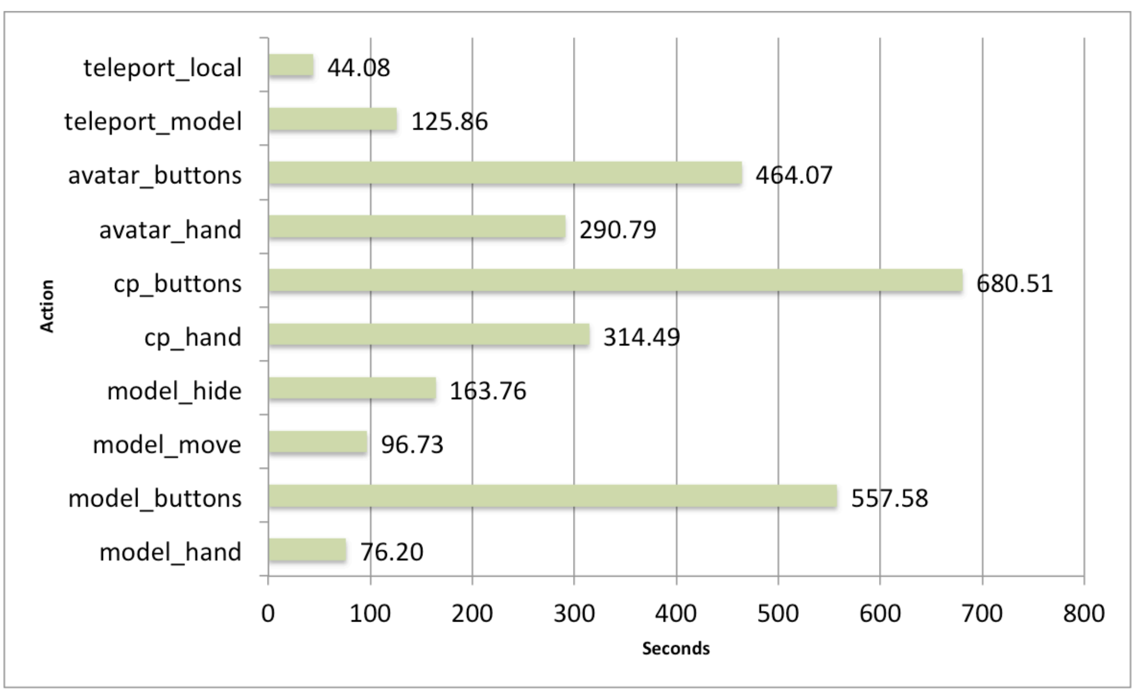


Figure 14

Average Time for Action Discovery – A bar graph visualization of the average discovery time of each action (that is, the first time each action was executed by the users).



Discussion

Note that if users had not discovered a method of navigation, selection, or manipulation by the time they touched the fifth orb, they were given a verbal tutorial about what they had not yet figured out.

Local teleportation was often the first action to be used/discovered by users, with the shortest average action discovery time of 44.08 seconds and always before the first orb was touched. This makes sense since the ground/floor covers a greater area on screen, therefore having a greater likelihood of being pointed at with the raycast while the user is "figuring things out". Also, users knew they'd have to move to reach the first orb, so pointing at the ground was instinctual, especially since a few of the users were familiar with teleportation already. Interestingly, some of the users first thought to point at the orb in order to collect it, instead of virtually moving next to it and touching it, even though the latter was the explained means of doing so. Furthermore, local teleportation was used much more than scale-model teleportation (on average 87 times vs. 12.7 times throughout), but this is to be expected since going from point A to point B with local teleportation requires multiple teleports, while using the scale-model often requires only one. Note that the ability to teleport using the small-scale model was sometimes discovered a tad late, after one or two orbs were collected, but all users discovered this on their own, except for one who was told after collecting the fifth orb. It was also noticeable that if an orb was on a different floor, or at a distant room on the same floor, users would prefer to use the scale-model to get there quicker. In the survey, both methods of teleportation were equally preferred, and one user preferred neither, but they all felt that there were no problems traveling where they wanted.

As a navigational aid for spatial awareness, the design of the small-scale model proved useful for participants, as they strongly agreed (6.9/7 Likert Scale average) in the survey that it was helpful in navigating the building space. They also highly agreed that choosing where to place the small-scale model was helpful and intuitive, and that they would not have wanted it to be stationary (in its own room or outside of the model). Having the full-scale model walls go down on the same floor where the small-scale model was likely contributed to the effectiveness of moving the small-scale model around, especially since users generally concurred and understood that the walls going down prevented obstruction with scale-model interaction. Users also agreed that they could easily tell where they were located within the full-scale model, often summoning the small-scale model to see where they were. This was likely helped by the small, red avatar in the small-scale model, which represented the user's current location and orientation within the full-scale model. This is all according to their survey answers in Figure 10. Half of the users would start moving the small-scale model before they even touched the first orb, and the other half did so before touching the fourth orb. It was moved around 24.8 times on average.

Regarding interaction, the pointing and grabbing selection method was said to be highly intuitive (6.4/7 Likert average) among users, while there was more a polarized choice in manipulation techniques. The raycast changing colours, affordances such as the spokes around the table, and magic objects, such as the ghost hands and teleportation avatar, likely helped in making the interactive potential of things noticeable. That being said, not all manipulation possibilities were as equally obvious to users. The most apparent manipulation was the rotation of the scale-model table, with most users, on average, selecting and spinning the table using hand-motion before they even touched the first orb. Second was the small-scale model's cutting plane, which was selected and moved vertically, using hand-motion after the first few orbs were touched. The main problem was with rotating the teleportation avatar, which is something that over 80% of users did not discover this until being told about it after touching the fifth orb. Users would point at the ground/floor, click, and then release without holding; it seems the grabbing metaphor did not extend to "grabbing" the floor. One possibility for this is that there was no affordance or magic object indicating the ability to rotate the teleportation avatar. A future study focusing specifically on studying the affects of different affordances in various manipulation contexts could extend this to further make sense of it. Also, the usefulness of pre-orienting the avatar was met with mixed opinions, but that could be because many users had become accustomed to not using it before they knew they could. Some users did mention that they preferred physically turning around to pre-orient themselves since it felt more realistic, but whether they'd enjoy that or not after prolonged use is a question for a future study.

Recall that users had the choice of manipulating the small-scale model table, cutting plane, and teleportation avatar using either hand-motion or the directional buttons on the controller's trackpad. Even though they were told to keep an eye on the trackpad for buttons before they started, over 80% of users did not discover the directional buttons until being told about them after touching the fifth orb. It was mentioned by several users that attention was never focused directly on the controller, so it was easy to miss the buttons. Even once they were aware of the buttons, the directional buttons were not preferred and sometimes never used, with all ten participants preferring hand-motion and only two preferring both. Despite their survey answers, it is an interesting observation that although the buttons were almost never used for manipulating the small-scale model table or cutting plane, they were almost used equally as many times as hand-motion for pre-orienting the transportation avatar. The purpose of the buttons was to allow for a more precise, less strenuous, and unrestricted control over the manipulation of objects, but it appears that this was not required for general usage, especially since the sophistication of the modern VR hardware already allows for precise hand-motion. Perhaps another study where long-term usage is considered would better show the cost-benefit analysis of using directional buttons over hand motion. According to Figure 8, users agree that it is conceptually useful to have multiple options for manipulation techniques; however, their options were mixed on how intuitive it felt in this simulation.

User Suggestions and Design Improvements

The design of this architectural model explorer purposely featured basic features and manipulation techniques for simple use. Users were asked to see what other features they'd like included, and which they'd like improved.

For the cutting plane, it was suggested that the vertical apparatus should be redesigned to not be on only one side of the table, since it's sometimes occluded. Plus, it was suggested that the cutting plane should snap to the bottom of each floor so that the height of the walls wouldn't have to be fidgeted with in order to expose the floor itself (basically section off the floor as a whole chunk). For selection, a parabolic pointer would have been preferred for teleportation, as well as the ability to teleport to upper or lower floors simply by pointing up or down. The option to use two controllers was also mentioned.

For some users, especially those who were in VR for the first time, more scenic realism was requested for a less jarring experience. They asked for hand railings in the staircase, fences outside so they didn't feel like they'd fall of the platform, and furniture. One user felt that the teleportation avatar was too intrusive to the realism, while one user had no idea that it even was an avatar and was confused as to what/who it was.

Alternative future testing could be done with a different type of task for the user study. One participant felt like they were playing a game looking for the orbs instead of exploring a building, which was not the intention.

Conclusion

Based on the results of the user study, the design of the program was successful in being an architectural model viewer for naïve viewers. Users discovered most interaction on their own and did not need clarification or reminders on how to interact; most stuff was apparent. The task was completed in roughly 13 minutes on average, and no one got stuck.

Bringing the small-scale model with you is useful and helpful, and perhaps the most important takeaway from this study. A small-scale version of a building model that can be relocated is a great multi-purpose tool for navigational aid. It combines the benefits of a WIM with the metaphor of a real architectural model on a desk. The user's navigational flow is not disrupted since they can bring the small-scale model to them, instead of going to the model. Its unchangeable scale, and limited features (rotation, teleportation, using the cutting plane) were enough to help new users without overwhelming, confusing, or frustrating them. Future user studies should focus specifically, and separately, on the individual components of the small-scale model's usage; however, in a holistic sense, the design used here worked well.

Being able to teleport both locally and using the small-scale model is advantageous, especially when a building design is complicated and large. Pre-orienting the avatar before teleporting was useful, but not necessary, at least in the context of this architectural model. Those who used it, enjoyed it, and those who didn't, didn't feel they needed it.

Pointing and grabbing for selection is useful for self-discovery and quick learning, and indication through magic/affordance cues is especially important. It cannot be assumed that even if a manipulation technique is possible, easy to use, and explained, that it will be discovered. Users were given the option for manipulation via hand motion or directional buttons, but hand-motion was good enough for general usage in architectural VR, and more immersive. It was evident by the user study that inexperienced VR users are instinctively using motion to move and manipulate. There was a chance that they'd stick with button presses to interact with everything, since that is how most computer interaction works, but this was not the case. The lack of directional button usage also showed that putting any input indicators directly on the controller should be avoided. Note, however, that for those who do not want to stand up or physically move or turn, having alternative manipulation techniques like buttons is probably useful; a deeper study is needed to confirm this.

For anyone designing an architectural model explorer in VR for the average user, the research in this paper can potentially be used as a preliminary guide.

Acknowledgements

Thank you to Ryerson's Department of Computer for providing me with a workstation and access to the research lab in 211 to work on this thesis. And thank you to Dr. Tim McInerney for the HTC Vive and PC, and for his guidance, knowledge, and supervision.

References

[1] Abdelhameed, W. A. (2013). Virtual Reality Use in Architectural Design Studios: A Case of Studying Structure and Construction. Procedia Computer Science, 25, 220– 230. https://doi.org/10.1016/j.procs.2013.11.027

[2] Al-Adhami, M., Ma, L., & Wu, S. (2018, July 22). Exploring Virtual Reality in Construction, Visualization and Building Performance Analysis. Presented at the 34th International Symposium on Automation and Robotics in Construction, Taipei, Taiwan. https://doi.org/10.22260/ISARC2018/0135

[3] Boletsis, C., & Boletsis, C. (2017). The New Era of Virtual Reality Locomotion: A Systematic Literature Review of Techniques and a Proposed Typology. Multimodal Technologies and Interaction, 1(4), 24. https://doi.org/10.3390/mti1040024

[4] Bowman, D. A. (2000). 3D user interface design. ACM SIGGRAPH 2000, Course Notes, 36, 200–0.

[5] Bowman, D. A., & al, et. (1997). An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments.

[6] Burtnyk, N., Khan, A., Fitzmaurice, G., Balakrishnan, R., & Kurtenbach, G. (2002). StyleCam: Interactive stylized 3D navigation using integrated spatial & temporal controls. In Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (Vol. 4, pp. 101–110). Paris: ACM.

[7] Coroado, L., Pedro, T., D ’alpuim, J., Eloy, S., & Dias, M. (2015, January 1). VIARMODES: Visualization and interaction in immersive virtual reality for architectural design process.

[8] Elvezio, C., Sukan, M., Feiner, S., & Tversky, B. (2017). Travel in large-scale head-worn VR: Pre-oriented teleportation with WIMs and previews. 2017 IEEE Virtual Reality (VR), 475–476. https://doi.org/10.1109/VR.2017.7892386

[9] Fitzmaurice, G., Matejka, J., Mordatch, I., Khan, A., & Kurtenbach, G. (2008). Safe 3D navigation. In Proceedings of the Symposium on Interactive 3D graphics and games (pp. 7–15). Redwood Shores, CA: ACM. doi:10.1145/1342250.1342252

[10] Frees, S. (2010). Context-driven interaction in immersive virtual environments. Virtual Reality, 14(4), 277–290. https://doi.org/10.1007/s10055-010-0178-2

[11] Garrido-Vásquez, P., & Schubö, A. (2014). Modulation of visual attention by object affordance. Frontiers in Psychology, 5. https://doi.org/10.3389/fpsyg.2014.00059

[12] Glueck, M., & Khan, A. (2011). Considering multiscale scenes to elucidate problems encumbering three-dimensional intellection and navigation. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 25(04), 393–407. https://doi.org/10.1017/S0890060411000230

[13] Kopper, R., Ni, T., Bowman, D. A., & Pinho, M. (2006). Design and Evaluation of Navigation Techniques for Multiscale Virtual Environments. IEEE Virtual Reality Conference (VR 2006), 175–182. https://doi.org/10.1109/VR.2006.47

[14] Koutamanis, A. (2006). BUILDINGS AND AFFORDANCES. In J. S. Gero (Ed.), Design Computing and Cognition '06 (pp. 345–364). https://doi.org/10.1007/978-1-4020-5131-9_18

[15] Langbehn, E., Lubos, P., & Steinicke, F. (2018, March 26). Evaluation of Locomotion Techniques for Room-Scale VR: Joystick, Teleportation, and Redirected Walking.

[16] Luo, Z., Luo, W., Wickens, C. D., & Chen, I.-M. (2010). Spatial learning in a virtual multilevel building: Evaluating three exocentric view aids. International Journal of Human-Computer Studies, 68(10), 746–759. https://doi.org/10.1016/j.ijhcs.2010.06.004

[17] Motta, C. F., & Luis, M. (2017). Visualization of Architectural Models in Virtual Really. CrazyBump normal mapping. http://www.crazybump.com/. Accessed: 2015-06- 23. Retrieved from http://vitela.javerianacali.edu.co/handle/11522/8233

[18] Nguyen-Vo, T., Riecke, B. E., & Stuerzlinger, W. (n.d.). Simulated Reference Frame: A Cost-Effective Solution to Improve Spatial Orientation in VR. 8.

[19] Pinchbeck, D. (2009). An affordance based model for gameplay. 9.

[20] Salingaros, N. (2017). WHY WE NEED TO "GRASP" OUR SURROUNDINGS: OBJECT AFFORDANCE AND PREHENSION IN ARCHITECTURE. JOURNAL OF ARCHITECTURE AND URBANISM, 41(3), 163–169. https://doi.org/10.3846/20297955.2017.1376003

[21] Seibert, J., & Shafer, D. M. (2018). Control mapping in virtual reality: effects on spatial presence and controller naturalness. Virtual Reality, 22(1), 79–88. https://doi.org/10.1007/s10055-017-0316-1

[22] Shin, D.-H. (2017). The role of affordance in the experience of virtual reality learning: Technological and affective affordances in virtual reality. Telematics and Informatics, 34(8), 1826–1836. https://doi.org/10.1016/j.tele.2017.05.013

[23] Stoakley, R., J. Conway, M., & Pausch, Y. (1970). Virtual Reality on a WIM: Interactive Worlds in Miniature. https://doi.org/10.1145/223904.223938

[24] Sun, C., Hu, W., & Xu, D. (2018). Navigation modes, operation methods, observation scales and background options in UI design for high learning performance in VR-

based architectural applications. Journal of Computational Design and Engineering. https://doi.org/10.1016/j.jcde.2018.05.006

[25] Wang, X. (2009). Augmented Reality in Architecture and Design: Potentials and Challenges for Application. International Journal of Architectural Computing, 7(2), 309–326. https://doi.org/10.1260/147807709788921985

[26] Wingrave, C. A., Haciahmetoglu, Y., & Bowman, D. A. (2006). Overcoming World in Miniature Limitations by a Scaled and Scrolling WIM. 3D User Interfaces (3DUI'06), 11–16. https://doi.org/10.1109/VR.2006.106

[27] Wolf, K., Funk, M., Khalil, R., & Knierim, P. (2017, November 26). Using virtual reality for prototyping interactive architecture. 457–464. https://doi.org/10.1145/3152832.3156625

[28] Yu, D., Liang, H.-N., Lu, F., Nanjappan, V., Papangelis, K., & Wang, W. (n.d.). Target Selection in Head-Mounted Display Virtual Reality Environments. 27.

[29] Zheng, M. (2017). Navigation in VR for visual analytics using the latest consumer-grade hardware (Masters, OCAD University). Retrieved from http://openresearch.ocadu.ca/id/eprint/1590/